

**NUMERICAL MODELLING OF
SCALE-DEPENDENT DAMAGE AND FAILURE
OF COMPOSITES**

BOYANG CHEN

(DIPLOME, ÉCOLE POLYTECHNIQUE, FRANCE)

(B.ENG.(HONS.), NUS, SINGAPORE)

A THESIS SUBMITTED

FOR THE DEGREE OF NUS-IMPERIAL COLLEGE

JOINT DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

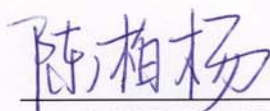
2013

DECLARATION

I hereby declare that this thesis is my original work and it has been
written by me in its entirety.

I have duly acknowledged all the sources of information which have been
used in the thesis.

This thesis has also not been submitted for any degree in any university
previously.



BOYANG CHEN

August 21, 2013

Acknowledgements

To my supervisors, Prof. Tong-Earn Tay, Dr. Silvestre Taveira Pinho and Dr. Pedro Miguel Baiz Villafranca, for your continuous support, guidance and advice throughout the journey of my PhD. Special thanks to Dr. Baiz and Prof. Tay for initiating this collaboration, and to Dr. Baiz for introducing Dr. Pinho into this collaboration.

To the research scholarship of National University of Singapore, for funding this project; and to the joint PhD programme between National University of Singapore and Imperial College London, for providing such a collaborative platform for PhD researches.

To Dr. Nelson Vieira De Carvalho, for your active engagement and valuable input on the development of the floating node method (Chapter 4). To Dr. Soraia Pimenta, for the help on graphics and Latex. To Dr. Matthew John Laffan, for providing the material property data of the IM7-8552 carbon/epoxy composite for the work in Chapter 3. To Dr. Stephanie Miot, Dr. Gaurav M. Vyas and Dr. Julian Dizy Suarez, for the help on Abaqus installation and using the HPC of Imperial College. To Dr. Martin Whiteside, for the help on computational resources. To Dr. Adam Connolly, for the help on Matlab and Shell script. To Dr. Xiu-Shan Sun, Dr. Muhammad Ridha and Dr-to-be. André Antoine Renaud Wilmes, for the useful discussions on the modelling of composites. To Silvestre, Adam and André for the floating moments in Brazil.

To all the friends that I have met in different corners of the world during my PhD, for the love, joy and reflections that you have brought to me.

Finally, to my parents, Chen Sheng-Yi and Yang Hai-Yan, for your constant love, trust, support and encouragement in my life; and to Yu-Hua, for your understanding and support of my work, for your love and company, as well as the changes and sacrifices you've made for us.

Contents

Summary	vii
List of Figures	viii
List of Tables	xiii
Nomenclature	xxiv
1 Introduction	1
1.1 Overview of the failure mechanisms of composites	3
1.2 Introduction of the open-hole tension size effects	4
1.3 Introduction of the thickness-dependence of translaminar fracture toughness	8
1.4 Brief review of the failure theories	9
1.5 Objectives of the research	10
1.6 Structure of the thesis	11
2 Literature review of numerical methods for the failure modelling of composites	13
2.1 Introduction	13
2.2 Remeshing	14
2.3 Stiffness degradation method	16
2.4 Cohesive element	16
2.5 Smeared crack formulation	20
2.6 eXtended Finite Element Method	21
2.7 Phantom Node Method	23
2.8 Discussions	25
3 Open-hole tension size effect	28

3.1	Introduction	28
3.2	Failure Theory	32
3.2.1	Fibre Failure	32
3.2.2	Thickness Dependence of Translaminar Fracture Toughness	34
3.2.3	Matrix Failure	36
3.2.4	Delamination	39
3.3	Mesh Refinement Study of Finite Element Models	43
3.3.1	Classical Lamination Theory (CLT) model	43
3.3.2	Continuum Shell Laminate with Interface (CSLI) model	44
3.3.3	Continuum Shell Perfect Bonding (CSPB) model	50
3.4	Size Effect Predictions	51
3.4.1	Sublaminar-scaling thickness size effect	51
3.4.2	Ply-scaling thickness size effect	54
3.4.3	In-plane size effect of sublaminar-scaled specimens	58
3.4.4	In-plane size effect of ply-scaled specimens	58
3.4.5	Parametric sensitivity analysis	61
3.5	Conclusion	63
3.6	Acknowledgement	65
3.7	Publications	65
4	Floating Node Method	66
4.1	Overview of the Phantom Node Method	69
4.1.1	Introduction	69
4.1.2	Without a discontinuity	71
4.1.3	With a strong discontinuity	72
4.1.4	With other types of discontinuities and scenarios	74
4.1.5	Comparison with other methods	75
4.2	Floating Node Method	77
4.2.1	Overview of the approach	77
4.2.2	Without a discontinuity	78
4.2.3	With a strong discontinuity	78
4.2.4	With weak discontinuities and cohesive cracks	82

4.2.5	Different geometries for the discontinuities and integration	84
4.2.6	Element topology and assembly	89
4.2.7	Comparison with other methods	90
4.2.8	Formulation of FN elements for composite laminates	92
4.2.9	Other details	96
4.3	Validation	105
4.3.1	Convergence of Stress Intensity Factors for an edge crack	105
4.3.2	Evaluation of Stress Intensity Factors for a centre slant crack	105
4.3.3	Double Cantilever Beam bending test	106
4.4	Applications	109
4.4.1	Crack density evolution in a toughened glass/epoxy cross-ply laminate	109
4.4.2	Crack density evolution in AS4/3501-6 carbon/epoxy cross-ply laminates	113
4.5	Discussion and conclusion	120
4.6	Acknowledgement	122
4.7	Publications	122
5	Conclusions	124
6	Future work	128
6.1	Developing a Multi-scale FNM element for laminates	128
6.2	Extension of the FNM to higher-order and 3D elements . . .	129
6.3	Reliable strength predictions of the ply-scaled open-hole tension laminates	130
6.4	Modelling of delamination migration in composite laminates	131
6.5	Evaluation of the edge status variable approach of the FNM in representing a large number of discontinuities	132
6.6	Error estimation and adaptivity	133
6.7	Strain smoothing in distorted sub-elements	133
	Bibliography	135

Appendices	154
A Phantom Node Method Evaluation and Extension	155
A.1 Literature Review	155
A.2 Comparison of FEM and PNM	156
A.3 Extension	161
A.3.1 Separation of numerical domain and material domain	161
A.3.2 Modelling cohesive cracks by PNM cohesive elements	163
A.4 Applications	165
A.4.1 Modelling of single cohesive crack	165
A.4.2 Modelling of multiple crack interactions in a compos-	
ite laminate	166
A.5 Discussions	172
A.6 Conclusion	175
A.7 Publications	175
B Error in the mapping of a straight crack in the PNM/XFEM	176
C Sample codes for the implementation of the floating node method	179
C.1 Sample Abaqus UEL subroutine	180
C.2 Input file	209
C.2.1 Raw input file from Abaqus	209
C.2.2 Pre-processed input file for UEL	212
C.3 Pre-processing Matlab programme for Job-1	215
C.4 Post-processing Matlab programme for Job-1	223

Summary

This research focuses on establishing an accurate numerical model for the failure modelling of composite laminates. A computational study of the size effects of open-hole tension composite laminates is carried out, using existing failure theories and numerical methods. Translaminar fracture toughness has recently been experimentally determined to be thickness dependent; it is accounted for in the numerical model as a new model input. The thickness size effect in the strength of laminates failed by pull-out is accurately predicted by a deterministic model. Models which neglect delamination are found to have mesh-dependent and over-estimated strength prediction. A model with cohesive elements between plies predicts the correct failure mode, but not the correct strength, for laminates failed by delamination.

Owing to the above conclusions, a floating node method is developed for modelling multiple discontinuities within a finite element. Extra nodes are used to represent the discontinuities. These extra nodes do not coincide with the real nodes; their position for each element is only defined once failure for that element is predicted. The proposed method is well suited for modelling weak and cohesive discontinuities, for the use of transition elements at the crack tip, for the representation of complex crack networks inside an element, and for use with the virtual crack closure technique. Validation examples show that the proposed method can predict stress intensity factors and crack propagation accurately. An application example shows that the proposed method can predict the transition from matrix cracking to delamination in cross-ply composite laminates by accurately representing T-shaped cracks inside an element.

List of Figures

1.1	Coordinates and notations of composites	2
1.2	Common failure modes in a composite laminate	3
1.3	Failure modes in an experiment from [155].	4
1.4	Matrix crack	5
1.5	Delamination	6
1.6	Three failure modes in open-hole tension experiments [53] . .	7
1.7	Edge-view image of an open-hole tension experiment [53] . .	7
1.8	Schematic drawing of the interaction between different failure mechanisms [154].	8
1.9	Fractographic images of the translaminar fracture surface [79]	9
2.1	Remeshing	15
2.2	Cohesive zone theory	17
2.3	A typical cohesive element formulation [103].	18
2.4	Smeared crack formulation	21
2.5	A typical XFEM representation of a crack in a mesh	22
2.6	An illustration of the Phantom Node Method	24

2.7	Calculations of the Phantom Node Method	24
3.1	Inplane-scaling in the open-hole tension experiment [156] . .	29
3.2	Different thickness-scaling methods for laminates of lay-up [45 _m /90 _m / - 45 _m /0 _m] _{ns} [156]	29
3.3	Different strength size effects [53]	30
3.4	FEM implementation of the thickness dependence of translaminar fracture toughness	34
3.5	Illustration of matrix cracking and local material directions.	36
3.7	Four meshes of different element sizes for the open-hole model	44
3.8	Mesh refinement study of different numerical models	45
3.9	Failure patterns of the CLT and the CSLI model	47
3.10	Comparison of the CLT model and the CSLI model on mesh refinement	49
3.11	Comparison between simulation and experiment of the pull- out delamination patterns	52
3.12	Failure patterns of ply-scaled laminates	56
3.13	Comparison between simulation and experiment of the de- lamination patterns	57
3.14	The smeared crack model approximates the sharp matrix crack tip into a blunted crack tip on the interface	60
3.15	Summary of the predictions on strength size effects	61
4.1	Comparison of assembly architectures of different methods .	67
4.2	Phantom Node Method	70

4.3	FEM or remeshing	71
4.4	Floating Node Method	79
4.5	Weak discontinuity and cohesive crack	82
4.6	Examples of different geometries of elements, partitions and discontinuities that can be modelled by the FNM(see key in Figure 4.4).	85
4.7	FNM with one triangular partition and one quadrilateral partition	86
4.8	Example of local DoF, vertices and edges numbering for a FN element	87
4.9	A FN element can represent a wide range of cracking scenar- ios by adopting different partitioning strategies for integration	88
4.10	Modelling the intersection between matrix cracks and de- lamination with matching and non-matching meshes	93
4.11	The FN element formulated for cross-ply composites with a matching partition at the interface after matrix cracking (a); and the non-matching partition after matrix cracking (b).	94
4.12	Using local refinement elements and transition elements to represent the crack tip more accurately (see key in Figure 4.4).	98
4.13	For this edge crack model, the FNM converges monotoni- cally, unlike the PNM.	104
4.14	For this slant crack model, the FNM captures the SIF well in modes I and II for different angles θ	106
4.15	PNM and FNM comparison on the DCB example	107
4.16	DCB simulation using FNM	107

4.17	Modelling the transition from matrix cracking to delamination in a cross-ply composite specimen. The red dots indicate failure at the corresponding integration points.	112
4.18	The saturation crack density predictions	113
4.19	Matrix crack density evolution in [98]	114
4.20	FNM Simulation results of all twenty runs vs. experimental data	118
4.21	Average of the twenty simulation results vs. experimental data	119
4.22	Failure pattern predictions for the same section of all three laminates	119
6.1	A multi-scale FN element	130
6.2	Delamination migration in a multi-directional composite laminate from Tao and Sun [138]	132
A.1	Modelling of a crack by the FEM and the PNM	156
A.2	A FEM element vs. a PNM element over the same material domain	157
A.3	Complete distinction between the numerical domain and the material domain	161
A.4	Linear triangular FNM elements are equivalent to a FEM triangular element.	162
A.5	Linear rectangular PNM cohesive elements	163
A.6	Modelling of a cohesive crack in a PNM element	164
A.7	Modelling of multiple cohesive cracks in a PNM element . .	164
A.8	Single cohesive crack modelling by the PNM	166

A.9	Uniform tensile loading of a small section of a cross-ply laminate	167
A.10	Importance of matrix crack tip representation on the laminate interface [49]	168
A.11	A PNM element representing a cross-ply laminate	169
A.12	Two other models for the modelling of a cross-ply section . .	169
A.13	Comparison of load-displacement curves of Abaqus FEM result, Abaqus PNM result and the extended PNM element result (UEL)	170
A.14	Comparison of failure patterns by three different models . .	171
A.15	Situations where the mapping between the FEM element and the PNM element does not exist	173
A.16	Comparison of FEM and PNM solutions in a generic case . .	174
B.1	Geometry of the cracked domain	178

List of Tables

2.1	Some numerical methods for representing discontinuities in solids.	27
3.1	IM7/8552 carbon/epoxy material properties for the open-hole laminates [56]	42
3.2	Sublamine-scaling thickness size effect	51
3.3	Ply-scaling thickness size effect	54
3.4	In-plane size effect of sublamine-scaled specimens	58
3.5	In-plane size effect of ply-scaled specimens	60
3.6	Parametric sensitivity analysis	63
4.1	Mechanical properties representative of carbon/PEEK [143] used for the DCB test.	108
4.2	Mechanical properties of toughened glass/epoxy composite .	111
4.3	Mechanical properties of AS4/3501-6 carbon/epoxy composite	115
A.1	Mechanical properties of a glass/epoxy composite (assumed)	165

Nomenclature

Abbreviations

4ENF		four point bend end-notched flexure
B-K		Benzeggagh and Kenane
C.E.		Cohesive Element
CDM		Continuum Damage Mechanics
CLT		Classical Lamination Theory
COH3D8	. . .	Abaqus 8-node 3D cohesive element
cohes. elem.	.	Cohesive Element
CSLI		Continuum Shell Laminate with Interface
CSPB		Continuum Shell Perfect Bonding
CT		Compact Tension
cv		coefficient of variation
DCB		Double Cantilever Beam
DoF		Degrees of Freedom
Elm		Element
FEM		Finite Element Method
FN, FNM	. .	Floating Node, Floating Node Method
PN, PNM	. .	Phantom Node, Phantom Node Method

S4RAbaqus 4-node conventional shell element with reduced in-
tegration

SC4RAbaqus 8-node continuum shell element with reduced in-
tegration

SIFStress Intensity Factors

SubSub-element

UELUser Element

VCCTVirtual Crack Closure Technique

WWFEWorld Wide Failure Exercise

WWFEIIthe second World Wide Failure Exercise

XFEMeXtended Finite Element Method

Lower-case Roman letters

a crack length

b thickness

$\mathbf{b}_j$ DoF for the Heaviside enrichment, $j \in J$

$\mathbf{c}_k^l$ DoF for the l th tip enrichment, $k \in K$

d diameter of the hole in open-hole laminates

$\mathbf{f}$ body force per unit volume

f failure criterion

fDOF_D array of internal floating DoF of an element

fDOF_j array of shared floating DoF on edge j of an element

h laminate half-thickness

h_1 thickness of the $[0_2]$ sub-laminate

h_2		thickness of the $[90_s]$ sub-laminate
i_{elem}		global index of an element
i_{node}		global index of a node
k		penalty stiffness used for the constraint between two DoF sets
l		length of the open-hole laminate
l_{ch}		cohesive zone length
$l_{\text{ch,II}}$		cohesive zone length under Mode II loading
$l_{\text{ch,slender,II}}$		cohesive zone length of a slender body under Mode II loading
ℓ_{CT}		length of the crack in the 2D element at the crack tip, used in VCCT
l_e		element characteristic length
ℓ_W		length of the crack in the 2D element in the wake of the crack tip, used in VCCT
m		ply-scaling ratio
n		sublaminate-scaling ratio
n_L		number of elements in the length direction
n_W		number of elements in the width direction
$\mathbf{n}(\bar{i}_{\text{node}})$		normal of the crack surface, stored in the dataset of floating node i_{node}
$\mathbf{n}(\bar{j})$		normal of the crack surface, stored in the dataset of edge j
n^{d}		number of double-blocked 0 plies in a laminate
n^{s}		number of single 0 plies in a laminate
$\mathbf{q}$		nodal displacement vector of a finite element

$\mathbf{q}_\alpha$		nodal displacement vector of the (sub-)element on Ω_α
$\mathbf{q}_{\text{CE}}$		nodal displacement vector of a cohesive element
$\mathbf{q}_i$		displacement vector of the node i , $i \in I$
$\mathbf{q}_j$		displacement vector of the real node j
$\mathbf{q}_{j'}$		displacement vector of the phantom node j'
$\llbracket \mathbf{q} \rrbracket$		displacement jump across the crack at the edge in the wake of the crack tip, used in VCCT
$\llbracket \mathbf{q} \rrbracket (\bar{i}_{\text{node}})$	. .	$\llbracket \mathbf{q} \rrbracket$ stored in the dataset of the floating node i_{node}
$\llbracket \mathbf{q} \rrbracket (\bar{j})$		$\llbracket \mathbf{q} \rrbracket$ stored in the dataset of edge j
$\llbracket q_n \rrbracket$		normal component of $\llbracket \mathbf{q} \rrbracket$
$\llbracket q_t \rrbracket$		tangential component of $\llbracket \mathbf{q} \rrbracket$
$\mathbf{q}^\circ$		DoF vector related to the standard shape functions in XFEM
$\mathbf{q}^*$		DoF vector related to the enrichment functions in XFEM
r		radial coordinate of a point in polar coordinates
rDOF		array of real DoF in an element
t		thickness of a single ply
$\mathbf{t}$		traction on the material boundary
$\mathbf{u}$		displacement vector of a point
$\mathbf{u}_\alpha$		displacement vector of a point in Ω_α
u_{eq}		equivalent displacement
$\mathbf{v}$		test function
w		width of the open-hole laminate

$\mathbf{x}$	physical coordinates of a point
$\mathbf{x}_\alpha$	physical coordinates of a point in Ω_α
$\mathbf{x}_{\Gamma_{\Omega_c}}$	nodal coordinate vector of the cohesive element on Γ_{Ω_c}
$\mathbf{x}_j$	physical coordinates of the real node j
$\mathbf{x}_{j'}$	physical coordinates of the phantom node j'
$\mathbf{x}_\Omega$	nodal coordinate vector of the finite element on Ω
$\mathbf{x}_{\Omega_\alpha}$	nodal coordinate vector of the (sub-)element on Ω_α
$\mathbf{x}_r, \mathbf{x}_s$	physical coordinates of crack tips

Upper-case Roman letters

A_{CT}	crack area in the element at the crack tip, used in VCCT
$A(\bar{i}_{\text{node}})$	crack area in the element in the wake of the crack, stored in the dataset of floating node i_{node}
$A(\bar{j})$	crack area in the element in the wake of the crack, stored in the dataset of edge j
A_W	crack area in the element in the wake of the crack tip, used in VCCT
$\mathbf{B}$	strain-displacement matrix of a finite element
$\mathbf{B}_\alpha$	strain-displacement matrix of the (sub-)element on Ω_α
B	mixed-mode ratio in the B-K formula
$\mathbf{D}$	constitutive tensor
$\mathbf{D}_{CE}$	constitutive tensor of a cohesive element
D_{LL}	longitudinal shear penalty stiffness of cohesive elements
$D_{LL}^{\min}$	minimum longitudinal shear penalty stiffness of cohesive elements

D_{nn}	 normal penalty stiffness of cohesive elements
$D_{nn}^{\min}$	 minimum normal penalty stiffness of cohesive elements
D_{TT}	 transverse shear penalty stiffness of cohesive elements
$D_{TT}^{\min}$	 minimum transverse shear penalty stiffness of cohesive elements
E	 Young's modulus
Edge	 array which contains the global indices of the edges of an element
E'_{II}	 equivalent elastic modulus of orthotropic materials, used for the estimation of cohesive zone length under Mode II loading
$E'_{II,slender}$	. . . equivalent elastic modulus of a slender body, used for the estimation of cohesive zone length under Mode II loading
E_i	 Young's modulus of the direction i , $i = 1, 2, 3$
$\mathbf{F}$	 internal force vector at the crack tip, used in VCCT
F_n	 normal component of $\mathbf{F}$
F_t	 tangential component of $\mathbf{F}$
G	 shear modulus
$\overline{G}_{fc}$	 average translaminar fracture toughness of 0 plies in a laminate
G_{fc}^d	 translaminar fracture toughness of double-blocked plies
G_{fc}^m	 translaminar fracture toughness of m -blocked plies
G_{fc}^s	 translaminar fracture toughness of single plies
G_I	 Mode I energy release rate
G_{Ic}	 Mode I critical energy release rate
G_{II}	 Mode II energy release rate

G_{IIc}		Mode II critical energy release rate
G_{ij}		shear modulus for the shear deformation between directions i and j , $ij = 12, 23, 13$
G_{mc}		mixed-mode critical energy release rate
G_{N}		tensile strain energy before crack propagatioin
G_{S}		shear strain energy before crack propagatioin
H		Heaviside function
I		set of all nodes
J		set of nodes enriched by the Heaviside function
$\mathbf{J}$		Jacobian matrix of a finite element
$\mathbf{J}_{\alpha}$		Jacobian matrix of the (sub-)element on Ω_{α}
$\mathbf{J}_{\text{CE}}$		Jacobian matrix of a cohesive element
K		set of nodes enriched by the tip-enrichment functions
$\mathbf{K}$		stiffness matrix of a finite element
$\mathbf{K}_{\alpha}$		stiffness matrix of the (sub-)element on Ω_{α}
$\mathbf{K}_{\text{CE}}$		stiffness matrix of a cohesive element
K_{I}		Mode I stress intensity factor
K_{II}		Mode II stress intensity factor
L		Length
$\mathbf{N}$		shape function matrix of a finite element
$\mathbf{N}_{\text{CE}}$		shape function matrix of a cohesive element
N_{edge}		number of edges in an element
N_{fDOF_D}		number of internal floating DoF of an element

N_{fDOF_j}		number of shared floating DoF on edge j of an element
$N_{i/j/k}$		shape function of node i or j or k , $i \in I$, $j \in J$, $k \in K$
N_{node}		number of nodes in an element
Node		array which contains the global indices of the nodes of an element
N_{rDOF}		number of real DoF in an element
$\mathbf{Q}$		nodal force vector of a finite element
$\mathbf{Q}_\alpha$		nodal force vector of the (sub-)element on Ω_α
S_{L}		longitudinal shear strength for matrix cracking
S_{T}		transverse shear strength for matrix cracking
W		Width
W, CT		crack wake and crack tip elements, respectively
R, T		refinement and transition elements, respectively
X_{t}		fibre-directional tensile strength
Y_{c}		transverse compressive strength
Y_{t}		transverse tensile strength
$Y_{\text{t}}^{\text{ist}}$		In situ transverse tensile strength
Y_{t}^{UD}		transverse tensile strength of a unidirectional lamina

Lower-case Greek letters

δ		vector of separation of a cohesive crack
δ_{n}		normal separation of a cohesive crack
δ_{L}		longitudinal shear separation of a cohesive crack
δ_{T}		transverse shear separation of a cohesive crack

$\boldsymbol{\epsilon}$	strain tensor
$\boldsymbol{\epsilon}_\alpha$	strain tensor of the (sub-)element on Ω_α
ϵ	strain; user-defined non-positive number, used in the propagation criterion of the FNM
ϵ_n	normal strain on the potential fracture plane
ϵ_2^f	transverse failure strain of a composite lamina
η	exponent of the mixed-mode ratio B in the B-K formula
γ_L	longitudinal shear strain on the potential fracture plane
γ_T	transverse shear strain on the potential fracture plane
$\mu(i_{\text{node}})$	edge status variable for the edge containing the floating node i_{node}
$\mu(j)$	edge status variable for edge j
μ_L	frictional parameter for longitudinal shear on the matrix crack plane
μ_T	frictional parameter for transverse shear on the matrix crack plane
ν	Poisson's ratio
ν_{ij}	Poisson's ratio for the contraction in direction j with tension in direction i , $ij = 12, 21, 23, 32, 13, 31$
ϕ	angle between the matrix fracture plane and the through-the-thickness direction of the lamina
ϕ_o	ϕ of the lamina when it is under pure compression in the in-plane transverse direction
$\boldsymbol{\psi}_{\text{tip}}$	vector of tip enrichment functions
ψ_{tip}^l	l th tip enrichment function

σ	applied stress; remote loading
$\boldsymbol{\sigma}$	stress tensor
σ_n	normal stress on the potential fracture plane
σ_L	longitudinal shear stress on the potential fracture plane
σ_T	transverse shear stress on the potential fracture plane
σ_{eq}	equivalent stress on the potential fracture plane
σ_0	Material strength
τ_n	normal traction for delamination
τ_n^c	normal strength for delamination
τ_L	longitudinal shear stress for delamination
τ_L^c	longitudinal shear strength for delamination
τ_T	transverse shear stress for delamination
τ_T^c	transverse shear strength for delamination
$\boldsymbol{\tau}_c$	traction on the cohesive surface
θ	the angular coordinate of a point in polar coordinates
θ	orientations of the crack with respect to the horizontal direction in the slanted crack model
ξ	natural coordinates of a point
$\xi(i_{\text{node}})$	natural coordinates of the crack on the edge containing the floating node i_{node}
$\xi(j)$	natural coordinates of the crack on edge j

Upper-case Greek letters

Γ_Ω	boundary of the original material domain
---------------------------	--

Γ_{Ω_c}		surface of a cohesive crack
Γ_{Ξ}		boundary of the integration domain in natural coordinates for Ω
$\Gamma_{\Xi_{\alpha}}$		boundary of the integration domain in natural coordinates for Ω_{α}
Γ_{Ξ_c}		integration domain in natural coordinates for Γ_{Ω_c}
Ω		original material domain
Ω_{α}		material domain partition $\alpha, \alpha = A, B, C, D, E, F$
Ξ		integration domain in natural coordinates for Ω
Ξ_{α}		integration domain in natural coordinates for Ω_{α}

Operators

$\mathcal{L}_{\mathbf{x}}$		differential operator which applies on $\mathbf{u}$ to obtain $\boldsymbol{\epsilon}$
$\mathcal{L}_{\boldsymbol{\xi}}$		differential operator which calculates the derivatives of $\mathbf{N}$ in natural coordinates
$\llbracket \bullet \rrbracket$		jump of a function over an interface in its domain
$\langle \bullet \rangle_+$		the Macaulay brackets

Chapter 1

Introduction

Fibre-reinforced composites are typically composed of continuous reinforcement fibres (made of carbon or glass, etc.) embedded in a matrix material (such as epoxy). They are usually fabricated into thin layers (Figure 1.1a), which are often referred to as plies. Let (x, y, z) represents the global coordinate system, and $(1, 2, 3)$ represent the local coordinate system of each ply, where the local direction 1 represents the fibre direction (sometimes referred to as the translaminar direction), local direction 2 is the in-plane transverse direction and local direction 3 is the normal of the laminate shell or the out-of-plane transverse direction (Figure 1.1a). It is generally assumed that local direction 3 coincides with the global direction z . A ply of angle θ indicates that the fibres of this ply are oriented at angle θ from the global direction x (Figure 1.1b). A laminate is composed of plies of different angles; they form the lay-up of the laminate which is represented by the bracket notation (Figure 1.1c). Some general rules of this notation are presented below:

- $[0/45/90]$ represents a laminate which has three plies with angles 0, 45 and 90, respectively;
- a numerical subscript represents the number of same-angled plies grouped together in the laminate; for example, $[0/45/90_2]$ indicates that two adjacent 90 plies are grouped together;
- a subscript s indicates that the laminate is symmetric about the mid-plane, e.g., $[0/45/90/90/45/0]$ is written as $[0/45/90]_s$;

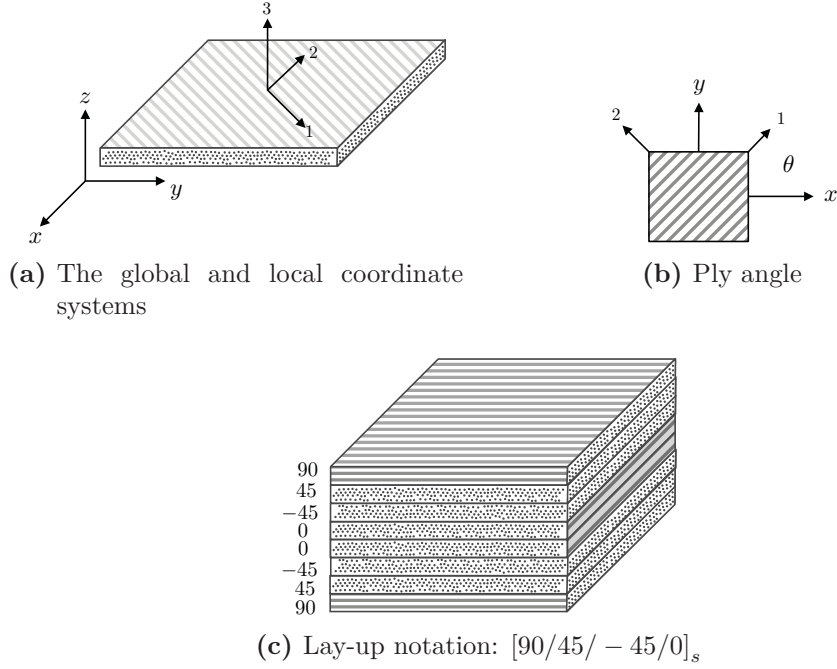


Figure 1.1: Coordinates and notations of composites

- $[0/45/-45/90]$ is written as $[0/\pm 45/90]$;
- $[0/45/0/45/90]$ is written as $[(0/45)_2/90]$ where repeated sub-laminate is written in the brackets with a numerical subscript indicating the number of times it is repeated;
- similarly as above, $[0/45/-45/45/-45/90]$ is written as $[0/(\pm 45)_2/90]$; and
- $[0/45/90/0/45/90/90/45/0/90/45/0]$ is written as $[0/45/90]_{2s}$, where the symmetric laminate is formed with repeated sub-laminates on both side of the mid-plane.

Fibre-reinforced composite materials possess considerable advantages over traditional metallic materials on both weight and performance. Recently, composites are replacing more and more traditional metallic materials in industrial applications. However, optimal design with composites is often not achieved due to uncertainties in the failure prediction. Despite decades of research, accurate failure prediction of composites remains a challenging topic. Unlike traditional metallic materials, the failure process of composites involves several distinctive failure mechanisms which

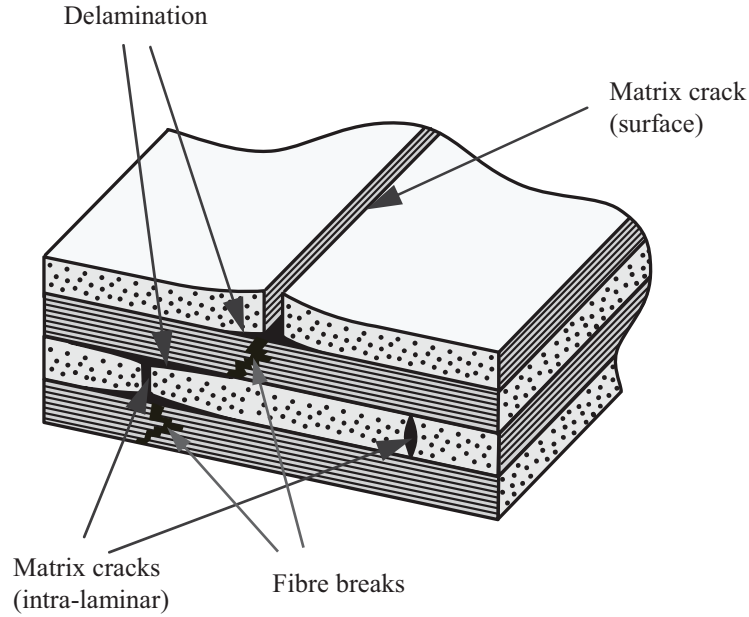


Figure 1.2: Common failure modes in a composite laminate (image modified from the original image in [66]).

often coexist and interact with each other to form complex fracture paths. In many cases, the high count of damage and failure of different driving mechanisms, as well as their interactions, poses great challenges for both the failure theories and numerical methods.

1.1 Overview of the failure mechanisms of composites

The failure of a composite laminate is often preceded by a combination of different mechanisms, such as fibre breaking, matrix cracking and delamination (Figure 1.2 and 1.3). Fibre breaks generally lead to the complete failure of the structure as they are the main load-bearing elements. Matrix cracks are usually composed of two failure mechanisms at the scale of the constituents (often referred to as the micro-scale), namely the fibre-matrix debonding and the micro-cracks within the epoxy resin (Figure 1.4a); they often join up and form matrix cracks whose length scale is comparable to

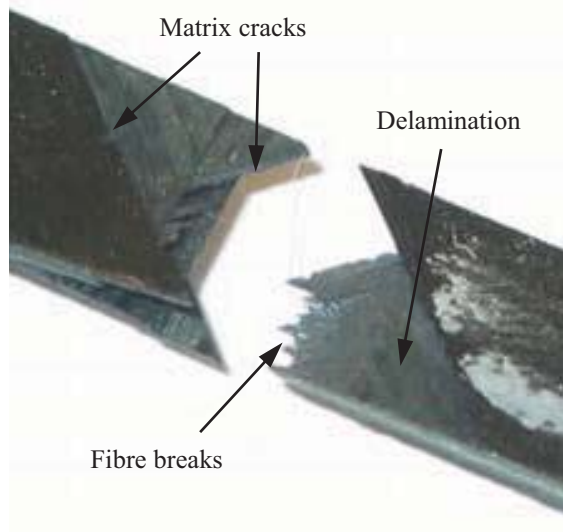


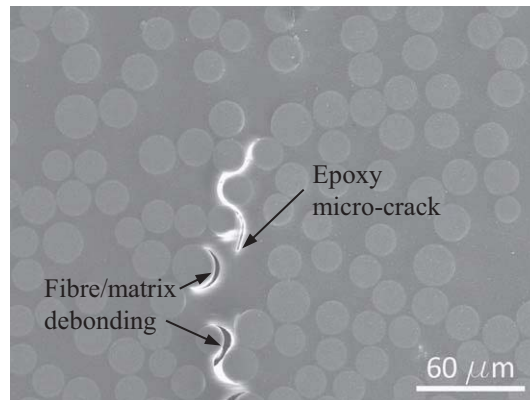
Figure 1.3: Failure modes in an experiment from [155].

that of the plies (Figure 1.4b). Delamination occurs between plies of different orientations. It can be triggered by direct out-of-plane loading on the laminate (Figure 1.5a), or by matrix cracks (Figure 1.5b).

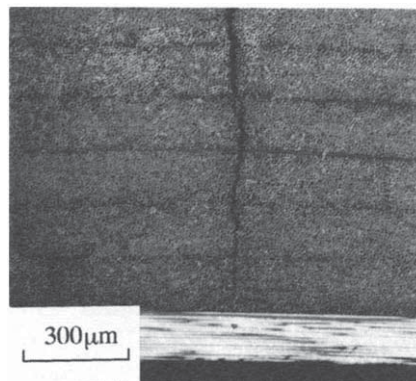
The above-mentioned failure mechanisms may not occur independently; they often interact with each other to form complex fracture paths within the laminate. Figure 1.5b shows an example of the matrix crack/delamination interaction for a cross-ply specimen under tensile loading; delamination initiated after the matrix crack reaching the 90/0 interface [137]. The next section will introduce an example, the open-hole tension problem, which demonstrates clearly the complex interacting mechanisms in a laminate, and reveals the effect of the specimen size on its failure.

1.2 Introduction of the open-hole tension size effects

Actual composite structures often contain holes which dictates the locations of failure initiation. As a result, the strengths of open-hole composite laminates are often used as limiting design criteria for actual composite structures. Therefore, the accurate predictions of the failure modes

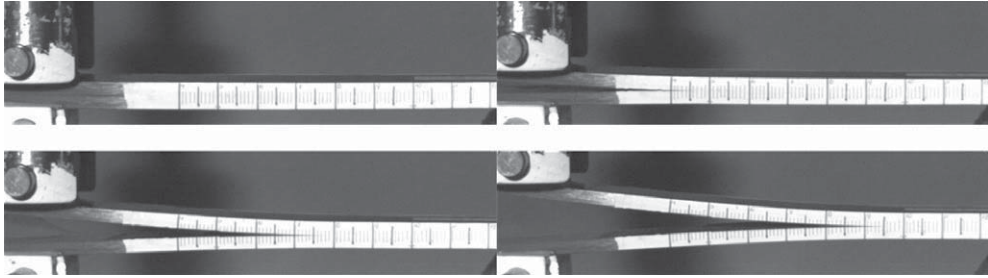


(a) A photograph from [30] showing fibre-matrix debonding and micro-cracking in the epoxy resin.

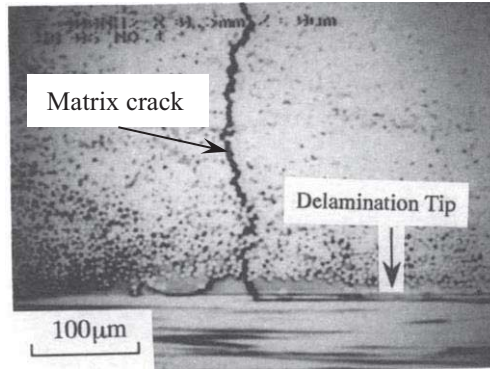


(b) Experimental image from [137] showing a matrix crack spanning over the thickness of several plies.

Figure 1.4: Matrix crack



(a) A series of snapshots of the experiment in [27] showing delamination growth under out-of-plane loading.



(b) Experimental image from [137] showing delamination propagating from the tip of a matrix crack.

Figure 1.5: Delamination

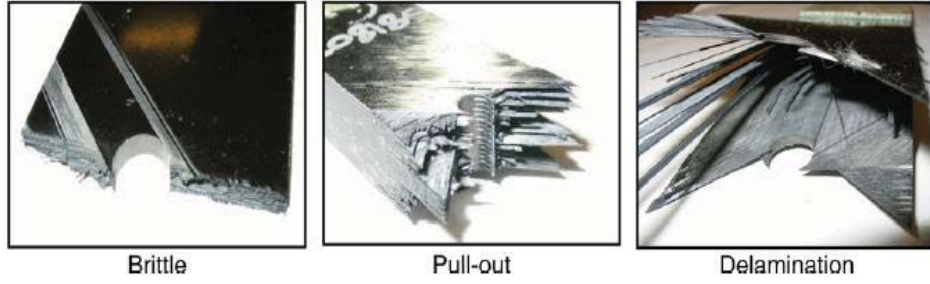


Figure 1.6: Three failure modes in open-hole tension experiments [53]

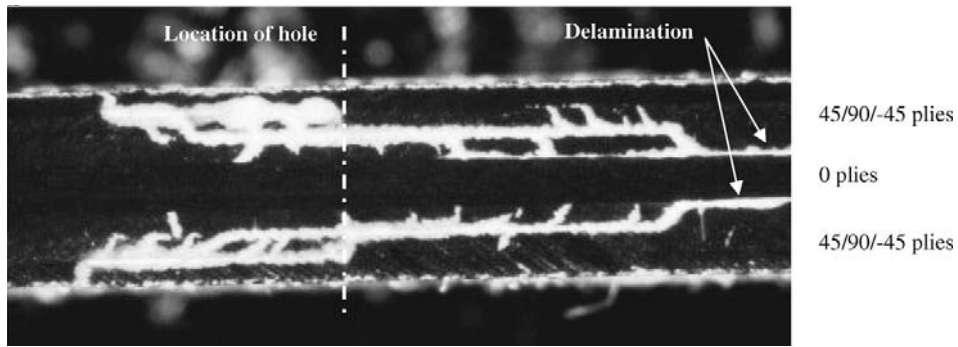


Figure 1.7: Edge-view image of an open-hole tension experiment in [53] showing complex fracture paths involving interactions between matrix cracks and delamination.

and strengths of open-hole composite laminates are particularly important. Green et al. [53] performed a series of detailed experiments on open-hole quasi-isotropic carbon-epoxy composites. Different scaling methods were applied in the experiment on the in-plane dimension, thickness dimension and ply lay-ups of the laminate. Clear size effects on the laminate strengths were recorded and distinctive dominant failure modes such as laminate brittle failure, ply pull-out, and delamination were observed, as shown in Figure 1.6. Complex interactions between different failure mechanisms were present (Figure 1.7 and 1.8).

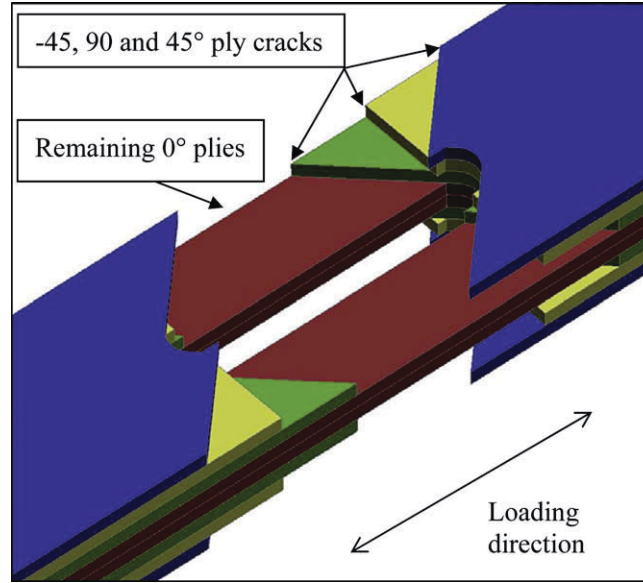


Figure 1.8: Schematic drawing of the interaction between different failure mechanisms [154].

1.3 Introduction of the thickness-dependence of translaminar fracture toughness

While the size effect on strength is not surprisingly new (see early reviews in [131–133, 152]), a size effect on fracture toughness of composites has been recently recorded by Laffan et al. [79]. Compact Tension (CT) experiments on pre-cracked cross-ply laminates were performed to determine the mode I translaminar (fibre direction) tensile fracture toughness of T300/920 carbon-epoxy composites. Their work reports large variations of the translaminar fracture toughness with respect to the thickness of the 0 ply-blocks in the tested specimens. The amount of fibre pull-out in the 0 plies of the $[(90/0_2)_8/90]_s$ laminate increases compared to that of the $[(90/0)_8/90]_s$ laminate which causes a significant increase of energy dissipation (Figure 1.9). The propagation translaminar fracture toughness of the 0 plies in $[(90/0_2)_8/90]_s$ is measured to be 132 kJ/m^2 , while in $[(90/0)_8/90]_s$ the translaminar fracture toughness is between 57 and 69 kJ/m^2 [79]. The change in translaminar fracture toughness with size is of

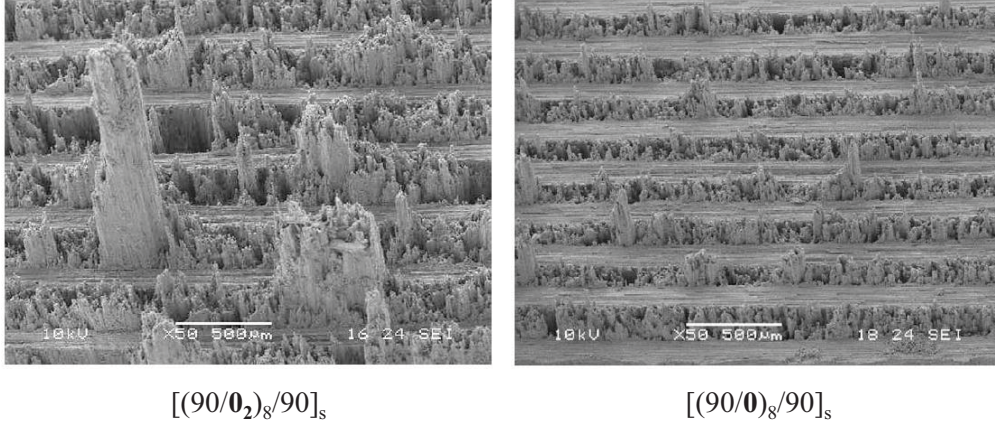


Figure 1.9: Fractographic images of the translaminal fracture surface [79]

$\approx 100\%$ with only a factor of two in the linear dimension (thickness). Another set of experiment in [82] examined the effect of 0 ply-block thickness on the structural-scale sub-critical damage development prior to 0 ply fracture and it showed that ply-blocking has a toughening effect on the global load-response of the laminate due to the larger extent of structural-scale matrix splitting and delamination which effectively relax the notch stress concentration.

The two experimental work in [79] and [82] show physics at different scales of the laminate and it is important that they both be included in the modelling of composites. Although the sub-critical damage such as matrix splitting and delamination have been commonly included in the modelling of composites [2, 56, 128, 135, 146], the thickness-dependence of translaminal fracture toughness has not been employed in any numerical modelling of composites and its importance on the failure predictions of composite structures remains to be examined.

1.4 Brief review of the failure theories

The prediction of failure in composites has been proven challenging, largely due to the complexity of the physics behind, as shown in the above three sections. The failure theories of composite materials have been extensively researched in the past. Early research in this area has mainly focused on

finding a criterion that determines the initiation of failure. A good review and comparison of these criteria can be found in the papers on the World Wide Failure Exercise (WWFE) [62, 125]. Criteria such as Puck [111] and Tsai-Wu [140] have shown some good predictions for the failure of a single lamina under two-dimensional loading. More complex failure theories that consider both the initiation and the propagation of different failure mechanisms under more complex loading conditions have been established by Dávila et al. [41], Pinho et al. [104] and Vyas et al. [148]. Recently, a second World Wide Failure Exercise (WWFEII) was organized where a selection of test cases on various materials ranging from pure epoxy resin to multi-angled laminate tubes were performed [61]. All of the tests were on relatively simple structures without notches or holes and delamination was not studied. Some state-of-the-art failure theories were compared and reviewed [70, 71]. A few physically-based theories such as Carrere [31], Pinho [104, 106, 108] and Puck [45] have shown good predictions on the majority of the test cases. These researches indicate that some of the existing failure theories have already reached a high level of fidelity in predicting the failure of unidirectional and multi-directional composite structures of relatively simple geometry. For complex problems such as the open-hole tension experiments by Green et al. [53], failure predictions have been performed by Camanho et al. [29], Hallett et al. [56], Abisset et al. [2], Song et al. [128], van der Meer et al. [146], and Swindeman et al. [135] using different failure theories and numerical methods; they achieved varied degrees of success. However, to the author's knowledge, no numerical work has successfully reproduced all of the main experimental observations, except the one by Hallett et al. [56] which uses a post-processed statistical failure theory for fibre breaking and employs a priori knowledge of the potential fracture paths in the creation of FEM meshes. A generic model (without the knowledge of potential fracture locations) which accurately predicts all of the experimental data and observations remains to be established.

1.5 Objectives of the research

This research aims to find an accurate model for composites which is capable of capturing the critical failure mechanisms and their interactions

in composite laminates, such as those shown in the open-hole tension experiments in [53] and the matrix crack/delamination interaction shown in Figure 1.5b. Such a model should incorporate:

- physically-based failure theories which predict accurately the initiations of different failure mechanisms in the material, and describe correctly the post-failure behaviours of the material, at the scales of modelling;
- the right combination of numerical tools that adequately represent the kinematics of the material during all stages of loading;
- the right model inputs.

Since several physically-based failure theories have already been established in the literature and moreover, demonstrated good performance in many test cases [70, 71], this research instead focuses on finding the right numerical tools for the modelling of failure in composites; it also examine the significance of a new model input, i.e., the thickness dependence of the translaminar fracture toughness. Specifically, the objectives of this research are:

- examine the capabilities and limitations of some commonly-used numerical methods on the open-hole tension problem in [53], using an existing failure theory;
- examine the importance of the thickness dependence of translaminar fracture toughness [79] on the failure predictions of open-hole composite laminates in [53];
- find, and if necessary, establish the right numerical tools to model accurately the different failure mechanisms and their interactions in composite laminates.

1.6 Structure of the thesis

In the rest of the thesis, different numerical methods for the modelling of damage and failure of composites will be reviewed in Chapter 2; Chapter 3

will then present the numerical study of the size effects of open-hole tension composite laminates [53]; based on the findings in Chapter 3, Chapter 4 will propose a novel numerical method which possesses several advantages in the modelling of discontinuities over existing numerical methods, and show the application of the method on an example involving extensive matrix crack/delamination interactions.

Chapter 2

Literature review of numerical methods for the failure modelling of composites

2.1 Introduction

The standard Finite Element Method (FEM) has achieved tremendous success in many areas of application, where analytical solutions of the problems are difficult to be obtained. However, the assumptions of FEM are violated once discontinuities appear within elements, although cracks occurring at element boundaries can still be handled using conventional FEM (e.g., through node duplication or remeshing). The polynomial shape functions cannot reproduce the discontinuous solution field associated with the discontinuities. This limitation of the FEM is not significant when only the first occurrence of failure is interested, whereby the modelling of the material behaviour afterwards is not needed and the analysis can effectively be stopped; thus the stress analysis is sufficient to predict the failure of the structure. In the case when crack propagations are to be modelled, methods which are capable of modelling evolving discontinuities are required to represent the progressive damage and failure of the structure. The rest of this chapter reviews some of the most commonly used methods for this purpose, namely the remeshing method, the stiffness degradation method, the cohesive element, the smeared crack formulation, the eXtended Finite

Element Method (XFEM), and the Phantom Node Method (PNM).

For the rest of this thesis, a strong discontinuity is defined as a jump in the displacement field. A weak discontinuity is defined as a jump in the strain field.

2.2 Remeshing

When crack growths need to be represented in a FE mesh, a natural solution is to re-apply the FEM on the new material domain which includes the new growths of cracks, i.e, the remeshing method. Among the different remeshing strategies, the h -adaptive remeshing [10], where the element formulations remain the same but the element locations, sizes and shapes are modified (often according to an *a posteriori* error estimator) to represent the new geometry, is the most commonly used approach for modelling evolving discontinuities [9, 26, 74, 75, 93–95, 101, 122–124, 134]. In h -adaptive remeshing (herein referred to as remeshing for brevity), discontinuities can only be propagated along the element edges. A typical remeshing procedure is shown in Figure 2.1 which was proposed in [134]. When modelling the propagation of a crack using remeshing, the geometrical description of the domain is updated to take into account the new crack boundaries, and part (as in local remeshing) or the whole (as in global remeshing) of the mesh is deleted; a new element-size requirement is determined based on an *a posteriori* error estimator; the mesh generator is called to regenerate the missing part of the mesh according to the new geometrical description and element-size requirement (the so-called delete-and-refill approach); then the element connectivities are updated and field variables transferred from the old mesh to the new one; and this process repeats until the loading finishes or discontinuities stop to grow. Remeshing can lead to an accurate representation of discontinuities with minimal element distortion and optimal mesh density; however, extra databases are needed to store and update the geometry and field variables; and frequent geometry updates and mesh regenerations are computationally expensive.

In the case of composite materials, extensive matrix cracks and delamination may develop well before the final collapse of the material [53, 155]; updating the geometry, mesh and element connectivities with respect to

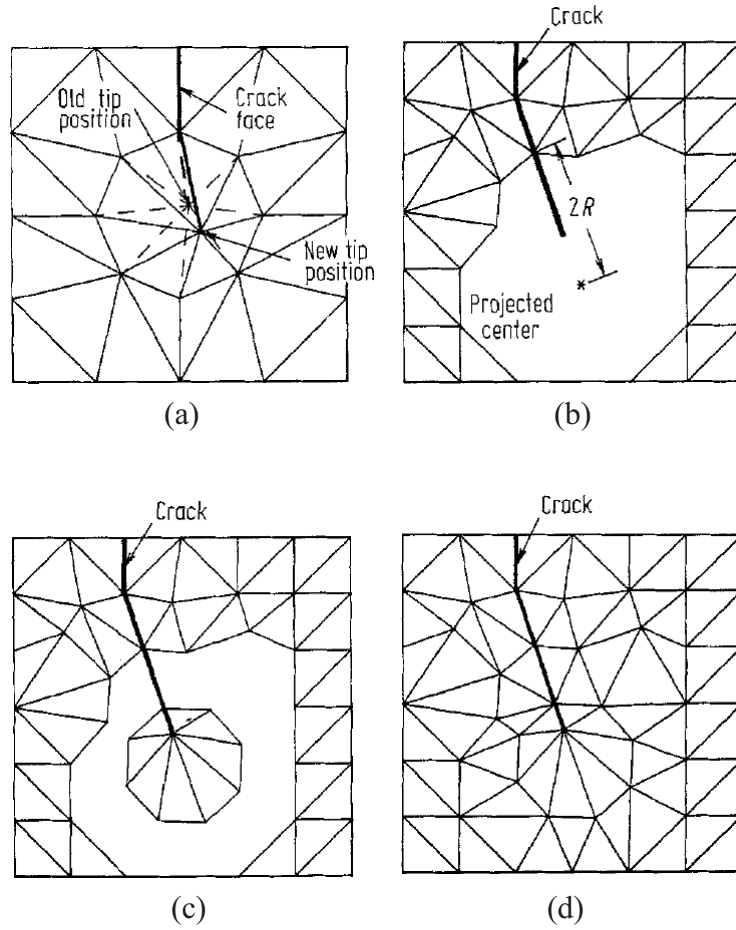


Figure 2.1: In remeshing as shown in [134], the advance of a crack in a mesh (a) is done in three steps: (b) removal of elements within a projected radius of the new crack tip; (c) forming quarter-point elements around the new crack tip; (d) meshing of the rest of the region [134].

all the crack growths in different layers can be complicated. Therefore, methods which can efficiently model a large number of crack propagations without requiring extensive modifications of the global quantities (geometry, mesh, and connectivities, etc.) are preferred for the progressive damage and failure analysis in composites.

2.3 Stiffness degradation method

The stiffness degradation method models the propagation of a crack through an element by simply degrading the relevant stiffness properties of this element, either to certain percentage of the original stiffness or according to a softening curve, both independent of length-scales. The initiation of failure is usually determined by a strength failure criterion. This method does not have any length-scale in the formulation; failure is entirely determined by local quantities such as local stress and strain in the element. Previous research in the literature has shown that this method suffers from mesh dependency on strength predictions for structures with stress-concentrated areas such as notches and holes [13, 29]. The predicted strength decreases with an increasing mesh-refinement. This is due to the fact that a finer mesh around a notch or a hole would predict a higher stress which would lead to an earlier fulfilment of the failure criterion. In the limit of infinitesimal element size, failure would occur with infinitesimal strain energy dissipation, so long as the stress or strain state of the element satisfies the failure criterion. This gradual loss of strain energy dissipation during fracture upon mesh refinement is globally reflected by a gradual decrease of structural strength, i.e., the mesh dependency of the strength prediction. This deficiency limits the applicability of the method in the quantitative failure analysis of composites.

2.4 Cohesive element

The cohesive element is a type of element formulated based on the cohesive zone theory. The cohesive zone theory was first introduced by Dugdale [48] and Barenblatt [12] and was further extended and then applied in FEM

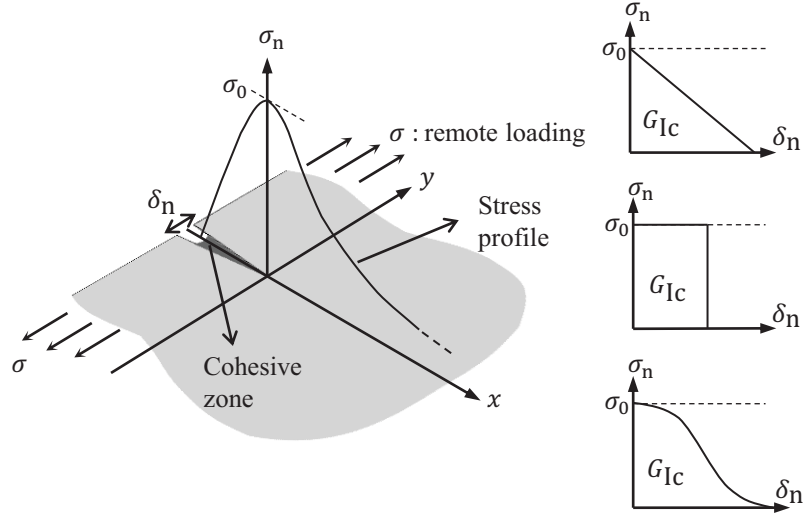
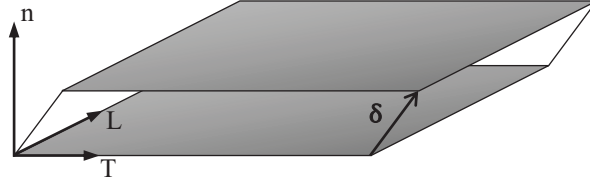
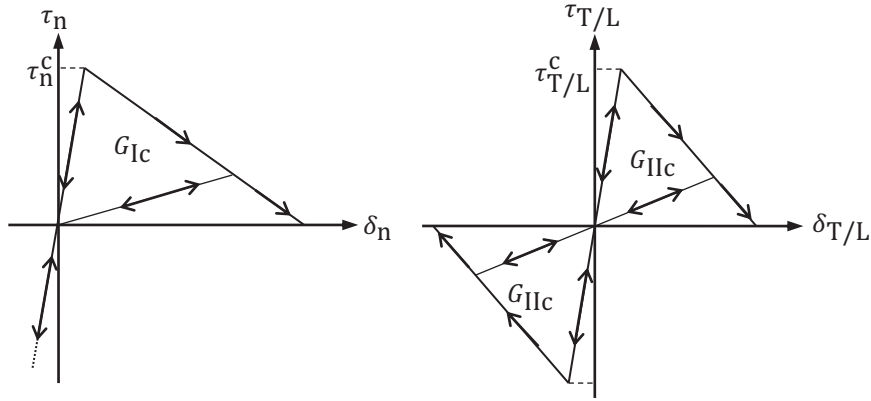


Figure 2.2: Left: a general stress profile of the cohesive zone under Mode I loading. Right: some representative cohesive laws describing the normal traction in the cohesive zone, σ_n , with respect to the normal separation of the crack surfaces, δ_n ; the area under the curve equals the value of the Mode I critical energy release rate of the material, G_{Ic} ; σ_0 is the material strength.

analysis by Hillerborg et al. [60]. The basic assumption of the theory is that there exists a cohesive zone between two new crack surfaces, such that the traction between them is non-zero, but in general decreases with separation (Figure 2.2). Unlike the strength-based failure theory used in the stiffness degradation method, the cohesive zone theory includes a length-scale parameter in its formulation. The per-area energy dissipation between two crack surfaces in the cohesive zone upon reaching complete (stress-free) separation must equal to the critical energy release rate of the material. In this way, a consistent energy dissipation during fracture is reached with respect to mesh refinement for numerical methods based on the cohesive zone theory, so that they do not suffer from the same strength mesh dependency associated with the stiffness degradation method. A large amount of work on cohesive zone theory is present in the literature. In general, numerical methods using cohesive zone theory can be classified into two categories: the continuum (smeared crack) method and the discrete (cohesive element) method. The former method will be reviewed in the next section.



(a) Topology of a cohesive element; δ is the vector of separation between the two initially-coinciding surfaces of the element.



(b) Constitutive laws of a cohesive element; τ_n, τ_T and τ_L are the normal stress and the two shear stresses on the interface; and τ_n^c, τ_T^c and τ_L^c are the normal strength and the two shear strengths; δ_n, δ_T and δ_L are the normal separation and the two shear separations; G_{Ic} and G_{IIc} the Mode I and II critical energy release rates of the interface.

Figure 2.3: A typical cohesive element formulation [103].

The topology of a 3D cohesive element is shown in Figure 2.3a. It is typically composed of a top and a bottom surface, which coincide with each other before loading. The traction between the two surfaces relates to their separation through the constitutive relationships employed in the formulation of the element (Figure 2.3b). Detailed formulations of cohesive elements can be found in [28, 103, 158]. Cohesive elements are able to give accurate predictions when the potential crack paths are known a priori; in such cases, cohesive elements can be inserted along the crack paths during preprocessing [56]. Therefore, cohesive elements are often used for the modelling of delamination [3, 28, 56, 58, 81, 103, 119, 141, 142]. Reviews of the development and applications of cohesive elements can be found in [157] and [153]. Study of the shape of cohesive laws has been performed in [3, 103, 119]. Both linear softening law and exponential softening law have been shown to give accurate predictions during static loading, with the former being more economical than the latter in terms of computation [3]; however, linear softening law has been found to be highly unstable during dynamic loading [103]. Viscosity regularization has been used in [52] to suppress the softening instability of cohesive elements. In this study, the loading conditions are all static, and linear softening law is used for cohesive elements (see Chapter 3 section 3.2.4); no viscosity regularization is employed. Discussions of element size and cohesive zone length are available in [58, 112, 141, 157]. A well-established finding on cohesive elements in the literature is that, in order to correctly capture the stress gradient in the cohesive zone and predict accurately the cohesive crack growth, the size of the cohesive element must be smaller than the cohesive zone length. A thorough investigation on the estimation of cohesive zone length has been performed in [55], where a formula has been proposed to determine the cohesive zone length and the maximum size of cohesive elements. Optimal values for strengths and stiffness of the cohesive elements have been studied in [55, 141, 142], which showed that using reduced strength values within acceptable ranges could lead to more accurate predictions [55, 141]. The penalty stiffness of the cohesive element must be high enough compared to those of the ply so as to simulate a perfect bonding between plies before delamination onset [141]. The penalty stiffness corresponding to different delamination modes can be related through the respective strength ratios

and fracture toughness ratios [142].

2.5 Smeared crack formulation

While cohesive elements have demonstrated good performance, they require the knowledge of the potential crack path before analysis. This requirement limits the applicability of the cohesive element in the case where the crack path cannot be determined a priori [43]. The smeared crack formulation was firstly developed by Bažant and Oh [16] (then named the crack band theory) for modelling the Mode I fracture of concretes. It provides an approximate representation of a cohesive crack, by smearing it over the domain of the element so that a degradation of the element stiffness suffices to describe the actual effect of the crack (Figure 2.4). The difference between the smeared crack formulation and the stiffness degradation method is that the former assumes the cohesive zone theory (Figure 2.2) as the constitutive behaviour of the crack in an element. The cohesive zone theory enforces a consistent fracture energy dissipation upon failure of the element, so that a length-scale parameter is embedded in the element formulation (Figure 2.4); this helps to avoid the spurious mesh dependency in strength predictions associated with the stiffness degradation method [16, 29, 105].

The smeared crack formulation was firstly employed in the modelling of composites for fibre breaking and matrix cracking by Pinho et al. [105] and demonstrated good performance. Smeared crack formulations are numerically efficient, robust and in general capable of modelling complex crack patterns, which makes it a suitable modelling method for composites. However, the smeared crack formulation has been shown to have mesh dependency at predicting the fracture paths, i.e., the predicted fracture path was strongly influenced by the structure of the mesh, in the modelling of concretes [43] and of a single composite ply [128, 144]. Studies on composite laminates using the smeared crack formulation have showed good levels of success in [29, 39, 108] and some limitations in [146].

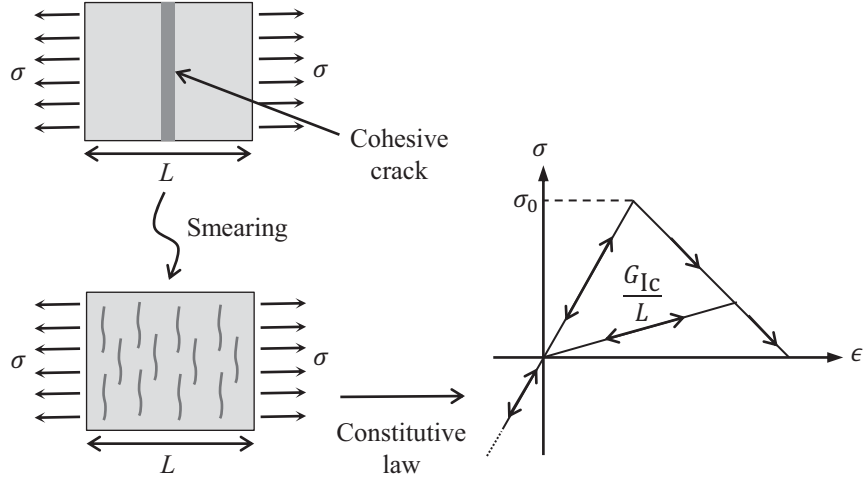


Figure 2.4: Smeared crack formulation; σ is the applied tension and ϵ is the corresponding strain; σ_0 is the material strength; L is a characteristic element length [105]; G_{Ic} is the Mode I critical energy release rate.

2.6 eXtended Finite Element Method

XFEM was firstly developed by Belytschko and Black [21], Moës et al. [92] based on the partition of unity theory by Melenk and Babuška [88]. XFEM introduces extra degrees of freedom (DoF) and enrichment functions to approximate the displacement fields in the neighbourhood of a crack. Nodes near the crack are divided into two groups according to their locations with respect to the crack (Figure 2.5). Nodes in the elements completely cut by a crack are enriched by the step function H (generally referred to as the Heaviside enrichment) to represent the displacement jump, and nodes in the elements containing a crack tip are enriched by the known (analytical) displacement solutions ψ_{tip} which are called tip-enrichment functions.

The step function used for the Heaviside enrichment is defined as:

$$H(\mathbf{x}) = \begin{cases} 1 & \text{for } \mathbf{x} \text{ on one side of the crack;} \\ -1 & \text{for } \mathbf{x} \text{ on the other side of the crack.} \end{cases} \quad (2.1)$$

where $\mathbf{x}$ indicates the physical coordinates of a point.

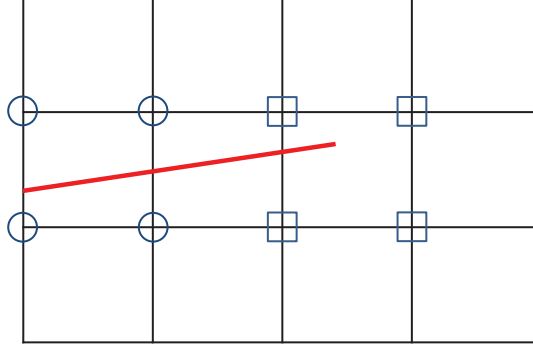


Figure 2.5: A typical XFEM representation of a crack (red line) in a mesh. Circled nodes belong to set J and they are enriched by the Heaviside function; squared nodes belong to set K and they are enriched by the tip-enrichment functions.

The XFEM approximated displacement solution $\mathbf{u}$ is generally expressed as the following [92]:

$$\mathbf{u}(\mathbf{x}) = \underbrace{\sum_{\forall i \in I} N_i(\mathbf{x}) \mathbf{q}_i}_{\text{FEM approximation}} + \underbrace{\sum_{\forall j \in J} N_j(\mathbf{x}) H(\mathbf{x}) \mathbf{b}_j}_{\text{Heaviside enrichment}} + \underbrace{\sum_{\forall l} \sum_{\forall k \in K} N_k(\mathbf{x}) \psi_{\text{tip}}^l(\mathbf{x}) \mathbf{c}_k^l}_{\text{tip-enrichment}} \quad (2.2)$$

where I is the set of all nodes, J is the set of nodes enriched by the Heaviside function and K is the set of nodes enriched by the tip-enrichment functions; N_i , N_j and N_k are respectively the shape functions corresponding to the node $i \in I$, $j \in J$ and $k \in K$; $\mathbf{q}_i$ is the FEM nodal displacement vector of node $i \in I$; $\mathbf{b}_j$ is the nodal DoF of node $j \in J$ associated with the Heaviside enrichment function; $\mathbf{c}_k^l$ is the nodal DoF of node $k \in K$ associated with the l th tip-enrichment function. For 2D linear elastic isotropic material:

$$\boldsymbol{\psi}_{\text{tip}}(\mathbf{x}) = \left\{ \sqrt{r} \sin\left(\frac{\theta}{2}\right), \sqrt{r} \cos\left(\frac{\theta}{2}\right), \sqrt{r} \sin\left(\frac{\theta}{2}\right) \sin(\theta), \sqrt{r} \cos\left(\frac{\theta}{2}\right) \sin(\theta) \right\} \quad (2.3)$$

where r and θ are the polar coordinates of a point, in the region close to the crack tip.

When crack propagation is studied, remeshing is avoided by progressively enriching new nodes affected by the crack. Therefore, cracks in XFEM are kinematically well represented. The enrichment functions in 2.3 are only valid for homogeneous isotropic materials containing a stress-

free crack under the assumptions of the linear elastic fracture mechanics. Different enrichment strategies should be used for weak discontinuities, for cohesive cracks, and for more complex materials such as orthotropic materials and bi-material interfaces. The cohesive segments method [73, 116, 117] models multiple cohesive cracks within existing elements with multiple Heaviside enrichment functions and additional DoF at the original nodes. The regularized XFEM [64, 65, 135] replaces the Heaviside enrichment with a series of B-spline basis functions such that standard Gauss integration methods can be used; no special integration technique such as the Delaunay triangulation is needed. Although modelling a large number of interacting cracks can prove challenging for XFEM, it has been used successfully to predict complex crack scenarios typical in composite materials; for instance in the context of delamination modelling [7, 97, 129, 130], micro-scale modelling [63], multi-scale modelling [59] and matrix crack/delamination interactions [65, 73, 135]. Comprehensive reviews of XFEM can be found in [23, 51]. The implementation of XFEM in existing FEM packages is not straightforward. Numerical integration over cracked elements typically requires special integration techniques such as the Delaunay triangulation. The mapping of cracks (between physical and natural coordinates) in quadrilateral elements (or elements with non-constant strain field) cannot be integrated exactly because a straight crack in physical coordinates is generally curved in the natural coordinates [51].

2.7 Phantom Node Method

Hansbo and Hansbo [57] proposed modelling discontinuities within an element by using additional DoF at each node and two sets of nodal shape functions. Each set of shape functions is truncated over one of the two material domains separated by the discontinuity. When modelling a strong discontinuity, this method is equivalent to XFEM with only the Heaviside enrichment function [5]. Belytschko and co-workers [6, 127] proposed the notion of phantom (or ghost) nodes and coined the Hansbo and Hansbo's method [57] as the PNM. When an element is cut by a discontinuity, extra (phantom) nodes are superposed with the original nodes and two superposing elements are formed (Figure 2.6). Effectively, only part of the domain

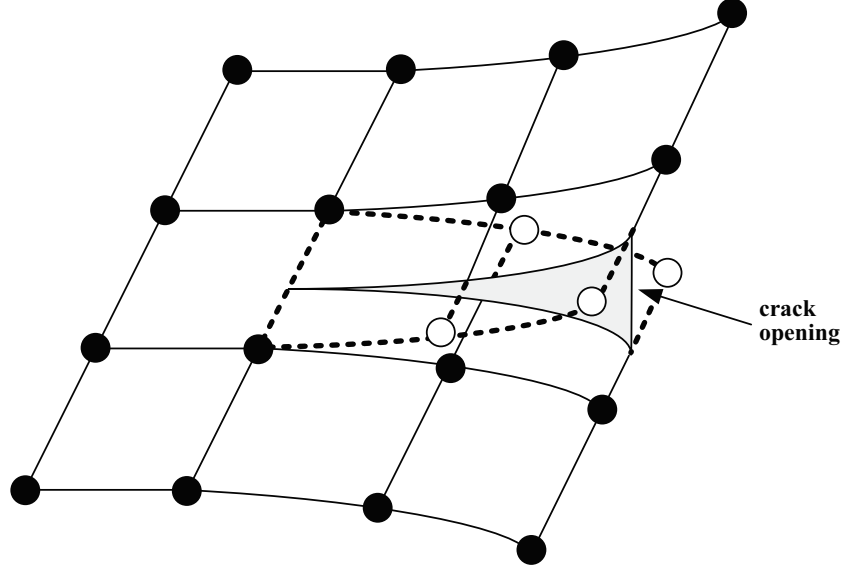


Figure 2.6: An illustration of the Phantom Node Method from [127]. Solid circles denote the original nodes and hollow circles denote the phantom nodes.

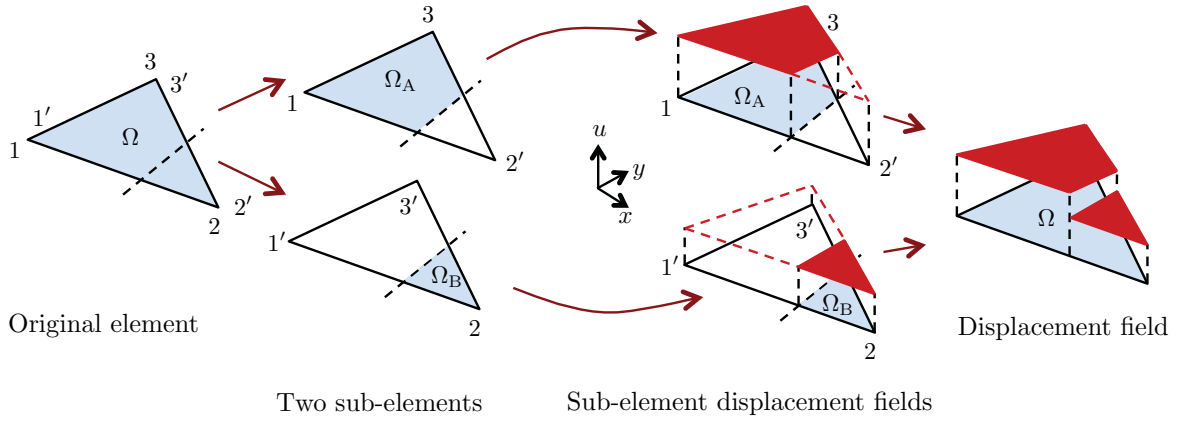


Figure 2.7: Calculations of the Phantom Node Method from [90]; 1, 2 and 3 denote the original nodes and 1', 2' and 3' denote the phantom nodes.

in each element is active, and the stiffness and nodal force integration is only performed on this active part of the domain (Figure 2.7). Since only standard shape functions are used for the displacement interpolation and each node has only displacement DoF, the PNM can be more readily implemented in existing FEM packages than XFEM, leading to numerous further developments and applications in the literature [33, 46, 49, 50, 83, 89–91, 113, 118, 126, 143, 146, 147]. It was applied in the modelling of failure in composites in [50, 83, 143, 146, 147] and showed good performance. Similarly to XFEM, the PNM is associated with approximations in the mapping of cracks (between physical and natural coordinates) in quadrilateral elements or elements with non-constant strain field [113]. Special integration technique such as the Delaunay triangulation is needed for the numerical integration over partial domains [143]. The illustration and formulation of the PNM will be presented in further details later in Chapter 4 Section 4.1.

2.8 Discussions

Table 2.1 summarises the relative advantages and disadvantages of a selection of existing methods for the modelling of progressive damage and failure. All the methods reviewed above have advantages and limitations on different aspects of modelling. Out of these methods, the smeared crack formulation is energetically consistent and its implementation in existing FEM packages is straightforward since it does not require changes in the existing mesh topology, and no special treatment is needed for the modelling of complex crack networks. The smeared crack formulation is commonly used in the failure modelling of composites [29, 105, 108, 128, 144] and it has shown varied aspects of success and limitations. More advanced methods such as XFEM and the PNM possess advantages in the kinematic representation of cracks; however, they induce complexity in the numerical integration and errors in the mapping between physical and natural coordinates. In the next chapter, the smeared crack formulation, together with some chosen failure criteria and cohesive laws, will be used in the modelling of fibre breaking and matrix cracking and cohesive elements will be used for the modelling of delamination; the performance of these two numerical tools will be evaluated on the open-hole tension problem of [53], before

introducing a novel numerical method in Chapter 4 which will be shown to possess several advantages over existing advanced method such as the PNM in the modelling of complex failure scenarios.

Table 2.1: Some numerical methods for representing discontinuities in solids.

Method	Advantages	Disadvantages
Remeshing [9, 26, 74, 75, 93–95, 101, 122–124, 134]	Accurate representation of discontinuities with optimal element size and shape.	Numerically inefficient.
Stiffness degradation method [13, 29]	Simplicity.	Smearing of the discontinuities and strength mesh dependency.
Cohesive element [3, 28, 56, 58, 81, 103, 112, 119, 141, 142, 153, 157]	Accurate representation of cohesive cracks.	A priori knowledge of the potential crack paths is required.
Smeared crack formulation [16, 29, 43, 100, 104, 105, 108, 110, 120, 144, 151]	Simplicity and numerical stability.	Smearing of the discontinuities and fracture path mesh dependency.
XFEM [7, 21, 23, 51, 59, 63, 65, 73, 92, 97, 116, 117, 129, 130, 135]	Accurate representation of discontinuities, and possibility to represent the singular stress field around a crack tip	Non-trivial implementation in existing FEM codes, approximation in the mapping of cracks (between physical and natural coordinates) and complexity of numerical integration over cracked elements.
PNM [6, 33, 46, 49, 50, 57, 83, 89–91, 113, 118, 126, 127, 143, 146, 147]	Accurate representation of discontinuities, and convenient implementation in existing FEM codes.	Approximation in the mapping of cracks (between physical and natural coordinates), and complexity of numerical integration over partial domains.

Chapter 3

Open-hole tension size effect

3.1 Introduction

It is often observed that the strength of composites is dependent on specimen size. This size effect has been studied previously by many researchers and comprehensive reviews of early studies can be found in [131–133]. It is generally established that the strength decreases with increasing specimen size. Recently, Green et al. [53] have performed a series of detailed experiments on open-holed quasi-isotropic ($[45_m/90_m/-45_m/0_m]_{ns}$) IM7/8552 carbon-epoxy composites. Three different scaling methods were applied in the experiment, namely in-plane-scaling, sublaminar-scaling and ply-scaling. In-plane scaling refers to the scaling of in-plane dimensions of the laminate, without modifying its geometry, lay-up and its thickness dimension (Figure 3.1). Figure 3.2 illustrates the two thickness direction scaling methods. Sublaminar-scaling of the composite laminate refers to the blocking of sub-laminates in the thickness direction of the laminate, without changing the in-plane geometry and dimensions of the laminate. In the case of the laminates in this study, the baseline laminate is of lay-up $[45/90/-45/0]_s$. A sub-laminate of lay-up $[45/90/-45/0]$ (i.e., half of the baseline laminate) is continuously added to the top and bottom of the baseline laminate, resulting in a thicker laminate of lay-up $[45/90/-45/0]_{ns}$, where n is the number of sub-laminates on each side of the mid-plane. Ply-scaling increases the thickness of each ply-block so as to increase the thickness of the laminate; it scales the baseline laminate of

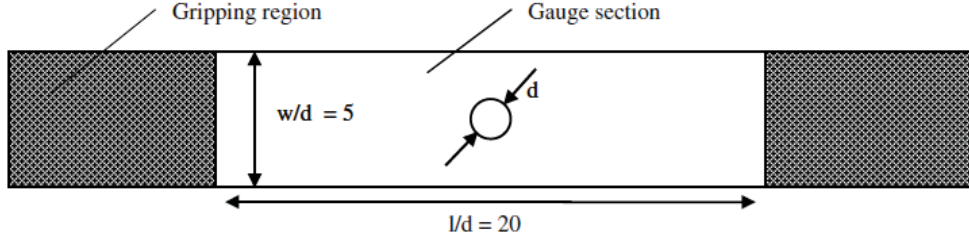
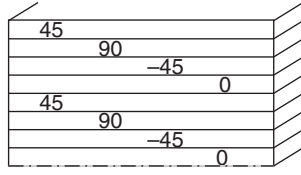
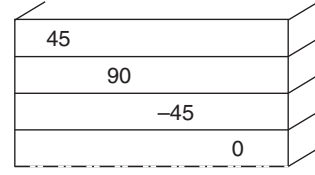


Figure 3.1: Inplane-scaling with respect to hole diameter d [156]



(a) Sublaminatescaling method, with $n = 1, 2, 4; m = 1$



(b) Ply-scaling method, with $m = 1, 2, 4; n = 1$

Figure 3.2: Different thickness-scaling methods for laminates of lay-up $[45_m/90_m/-45_m/0_m]_{ns}$ [156]

lay-up $[45/90/-45/0]_s$ into a laminate of lay-up $[45_m/90_m/-45_m/0_m]_s$, where m indicates the number of plies blocked together in the same direction.

The experiment recorded mainly four sets of size effect data, i.e. the in-plane size effect and thickness size effect of both sublaminated specimens and ply-scaled specimens. The size effects include both the change of the strength and that of the failure mode of the structure with respect to the different scaling methods. Three distinctive failure modes namely brittle failure, pull-out, and delamination (Figure 1.6) exist for specimens of different sizes and lay-ups. The failure mode of the sublaminated specimens is either pull-out or brittle and the load-drop is caused by fibre-breaking in 0 plies. Ply-scaled specimens however mostly fail in delamination mode and the load-drop is caused by delamination at the $-45_m/0_m$ interface prior to fibre-breaking of the 0 plies. In terms of the laminate strength, in-plane scaling of laminates of lay-up $[45/90/-45/0]_{4s}$ has been shown to cause a decrease in strength (Figure 3.3a); however, the in-plane scaling of laminates of lay-up $[45_4/90_4/-45_4/0_4]_s$ has lead to an increase in strength (Figure 3.3b). Sublaminated scaling has resulted in a gradual de-

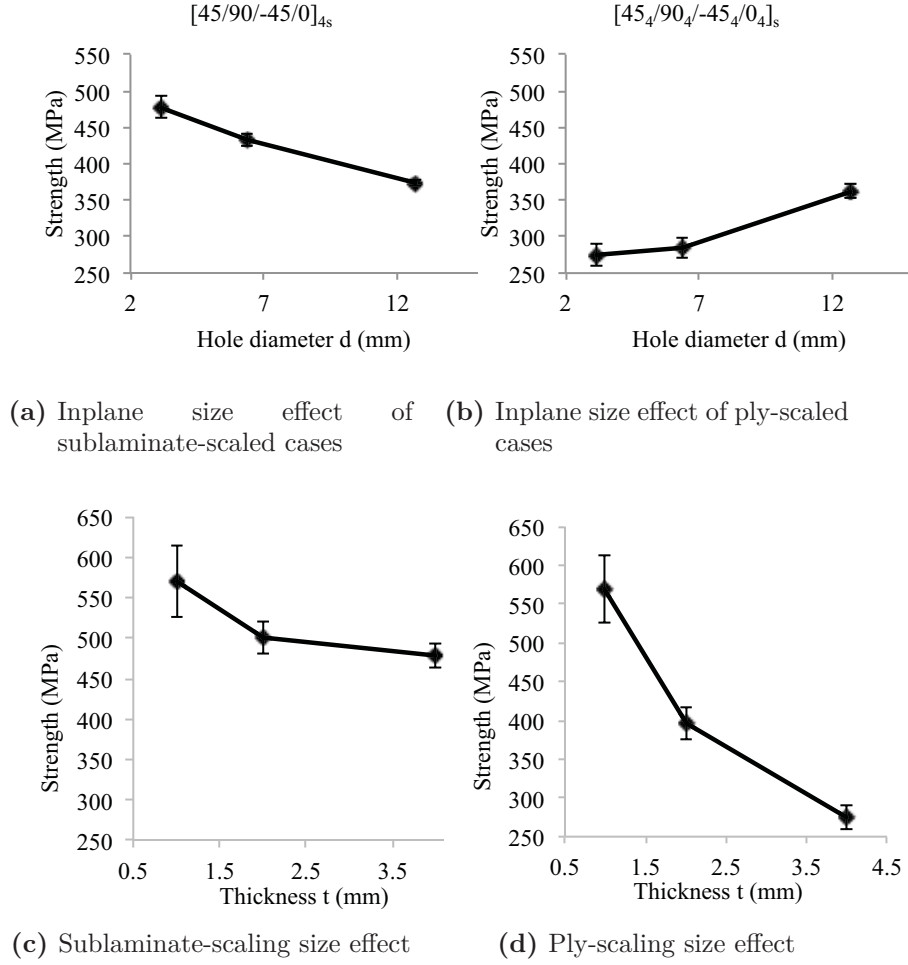


Figure 3.3: Different strength size effects [53]

crease of laminate strength (Figure 3.3c); ply-scaling has caused a decrease of strength (Figure 3.3d).

Numerical studies of size effects of open-hole composites have been performed by Camanho et al. [29], Hallett et al. [56], Abisset et al. [2], Song et al. [128], van der Meer et al. [146], and Swindeman et al. [135] using different approaches. Camanho et al. [29] used an energy-regularized Continuum Damage Mechanics (CDM) model to predict material local behaviour. The stress-strain relationship of each element was regularized such that the total energy dissipation upon failure equals the fracture toughness of the material. Shell elements formulated based on Classical Lamination Theory (CLT) were used in the model. Only the in-plane size effect of brittlely-failed sublaminate-scaled specimens was studied and good results

were obtained. Hallett et al.'s model [56] included cohesive elements inserted between plies to model delamination and also at certain potential crack paths in plies to model matrix cracking. A global criterion based on Weibull's statistical model was used to determine fibre failure of the entire structure. The model gave good predictions on the interaction between matrix cracks and delamination which shows that cohesive elements are versatile for matrix crack modelling if positioned correctly. Abisset et al. [2] used a brittle model for fibre breaking and a CDM model for matrix cracking and delamination. The coupling between matrix crack and delamination is accounted for in the material constitutive law for the interface cohesive elements. The in-plane size effect of both the sublaminates and the ply-scaled specimens, as well as the thickness size effect of the ply-scaled specimens, were studied. The thickness size effect of the sublaminates was not investigated. Song et al. [128] used the smeared crack formulation for ply failure and cohesive elements for delamination. A so-called aligned mesh is specifically constructed for each ply-block such that the element edges are aligned with the local fibre directions. Only the size effects of ply-scaled specimens were studied. The delamination failure mode associated with the ply-scaled specimens was well captured; the laminate strength predictions show considerable discrepancy with respect to the experimental data. van der Meer et al. [146] used the PNM to explicitly represent matrix cracks in plies and a smeared crack formulation for fibre failure modelling. Good predictions were obtained for the in-plane size effect of both the sublaminates and the ply-scaled specimens and for the thickness size effect of the ply-scaled specimens. The thickness size effect of the sublaminates was not successfully predicted. Swindeman et al. [135] used the regularized XFEM to model matrix cracking and delamination. Only ply-scaled specimens were modelled. The study focused on the modelling of the delamination failure mode; no fibre breaking was considered. Good predictions on the delamination failure mode and the ply-scaling size effects were achieved.

From the above review it is seen that for deterministic failure theories, i.e. theories without considering statistical variations of material properties, the prediction of the thickness size effect of the sublaminates remains a challenge.

This chapter presents a numerical study of the size effects on open-hole tensile composite laminates recorded in the experiment by Green et al. [53]. The smeared crack formulation is used for the modelling of fibre breaking and matrix cracking and cohesive elements are used for the modelling of delamination. Section 3.2 will present the failure theories used in this work. A discussion about the choice between deterministic theories and statistical theories will be given in Section 3.2.1. The thickness dependence of translaminar fracture toughness will be included in the numerical modelling of composites (Section 3.2.2). Some common numerical models of different complexities will be built; they will be tested for mesh convergence and a mesh-independent numerical model will be selected in Section 3.3. The sublaminar-scaling thickness size effect will be studied and the importance of including the thickness dependence of translaminar fracture toughness in the failure modelling of composites discussed in Section 3.4.1. The ply-scaling thickness size effect will be studied in Section 3.4.2. The in-plane size effect of the sublaminar-scaled specimens will be studied in Section 3.4.3 and that of the ply-scaled specimens in Section 3.4.4. Some parametric sensitivity analysis of the model will be presented in Section 3.4.5, followed by conclusions in Section 3.5.

3.2 Failure Theory

3.2.1 Fibre Failure

Carbon fibres normally show brittle behaviour and their strength usually decreases with increasing fibre length due to the Weakest Link Theory. This size effect of fibre strength can be characterized by a Weibull statistical distribution [150]. The fibre-matrix composite however is not exactly brittle. Fibre-matrix debonding and friction result in a high fracture energy and therefore a more gradual propagation of fracture [78, 79]. In the presence of stress concentrations caused by notches and holes, the fibres closer to notch tips and hole edges are subject to higher stresses than the rest of the domain; however, the Weakest Link Theory only applies under the assumption of a uniform stress state. Therefore, the influence of statistical variation of material strength on the load-bearing capacity of open-hole

composites is limited; a deterministic physical phenomenon may exist that is also responsible for the strength size effect observed. Previous studies on size effect of concrete has shown that a deterministic energetic size effect exists for materials with quasi-brittle behaviour, i.e. materials that undergo gradual propagation of fracture before final failure [14, 15, 17–19]. Composites fall into this category and therefore the same physics apply [20, 78]. Fibre failure initiates at higher stress locations and the fracture toughness is high enough to allow gradual propagation of fracture. The final failure is then determined by an energy balance, e.g. when the strain energy released balances the fracture energy dissipated. This energy balance is best captured by the cohesive zone theory [12, 42, 48, 60, 105, 144](see Chapter 2 Section 2.4).

In this work, uniform strength is applied throughout the material domain. Fibre failure is modelled here by a smeared crack formulation, with linear elasticity before failure onset, and a cohesive linear softening law after failure onset. A simple maximum stress criterion is used to determine failure onset, where the fibre directional tensile strength is X_t (see its value in Table 3.1). It is observed in the experiments [155] that the unidirectional lamina’s fibre failure strength under uniform tension is size-dependent. In [155], the authors changed each linear dimension by a factor of 8 (or the volume by a factor of 512) and obtained a change in strength of 14%. In the case of open-hole specimens, stress concentrations are present at the boundary of the hole, and the critical volume (over which the fibre failure strength is consequential) is very small compared to the volume of the structure. In this situation, it appears reasonable to use the value in [155] corresponding to the smallest volume tested, which is the approach followed in this chapter. An alternative would be to extrapolate the strength versus size curve in [155] to a smaller volume based on the assumption of the Weakest Link Theory (such as in [56, 146]). However it is difficult to justify the choice of any specific value for this volume without a theoretical model for the stochastic strength scaling under a generic stress field (note that the Weakest Link Theory is only valid for uniform loading, see e.g. [102]). In any case, the value of the tensile strength used should be representative for the volume of the specimen being considered and the stress distribution in that specimen.

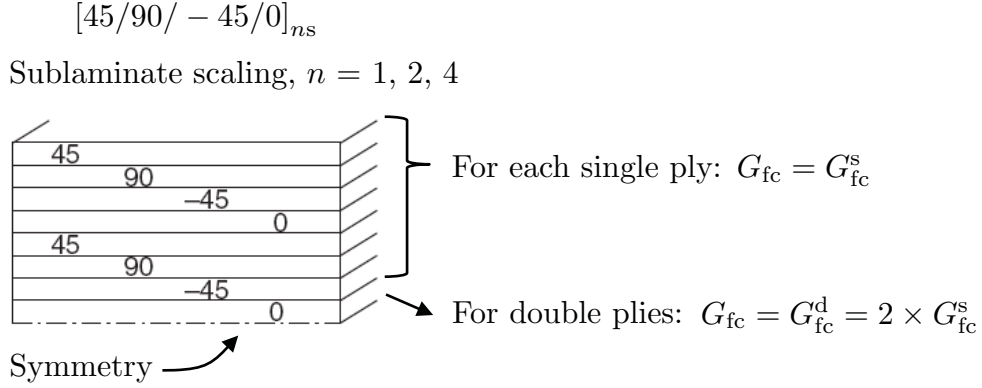


Figure 3.4: FEM implementation of the thickness dependence of translaminar fracture toughness. G_{fc}^s is the translaminar fracture toughness for a single ply and G_{fc}^d is that for the double plies in the middle of the laminate [79].

3.2.2 Thickness Dependence of Translaminar Fracture Toughness

Recent CT experiments on pre-cracked cross-ply laminates in [79] showed that the translaminar fracture toughness of T300/920 carbon-epoxy composites exhibits a dependence on the thickness of the ply-blocks. The propagation translaminar fracture toughness of the 0 plies in $[(90/0_2)_8/90]_s$ is measured to be about twice of that of the 0 plies in $[(90/0)_8/90]_s$ [79]. The change in translaminar fracture toughness with size is of $\approx 100\%$ with only a factor of two in the linear dimension (thickness). It appears therefore that the change in the fibre strength with size observed in [155] may be negligible in comparison to the toughness size effect. The experiments in [82] on the over-height CT test of composite laminates of different lay-ups have shown that thicker ply-blocks promote macro-scale matrix splitting which causes a blunting effect of the notch tip and a larger fracture process zone. This finding is in agreement with the observation in [79] where thicker 0 ply-blocks lead to relatively longer matrix cracks along the fracture path which in turn leads to a larger amount of fibre pull-out (Figure 1.9). However, the scale of the matrix cracks in Figure 1.9 is in the order of 0.1 mm (which is very small compared to the in-plane dimension of the specimen) and they should not be confused with the macro-scale matrix cracks in [82].

A meaningful measurement of the translaminar fracture toughness implies that the measured specimens fail in the desired failure mode, i.e., no significant failure should appear away from the fracture path. This in turn implies that the scale of the matrix cracks along the fracture path should be maintained small compared to the in-plane dimension of the specimen (which is the intended outcome in [79]).

To the author's knowledge, no numerical work has employed this thickness dependence of translaminar fracture toughness; and no quantitative mathematical model that describes the correlation of the geometry and the loading of the material with its in situ fracture toughness has been established. The focus in this work is to implement numerically this thickness dependence of the translaminar fracture toughness, and examine its importance on failure predictions. For the IM7/8552 carbon-epoxy material studied in this work, only the translaminar fracture toughness values of a single ply (G_{fc}^s) are available from Catalanotti et al. [32] and Laffan et al. [80] (Table 3.1). The IM7/8552 material and the T300/920 material tested in [79] are both fibre-reinforced composite materials of similar geometrical construction and dimensions. The translaminar fracture of both materials are composed of localized fibre breaking, matrix cracking, fibre-matrix debonding and fibre pull-out along the path of propagation [80]. The two materials have similar characteristics of translaminar fracture and only differ in the exact values of the strength and toughness of the respective constituents; if a scaling of 2 in thickness causes a scaling of 2 in translaminar fracture toughness of T300/920, a similar scaling phenomenon could be expected to occur in the translaminar fracture of IM7/8552. Therefore, in this study, it is assumed that the translaminar fracture toughness of the IM7/8552 composite will have similar thickness dependency as that of the T300/920 composite such that its value for the double 0 plies (G_{fc}^d) is twice of that of a single 0 ply (Table 3.1). However, experiments as presented in [79] could be conducted on the IM7/8552 material to verify such an assumption. This thickness dependency is applied to the numerical modelling in this chapter (Figure 3.4) and it will be shown later in Section 3.4.1 that this is critical to the prediction of thickness size effect of the sublaminar-scaled specimens.

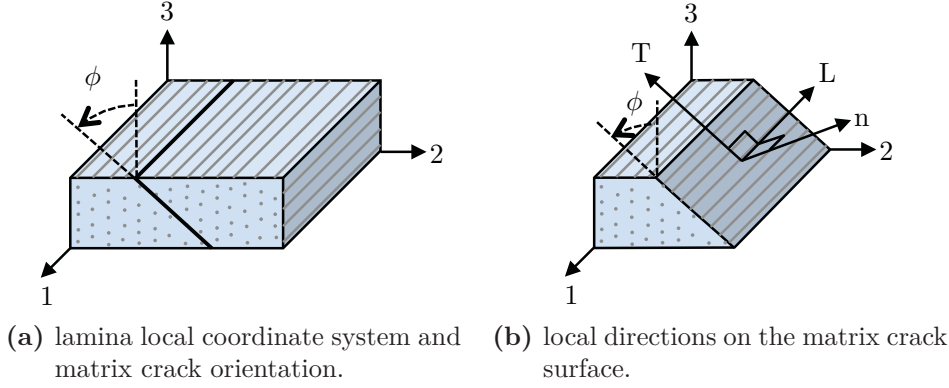


Figure 3.5: Illustration of matrix cracking and local material directions.

3.2.3 Matrix Failure

The criterion in Pinho et al. [104] is used here for matrix failure initiation. Figure 3.5a illustrates the local material orientations and the matrix crack orientation ϕ which is the angle between the crack surface and the through-the-thickness direction (direction 3). Figure 3.5b shows the normal direction (direction n), transverse shear direction (direction T) and longitudinal shear direction (direction L) on the matrix crack surface. The criterion for matrix failure considers the role of three stresses, namely the normal stress (σ_n), the transverse shear stress (σ_T) and the longitudinal shear stress (σ_L) on the matrix crack surface, as they are the active stress components on that surface; fibre-directional stress, σ_{11} , is not included in the failure criterion as its component on the matrix crack surface is zero. The three active stresses on the fracture surface can be calculated from the stresses in local coordinates as the following [104]:

$$\sigma_n = \frac{\sigma_{22} + \sigma_{33}}{2} + \frac{\sigma_{22} - \sigma_{33}}{2} \cos(2\phi) + \sigma_{23} \sin(2\phi), \quad (3.1)$$

$$\sigma_T = -\frac{\sigma_{22} - \sigma_{33}}{2} \sin(2\phi) + \sigma_{23} \cos(2\phi), \quad (3.2)$$

$$\sigma_L = \sigma_{12} \cos(\phi) + \sigma_{31} \sin(\phi). \quad (3.3)$$

Correspondingly, we have the three strains acting on the matrix crack surface:

$$\epsilon_n = \frac{\epsilon_{22} + \epsilon_{33}}{2} + \frac{\epsilon_{22} - \epsilon_{33}}{2} \cos(2\phi) + \gamma_{23} \sin(2\phi), \quad (3.4)$$

$$\gamma_T = -\frac{\epsilon_{22} - \epsilon_{33}}{2} \sin(2\phi) + \gamma_{23} \cos(2\phi), \quad (3.5)$$

$$\gamma_L = \gamma_{12} \cos(\phi) + \gamma_{31} \sin(\phi). \quad (3.6)$$

Matrix compressive failure is assumed when the following criterion is met:

$$\left(\frac{\sigma_T}{S_T - \mu_T \sigma_n} \right)^2 + \left(\frac{\sigma_L}{S_L - \mu_L \sigma_n} \right)^2 = 1, \quad \leftarrow \sigma_n < 0, \quad (3.7)$$

with

$$\mu_T = -\frac{1}{\tan(2\phi_o)}, \quad S_T = \frac{Y_c}{2 \tan(\phi_o)}, \quad \mu_L = S_L \frac{\mu_T}{S_T}, \quad (3.8)$$

where S_T and S_L are transverse and longitudinal shear strengths of the lamina; and μ_T and μ_L are transverse and longitudinal friction-like parameters; and Y_c is the transverse compressive strength; and ϕ_o is the angle between the matrix fracture surface and the through-the-thickness direction when the lamina is under pure compression in the in-plane transverse direction.

Matrix tensile failure is determined by the following criterion:

$$\left(\frac{\sigma_n}{Y_t} \right)^2 + \left(\frac{\sigma_T}{S_T} \right)^2 + \left(\frac{\sigma_L}{S_L} \right)^2 = 1, \quad \leftarrow \sigma_n \geq 0, \quad (3.9)$$

where Y_t is the transverse tensile strength. Generally, the angle ϕ is unknown and it needs to be determined from a few trial angles [104]. In this study, the matrix strengths for plies of different thickness and at different positions of the laminate are considered to be the same. This is a simplification; however the in situ matrix strengths could be calculated as proposed in [106]. The thermal residual stresses are also neglected here, which constitutes a limitation of the approach in this study. The issue of residual stresses due to curing is challenging because the laminates will gradually absorb moisture and relax the thermal stress after curing; therefore, the exact thermal stresses in the specimen prior to testing will be lower than those immediately after cure. As a result, it is difficult to have an accu-

rate estimate of the residual stresses in the specimen immediately before testing, which justifies the approach in this study of neglecting residual stresses. Otherwise, they could be included as done in [56, 146].

For failure propagation modelling, Pinho et al. [105] developed a smeared crack formulation for matrix failure which correctly takes into account of the fracture energy dissipation. In this work, the smeared crack formulation is used for matrix cracking. The mixed-mode linear cohesive softening law is based on the equivalent quantities:

$$\sigma_{\text{eq}} = \sqrt{\langle \sigma_{\text{n}} \rangle_+^2 + \sigma_{\text{T}}^2 + \sigma_{\text{L}}^2} \quad (3.10)$$

and

$$u_{\text{eq}} = \frac{\langle \sigma_{\text{n}} \rangle_+ \langle \epsilon_{\text{n}} \rangle_+ + \sigma_{\text{T}} \gamma_{\text{T}} + \sigma_{\text{L}} \gamma_{\text{L}}}{\sigma_{\text{eq}}} l_{\text{e}}, \quad (3.11)$$

where $\langle \bullet \rangle_+$ are the Macaulay brackets which return the value of the number enclosed if it is non-negative, and zero if otherwise; l_{e} is the characteristic length of the element [105]. The matrix mixed-mode fracture energy G_{mc} is defined using the B-K formula developed by Benzeggagh and Kenane [24]:

$$G_{\text{mc}} = G_{\text{Ic}} + (G_{\text{IIc}} - G_{\text{Ic}}) B^\eta, \quad (3.12)$$

where η is a material property. G_{Ic} and G_{IIc} are the Mode I and Mode II critical fracture energies. The mixed-mode ratio B is defined at the failure onset as:

$$B = \frac{G_{\text{S}}}{G_{\text{N}} + G_{\text{S}}}, \quad (3.13)$$

with G_{N} and G_{S} defined at the failure onset as:

$$G_{\text{N}} = \frac{\langle \sigma_{\text{n}} \rangle_+ \langle \epsilon_{\text{n}} \rangle_+ l_{\text{e}}}{2}, \quad G_{\text{S}} = \frac{(\sigma_{\text{T}} \gamma_{\text{T}} + \sigma_{\text{L}} \gamma_{\text{L}}) l_{\text{e}}}{2}. \quad (3.14)$$

After failure initiation, σ_{eq} softens with respect to u_{eq} and the total energy dissipation when σ_{eq} reduces to zero equals G_{mc} [105]. The values of the material parameters used in the matrix failure criterion are summarized in Table 3.1.

3.2.4 Delamination

Delamination is one of the most critical threats to the integrity and load-bearing capacity of composite laminates. Extensive research has been devoted to delamination in the past. Early work focuses on fracture mechanics studies of stationary pre-cracks or embedded cracks. A comprehensive and critical review of these methods can be found in [139]. Current computational power makes the simulation of delamination propagation within reach. It is often assumed that delamination happens in the homogeneous resin-rich region between plies and cohesive elements are used to represent the interface.

In this part of the work, the Abaqus cohesive element (COH3D8) [1] is used for delamination modelling. The element topology and constitutive laws are the same as shown in Figure 2.3. The cohesive tractions are related to the respective separations in the normal and two shear directions (Figure 2.3a) through the following equation:

$$\begin{bmatrix} \tau_n \\ \tau_T \\ \tau_L \end{bmatrix} = \begin{bmatrix} D_{nn} \left(1 - \frac{\langle \delta_n \rangle_+}{|\delta_n|} d\right) & 0 & 0 \\ 0 & D_{TT} (1 - d) & 0 \\ 0 & 0 & D_{LL} (1 - d) \end{bmatrix} \begin{bmatrix} \delta_n \\ \delta_T \\ \delta_L \end{bmatrix} \quad (3.15)$$

where D_{nn} , D_{TT} and D_{LL} are the penalty stiffness; d is the damage variable which equals 0 when there is no damage in the interface, and 1 when the interface is completely fractured (note that the crack is considered closed and thus the damage variable does not influence the τ_n under compression).

A quadratic interactive criterion is used to determine delamination initiation:

$$\frac{\tau_n^2}{\tau_n^c{}^2} + \frac{\tau_T^2}{\tau_T^c{}^2} + \frac{\tau_L^2}{\tau_L^c{}^2} = 1 \quad (3.16)$$

where τ_n^c , τ_T^c and τ_L^c are the normal strength and the two shear strengths. Delamination propagation is represented by updating the value of the damage variable d with respect to the separation according to the linear softening laws in Figure 2.3b. In the case of mixed-mode fracture, similarly as

in matrix cracking, the softening law is defined on equivalent traction σ_{eq} and equivalent separation u_{eq} which are defined as:

$$\sigma_{\text{eq}} = \sqrt{\langle \tau_{\text{n}} \rangle_+^2 + \tau_{\text{T}}^2 + \tau_{\text{L}}^2} \quad (3.17)$$

and

$$u_{\text{eq}} = \frac{\langle \tau_{\text{n}} \rangle_+ \langle \delta_{\text{n}} \rangle_+ + \tau_{\text{T}} \delta_{\text{T}} + \tau_{\text{L}} \delta_{\text{L}}}{\sigma_{\text{eq}}}. \quad (3.18)$$

The mixed-mode fracture energy is also determined by the B-K formula (3.12), with G_{N} and G_{S} defined at the failure onset as:

$$G_{\text{N}} = \frac{\langle \tau_{\text{n}} \rangle_+ \langle \delta_{\text{n}} \rangle_+}{2}, \quad G_{\text{S}} = \frac{(\tau_{\text{T}} \delta_{\text{T}} + \tau_{\text{L}} \delta_{\text{L}})}{2}. \quad (3.19)$$

After delamination initiation, σ_{eq} softens (i.e., d grows) with respect to u_{eq} and the total energy dissipation when σ_{eq} reduces to zero equals G_{mc} [1].

Delamination modelling using cohesive elements can be very sensitive to element size. The element size must be small enough to capture the stress gradient in the wake of cracks. A cohesive zone, where the material undergoes softening, develops ahead of the stress-free crack tip. Generally, cohesive element size must be smaller than the cohesive zone length to ensure a correct and smooth load-displacement solution [58, 112, 141, 157]. A practical rule was proposed in [58] to estimate the numerical cohesive zone length for the bilinear cohesive law and it is used in this work to determine the maximum cohesive element size. Typically the cohesive zone length under Mode I loading is smaller than that under Mode II loading [157]. Since the loading in this study is purely in-plane, delamination propagation is mainly driven by shear. Therefore, only the formula for the calculation of Mode II cohesive zone length in [58] is used:

$$l_{\text{ch}} = 0.5 \times \min(l_{\text{ch,II}}, l_{\text{ch,slender,II}}) \quad (3.20)$$

where

$$l_{\text{ch,II}} = E'_{\text{II}} \frac{G_{\text{IIc}}}{\tau_{\text{T}}^{\text{c}^2}} \quad (3.21)$$

and

$$l_{\text{ch,slender,II}} = \sqrt{E'_{\text{II,slender}} \frac{G_{\text{IIc}}}{\tau_{\text{T}}^{\text{c}^2}} h} \quad (3.22)$$

where h is the laminate half-thickness, E'_{II} is an equivalent elastic modulus for orthotropic materials and $E'_{\text{II,slender}}$ is an equivalent elastic modulus for a slender body. The expressions of E'_{II} and $E'_{\text{II,slender}}$ can be found in [58]. Different values of h would lead to different estimations for the cohesive zone length. The most conservative approach would be to assume that delamination happens at all interfaces and that h would be the thickness of the thinnest ply-block. In the case of the sublaminates-scaled specimens, h equals to the single ply thickness t ; and in the case of the ply-scaled specimens of the lay-up $[45_m/90_m/-45_m/0_m]_s$, h equals $m \times t$. Generally, two to three cohesive elements are required in the numerical cohesive zone to achieve accurate delamination predictions [58].

The penalty stiffness D_{nn} , D_{TT} and D_{LL} are artificial parameters used to constrain the separation between the cohesive surfaces before delamination onset, so that perfect-bonding between plies can be assumed before delamination. Therefore, they must be large enough compared to the stiffness of the lamina. The following formulae from [141] are used to calculate the minimum values of the penalty stiffness:

$$D_{\text{nn}}^{\min} = \frac{50 E_3}{t}, \quad D_{\text{TT}}^{\min} = \frac{50 G_{13}}{t}, \quad D_{\text{LL}}^{\min} = \frac{50 G_{23}}{t} \quad (3.23)$$

where t is the thickness of a single ply and E_3 , G_{13} and G_{23} are the elastic moduli of the lamina.

A relationship between the normal and shear penalty stiffness was proposed in [142] to ensure an accurate simulation of delamination growth under mixed-mode loading:

$$D_{\text{TT}} = D_{\text{LL}} = D_{\text{nn}} \frac{G_{\text{Ic}}}{G_{\text{IIc}}} \left(\frac{\tau_{\text{T}}^c}{\tau_{\text{n}}^c} \right)^2 \quad (3.24)$$

In this work, $D_{\text{nn}} = D_{\text{nn}}^{\min}$. D_{TT} and D_{LL} are calculated from (3.24) and they are verified to be larger than $D_{\text{TT}}^{\min}$ and $D_{\text{LL}}^{\min}$ respectively in Equation 3.23. The values of the material parameters used in the delamination failure criterion are summarized in Table 3.1. The values of delamination strengths are estimated based on the fact that reduced strength values would lead to a larger cohesive zone which then spans more cohesive elements; it has been studied in [58, 141] that using such a practice would get

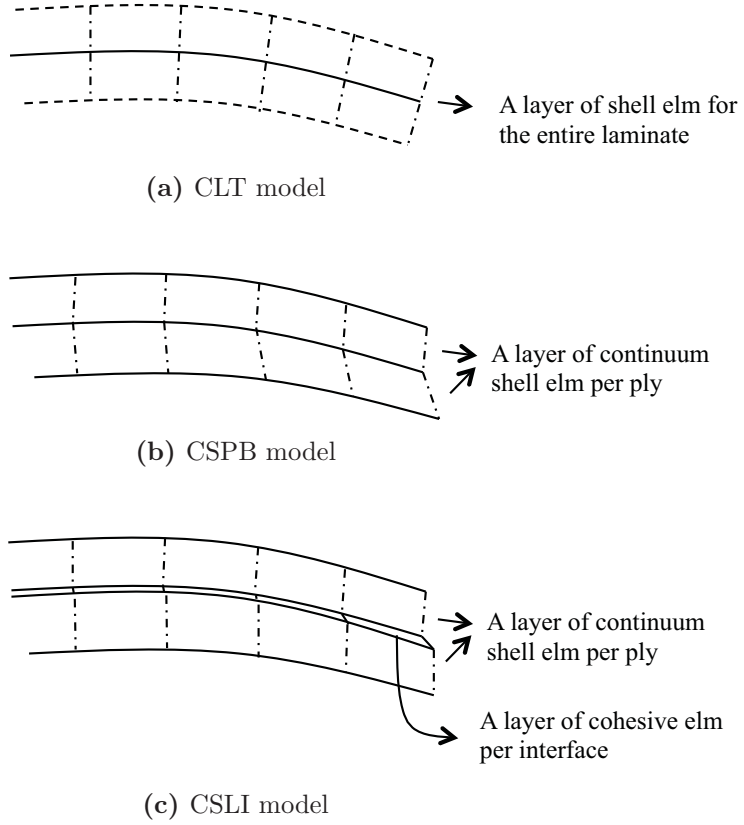
more accurate solutions, provided that the artificial reduction in strength is within a reasonable limit [58]. From the benchmark studies in [58], it appears that as long as the reduced normal strength is more than 30% of the real normal strength and the reduced shear strength more than 50% of the real shear strength, the predictions of the cohesive elements in the DCB and ENF simulations are still in reasonably close agreement to those obtained with the real strength values. The strength values of cohesive elements in this work are chosen on this basis, instead of using the usual strength values of $\tau_n^c=60$ MPa and $\tau_T^c = \tau_L^c=90$ MPa in [56, 146].

Table 3.1: IM7/8552 carbon/epoxy material properties for the open-hole laminates [56]

E_1	$E_2 = E_3$	$\nu_{12} = \nu_{13}$
161 GPa	11.4 GPa	0.32
ν_{23}	$G_{12} = G_{13}$	G_{23}
0.43	5170 MPa	3980 MPa
X_t [155]	Y_t	Y_c [72]
2806 MPa	60 MPa	185 MPa
S_L	ϕ_o [105]	t
90 MPa	53°	0.125 mm
G_{fc}^s [80]	τ_n^{c*}	$\tau_T^{c*} = \tau_L^{c*}$
112.7 kJ/m ²	40 MPa	50 MPa
G_{Ic}^{**}	G_{IIc}^{**}	η
0.293 kJ/m ²	0.631 kJ/m ²	1

* Estimated values

** Values measured at Imperial College London



3.3 Mesh Refinement Study of Finite Element Models

In this section, three different FE models are examined for accuracy and convergence on one selected test case of hole diameter $d = 3.175$ mm and lay-up $[-45/90/45/0]_s$ for four different in-plane meshes (Figure 3.7). They are namely the Classical Lamination Theory (CLT) model, the Continuum Shell Laminate with Interface (CSLI) model and the Continuum Shell Perfect Bonding (CSPB) model. The interpolation order is 1 for all the elements used in this study. The failure theories presented in Section 3.2 are written in an Abaqus user-material subroutine. All material properties used are summarized in Table 3.1.

3.3.1 Classical Lamination Theory (CLT) model

The CLT model uses a single shell element based on CLT to represent all plies through the thickness (Figure 3.6a). The CLT model is implemented

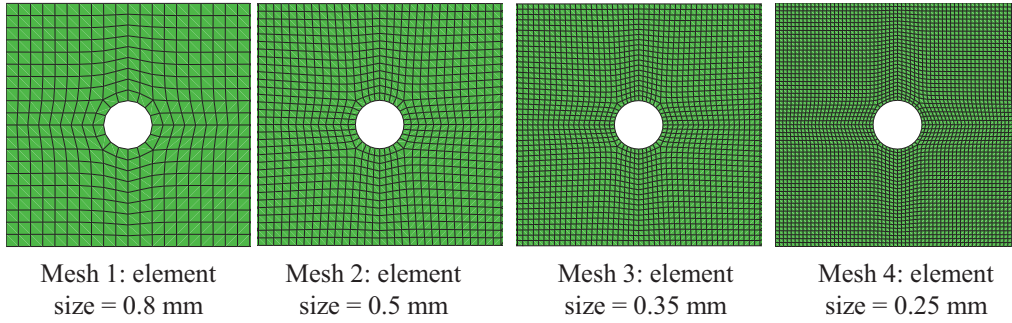


Figure 3.7: Four meshes of different element sizes for the open-hole model

as standard in Abaqus; it can be built using Abaqus conventional shell element S4R with the composite lay-up option. The stiffness of the shell element is obtained from through-the-thickness integration of the stiffness of all the plies in the laminate. The CLT model is here tested for accuracy and mesh objectivity. The strength (maximum load divided by cross-sectional area of the gripping end) predictions by CLT model are summarized in Figure 3.8 with the curve labelled ‘CLT’. It can be seen that both mesh-dependency and over-prediction are present on the CLT results. Since one of the most original features of the model is the use of a higher translaminar fracture toughness for the two 0 plies in the middle (Figure 3.4), the value of this property is reduced to be equal to that of the single ply. In other words, the thickness variation of translaminar fracture toughness is suppressed, to investigate if the over prediction and mesh dependency still exist. The results are shown in Figure 3.8 with the curve labelled ‘CLT(b)’ and it can be seen that the over prediction and mesh dependency are still present.

3.3.2 Continuum Shell Laminate with Interface (CSLI) model

The same problem is analysed using a more detailed FE model, i.e., the CSLI mode. The CSLI model is composed of one layer of Abaqus continuum shell elements (SC8R) to represent each ply and one layer of Abaqus cohesive elements (as presented in Section 3.2.4) between plies of different orientations to model delamination (see Figure 3.6c). The in-plane mesh

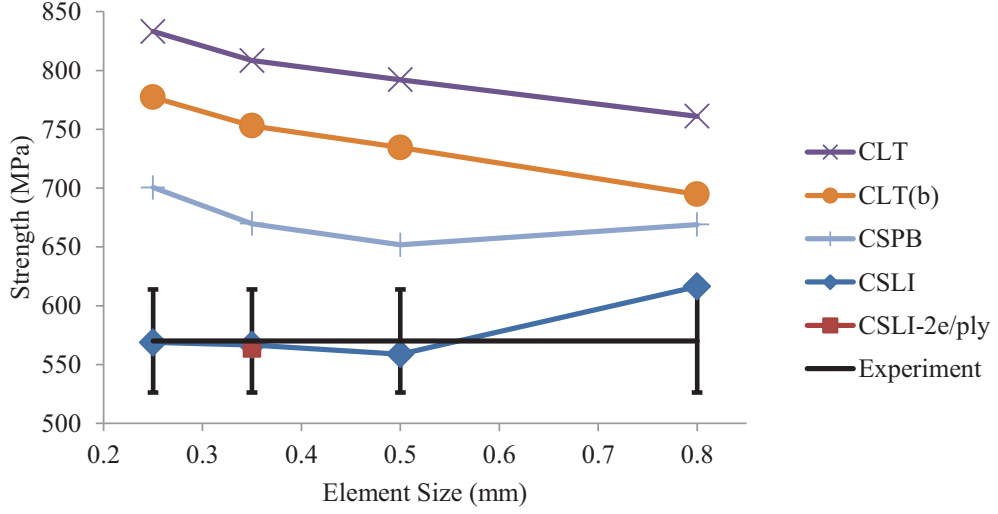


Figure 3.8: Mesh refinement study of different numerical models

of the interface is the same as that of the ply (Figure 3.7). Using Equation 3.20 with $h = t$, the numerical cohesive zone length l_{ch} is calculated to be equal to 1.14 mm. If a minimum of two elements are required in the numerical cohesive zone, the maximum element size should be 0.57 mm. Of the four meshes examined (Figure 3.7), this requirement is satisfied on Mesh 2, 3 and 4. The results of this model are recorded in the curve labelled ‘CSLI’ in Figure 3.8. In contrast with the CLT model, the strength predictions from the CSLI model from Mesh 2 onwards are very close to the experimental data and their values do not vary significantly with respect to the element size. The result on Mesh 1 shows some over-prediction. This could be due to the fact that the cohesive element size in Mesh 1 (0.8 mm) is too large to capture the stress gradient in the cohesive zone and this often leads to a delay in the prediction of delamination onset [58]. It could also be due to the fact that the ply element is too large in Mesh 1 to capture the stress concentration around the hole, leading to an under-prediction of the stress at the hole edge which delays the initiation of failure. The mesh convergence in the thickness direction is also examined. The Mesh 3 model is run with two layers of continuum shell elements per ply thickness. The result is displayed by the data point labelled ‘CSLI-2e/ply’ in Figure 3.8 and it is almost the same as the result with one layer per ply thickness,

indicating that one layer of elements per ply thickness is sufficient in this case.

The failure pattern predictions of both the CLT model and the CSLI model are consistent for the different mesh densities. The results at both the initial stage and the final stage of loading on Mesh 2 of the two models are compared in Figure 3.9. It can be seen that, during the initial stage of the loading, the two models have similar predictions of the failure patterns. The 90 plies on both models exhibit an ‘X’ shaped matrix crack pattern. This is due to the fact that the 90 ply is bonded with the ± 45 plies before delamination, such that the stress concentrations of the ± 45 plies which form an ‘X’ shaped region are transferred onto the 90 ply which lead to matrix crack initiation at these locations. The final failure patterns predicted by the two models are very different from each other. In the predictions from the CLT model, both the ± 45 plies and the 0 ply have extensive fibre failure and the 0 ply has fibre failure not only across the width of the lamina but also in the ‘X’ shaped area as shown in the figure. The experiments [53] recorded that the specimen failed in the pull-out mode (Figure 1.6). CLT assumes perfect-bonding between all plies, therefore delamination is neglected in its formulation. The direct consequence of this simplification is that the perfect-bonding assumption may be too strong in areas where the difference between the displacement fields of adjacent plies should have caused delamination. This assumption forces the stresses that should have been released by delamination to be transferred into the plies and cause excessive ply failure. Therefore, excessive fibre failure is predicted by the CLT model in the ± 45 plies and on a large area of the 0 ply which is not observed in the experiments [53]. In the CSLI model’s predictions, fibre failure mainly occurs across the width of the 0 ply and on the sides of the -45 ply. A significant amount of delamination is predicted at all the three interfaces by the CSLI model. The 45/90 interface has triangular-shaped delamination areas located roughly between the boundaries defined by the matrix cracks in the 45 and 90 plies, which indicates that the 45 ply is pulled out of the laminate. The -45/0 interface has delamination around the hole but it does not propagate across the entire width. This corresponds well with the failure patterns of the -45 ply where matrix cracks are predicted around hole and fibre failure is predicted on the sides, indi-

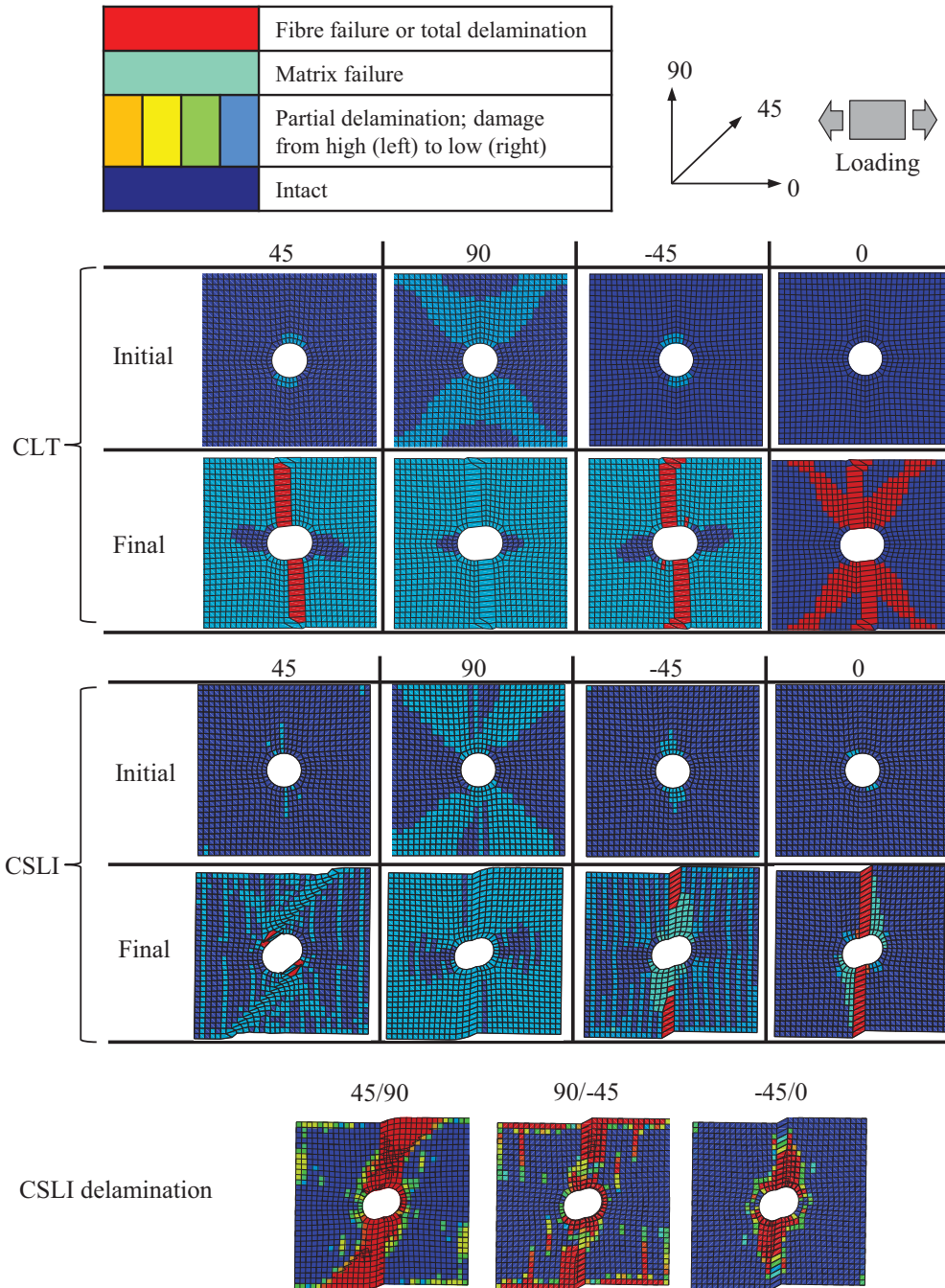


Figure 3.9: Failure patterns of the CLT and the CSLI model

cating that delamination around the hole has allowed the pull-out in this area and the bonding on the sides has helped propagate the fibre fracture from the 0 ply into the -45 ply. Some discrepancy exists between the matrix cracking prediction of the CSLI model and the experimental record, where the discrete matrix splits which should occur along the fibre direction, were not accurately captured. For example, some matrix cracks in the ± 45 plies occur along the 90 degree direction, rather than along the ± 45 directions (see Figure 3.9 CSLI model, final matrix failure patterns of ± 45 plies). This discrepancy is believed to be due to the inherent limitation of the smeared crack model in correctly capturing the propagation directions of cracks, especially when the mesh is not oriented along the actual crack propagation direction. Similar findings of this crack path mesh dependency of smeared crack models have been reported in [43, 128, 144, 146]. This discrepancy did not seem to affect the accuracy of the strength predictions of the CSLI model on this case, presumably due to the fact that the critical failure mechanisms of the specimens in this study are fibre breaking, delamination and longitudinal matrix splitting; the matrix cracking in the ± 45 directions serve to join up the more critical cracks, rather than contributing significantly to the global fracture energy dissipation. However, in a generic situation, a mesh independent model for matrix cracking (such as those in [43, 128, 144, 146] or the method presented in Chapter 4) should be employed.

The perfect-bonding assumption which neglects delamination is also responsible for the over-prediction and the mesh-dependency of the predicted strength by the CLT model (Figure 3.8) and it is illustrated in Figure 3.10. Without delamination, the deformation of the failed element would be constrained by adjacent plies. Loads are then transferred onto the surrounding elements which cause an over-prediction of the load-bearing capacity of the laminate. As the mesh is refined and the element gets smaller, the deformation of the failed element is further constrained, resulting in a smaller volume of failed region and hence a larger strength over-prediction. In the CSLI model where delamination is predicted, the elements in the middle ply are not constrained by the elements in the adjacent plies and thus capture better the actual stress-state of the material. Upon failure, the failed region returns to a stress-free state where the elements on both sides of the

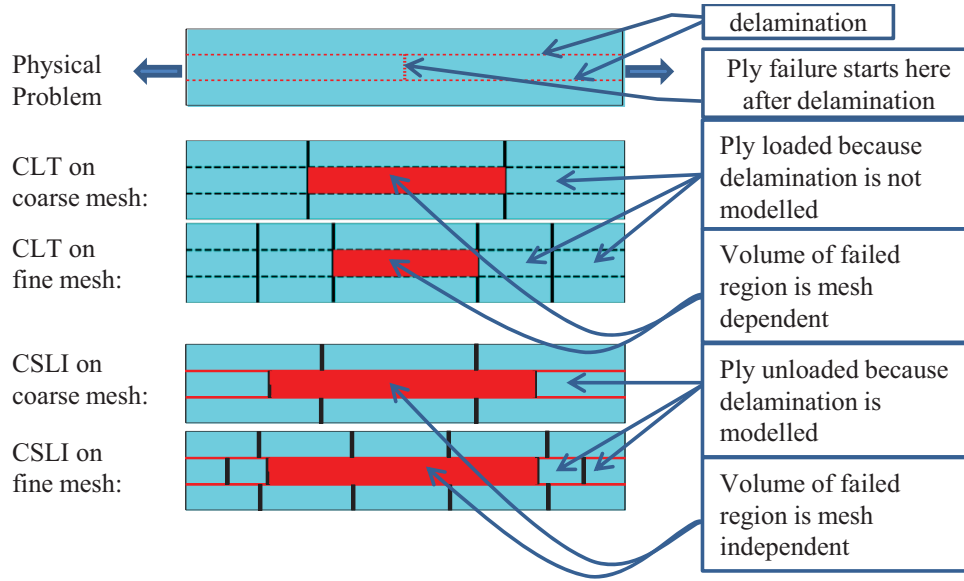


Figure 3.10: Comparison of the CLT model and the CSLI model on mesh refinement

failed element return to their original shape, leaving the failed element having a volume independent of the mesh and consequently the region having a mesh-independent stress-state.

It is worth mentioning that the mesh dependency problem discussed above is of a different nature from that associated with the stiffness degradation method discussed in Chapter 2 Section 2.3. In the latter case, an opposite trend is typically observed where the strength prediction decreases with an increasing mesh-refinement. This mesh dependency is due to the fact that the theory is only based on local stress and strain without a length-scale parameter [13]. The smeared crack formulation in [105] (which is the chosen method in this study) has an inherent length-scale in the formulation to ensure consistent energy dissipation upon fracture, which mitigates the strength mesh dependency associated with the stiffness degradation method (Section 2.5).

3.3.3 Continuum Shell Perfect Bonding (CSPB) model

In order to verify that neglecting delamination modelling can cause strength over prediction and mesh dependency, the CSLI model is modified only at the interface between plies, where the cohesive elements between plies are replaced with tie-constraints so that no delamination is allowed. This model is basically composed of plies that are bonded together (Figure 3.6b). The results on the same four meshes are presented in Figure 3.8 with the curve labelled ‘CSPB’. It can be seen that the over-prediction and mesh-dependency reappear, especially on Mesh 3 and Mesh 4. Both the CLT model and the ‘CSPB’ model assume perfect-bonding between plies. The over-prediction by CSPB model, although being significant, is however much lower than that of the CLT model. This is due to the fact that these two models have different linear elasticity approximations. In the case of in-plane deformation, the CLT model assumes a constant strain across the thickness of the entire laminate (which neglects the transverse shear deformation of the plies); while the CSPB model has each ply modelled independently by a layer of elements (and thus the transverse shear deformation of each ply is allowed). The CSPB model has therefore more DoF across the thickness direction which allow a better representation of the through-thickness deformation and therefore a relatively more accurate prediction than the CLT model.

From the above study, it can be concluded that when composite laminates fail in the pull-out mode, neglecting delamination in the numerical models will lead to over-prediction and mesh dependency. Therefore, modelling delamination is critical for the strength prediction of laminates, even if delamination is not the major failure mode and in-plane failure is the primary concern. The CSLI model is used for the following size effect studies.

3.4 Size Effect Predictions

3.4.1 Sublamine-scaling thickness size effect

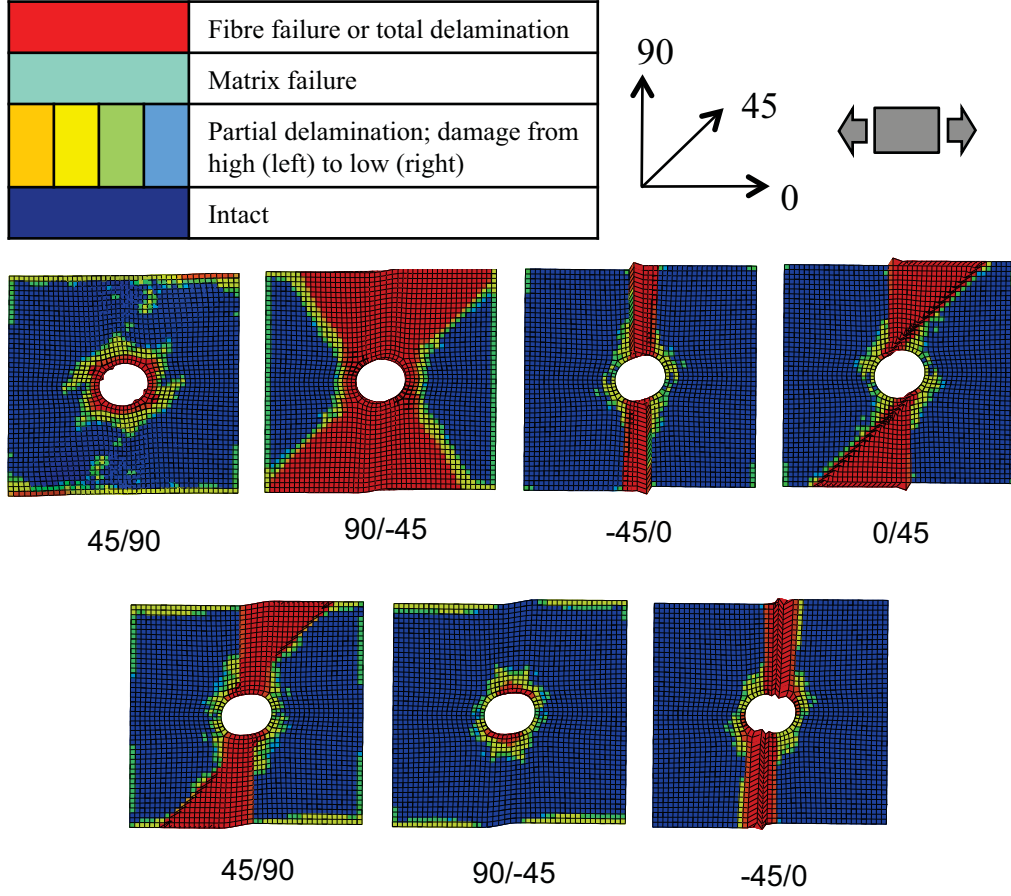
Table 3.2: Sublamine-scaling thickness size effect:
hole diameter $d = 3.175$ mm, lay-up:
 $[45/90/-45/0]_{ns}$

Thickness scaling	Strength (MPa)	
	Experiment	Simulation
$n = 1$	570 (cv ^a , 7.69%)	566.6 (-0.6% ^b)
$n = 2$	500 (cv, 3.95%)	510 (+2%)
$n = 4$	478 (cv, 3.09%)	499.4 (+4.48%)

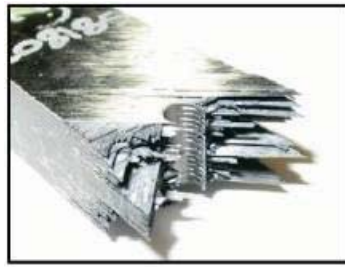
^a coefficient of variation

^b percentage difference compared to experimental data

CSLI models with the same mesh as Mesh 2 were created for the two thicker sublamine-scaled specimens, as shown in Table 3.2. It can be seen that for specimens of the same in-plane scales, strength decreases while sublaminae are blocked and the rate of decrease slows down as the number of sublaminae increases. Simulation results correctly predict these two phenomena and they are in good agreement with the experimental values. The failure modes of the two thicker sublamine-scaled specimens are both the pull-out mode and the predictions are similar as the one shown in the parts labelled ‘CSLI’ and ‘CSLI delamination’ in Figure 3.9. The delamination patterns on all the interfaces of the laminate of hole diameter $d=3.175$ mm and of lay-up $[45/90/-45/0]_{2s}$ are plotted in Figure 3.11a and compared against the experimental pull-out image in Figure 3.11b. Clear triangular-shaped delamination patterns are visible in the predictions which correlate well with the experimental delamination observation. Since the final load-drop of the sublamine-scaled specimens is primarily determined by fibre breaking of the 0 plies, a simple rule of mixture of the fracture toughness of



(a) Simulation delamination patterns on all the interfaces



Pull-out

(b) Experimental pull-out image [53]

Figure 3.11: Comparison between simulation and experiment of the pull-out delamination patterns for the sublaminates of $d=3.175$ mm and of lay-up $[45/90/-45/0]_{2s}$

all the 0 plies helps explain the thickness size effect. We can express the averaged fracture toughness of all the 0 plies of the laminate $[45/90/-45/0]_{ns}$ as:

$$\overline{G}_{fc} = \frac{G_{fc}^d n^d + G_{fc}^s n^s}{n^d + n^s} \quad (3.25)$$

where n^s is the number of single 0 plies and n^d is the number of double-blocked 0 plies. $G_{fc}^d \approx 2 \times G_{fc}^s$ as reported in [79]. It can be seen that as more sublaminates are blocked, $n^s = 2n - 2$ while $n^d = 2$ (two plies at the center). Thus:

$$\overline{G}_{fc} = \frac{2G_{fc}^d + (2n - 2)G_{fc}^s}{2n} = G_{fc}^s + \frac{G_{fc}^d - G_{fc}^s}{n} \quad (3.26)$$

Therefore as n increases, $\overline{G}_{fc}$ decreases towards G_{fc}^s and the rate of change reduces towards zero. This thickness dependence of average fracture toughness of the laminate causes a deterministic thickness size effect of the laminate strength.

It is worth mentioning that apart from the thickness dependence of translaminar fracture toughness, other factors also exist which contribute to this sublaminar-scaling thickness size effect. These factors include the Weibull size effect of translaminar strength [155] and the notch blunting and relaxation of stress concentration caused by the earlier longitudinal matrix splitting in the blocked 0 plies [82]. To the knowledge of the author, only two earlier studies have analysed the thickness effect of sublaminar-scaled cases, i.e., [56] and [146], and mixed results have been obtained on the prediction of sublaminar-scaling thickness size effect, where the latter work showed that a deterministic model (without taking into account the thickness dependence of translaminar fracture toughness) could not reproduce this thickness effect. Considering that the model in [146] accurately represented the matrix crack splitting using the Phantom Node Method, this indicates that capturing the matrix splitting alone is not enough to predict the thickness effect of the sublaminar-scaled cases; other aspects of the physics of fracture in composites need to be included, such as the size effects in translaminar strength [155] and fracture toughness [79]. The current model only employed the size effect of the translaminar fracture toughness since it is found to be more pronounced than that of the translaminar

strength; ideally, the size effect in strength could also be incorporated in the model to better represent the physics and improve the size effect predictions.

3.4.2 Ply-scaling thickness size effect

Table 3.3: Ply-scaling thickness size effect: hole diameter $d = 3.175$ mm, lay-up: $[45_m/90_m/-45_m/0_m]_s$

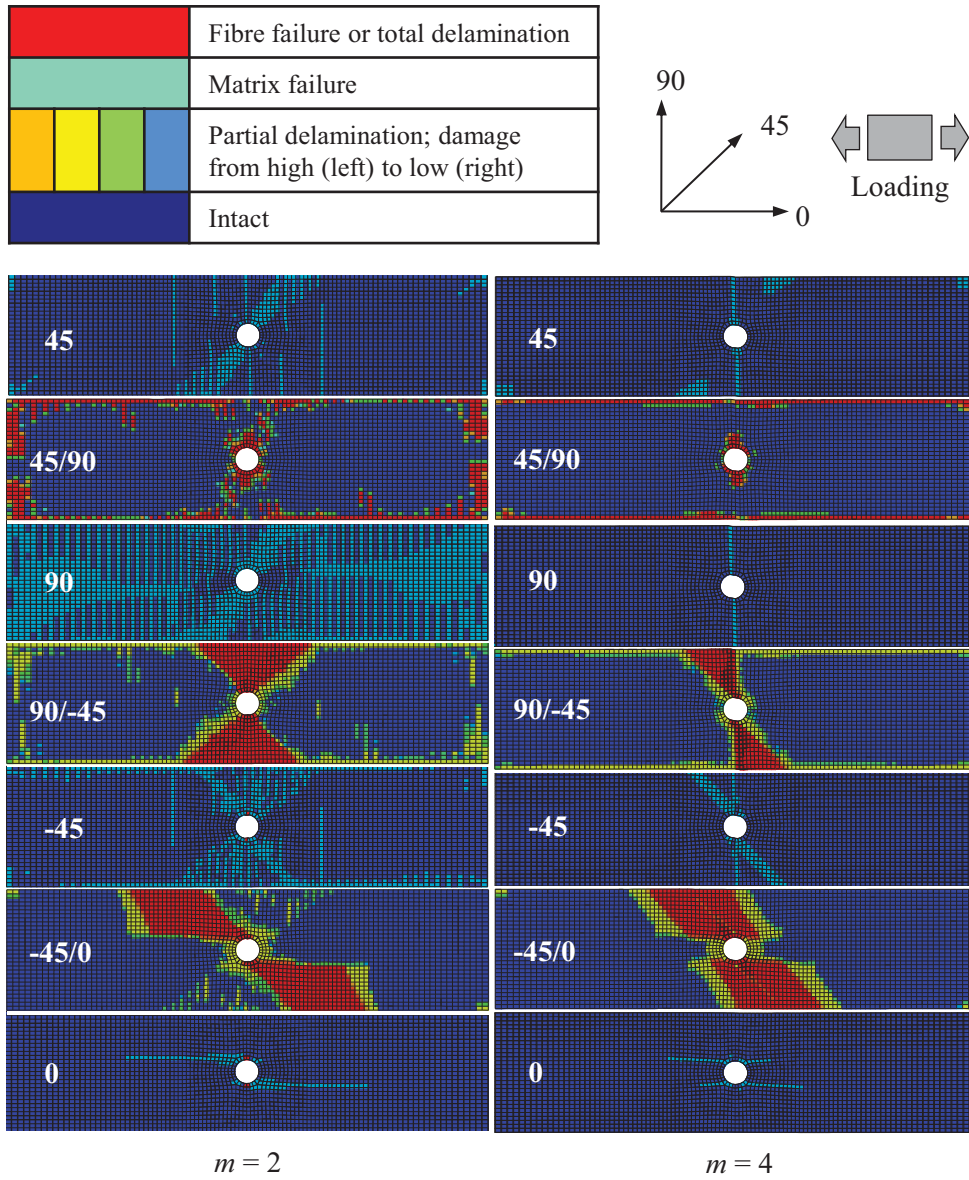
Thickness scaling	Strength (MPa)	
	Experiment	Simulation
$m = 1$	570 (cv, 7.69%)	566.6 (-0.6%)
$m = 2$	396 (cv, 5.18%)	410.8 (+3.7%)
$m = 4$	275 (cv, 5.56%)	278.4 (+1.2%)

The thickness size effect study of ply-scaled specimens is also studied. CSLI models were created for the two thicker ply-scaled cases, as shown in Table 3.3. The element size is determined by Equation 3.20 for the entire area of the laminate. For the area close to the hole, the value of the laminate half thickness h used in Equation 3.22 is assumed to be equal to $m \times t$ for the lay-up $[45_m/90_m/-45_m/0_m]_s$. For the area close to the grip, a more relaxed size restriction is used by assuming that the delamination between the 0 plies and other plies is most critical and h is set to be equal to $3 \times m \times t$. The thickness dependence of translaminar fracture toughness is assumed to be valid for thicker blocked plies and its value increases linearly with respect to its thickness:

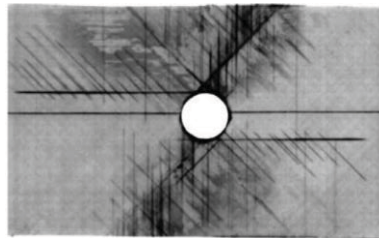
$$G_{fc}^m = m \times G_{fc}^s \quad (3.27)$$

where m is the number of plies blocked together and G_{fc}^m is the translaminar fracture toughness for the m blocked plies.

The strength predictions are very close to the experimental values (Table 3.3). The strength decreases significantly with respect to the increase in laminate thickness. It is observed in the simulation that the load-drop is accompanied by a widespread propagation of delamination along the $-45/0$ interface. This compares well with the experimental observations [53]. The failure patterns, immediately after the load-drop, for all the plies and the interfaces of the 2 mm thick and the 4 mm thick laminates, are plotted in Figure 3.12a and compared against an experimental X-Ray image in Figure 3.12b. It can be observed that the predicted delamination areas are within the boundaries defined by the matrix cracks from adjacent plies, which are in good agreement with those shown on the experimental image. A comparison of the delamination patterns on all the interfaces is shown in Figure 3.13, where the predictions are compared against the experimental X-Ray images from [128]. A close agreement between the simulation patterns and experimental images can be observed. In the simulation, both the 2 mm thick and the 4 mm thick specimens have the largest failure occurring at the $-45/0$ interface and they are of approximately the same area. The decrease in strength on thicker ply-scaled laminates can be explained by the fact that, for the same critical extent of delamination at the $-45/0$ ply interface, which requires a certain amount of strain energy released from the specimen, the thicker laminate is able to provide this amount of strain energy (from a larger volume of material) with a smaller strain energy density (which corresponds to smaller critical stresses and strains). This is reflected globally as smaller global strains and cross-sectional stresses at the point of load-drop for thicker ply-scaled laminates. Similarly as in the case of pull-out failure patterns (Figure 3.9, CSLI Final), the matrix cracks in the ± 45 plies are not all aligned correctly along the fibre directions (see Figure 3.12a, ± 45 failure patterns). This is the inherent limitation of the smeared crack model in capturing correctly the crack propagation directions when the mesh is not aligned with the crack path [43, 128, 144, 146]. The consequence of this limitation is however not found to be significant in this case, primarily because the most critical cracks are the 0 ply longitudinal matrix splits rather than the ± 45 cracks [53, 153].



(a) Simulation results



(b) X-Ray image of the experiment of the specimen with $m = 4$ [56]

Figure 3.12: Failure patterns for ply-scaled laminates of $d=3.175$ mm and of lay-up $[45_m/90_m/-45_m/0_m]_s$

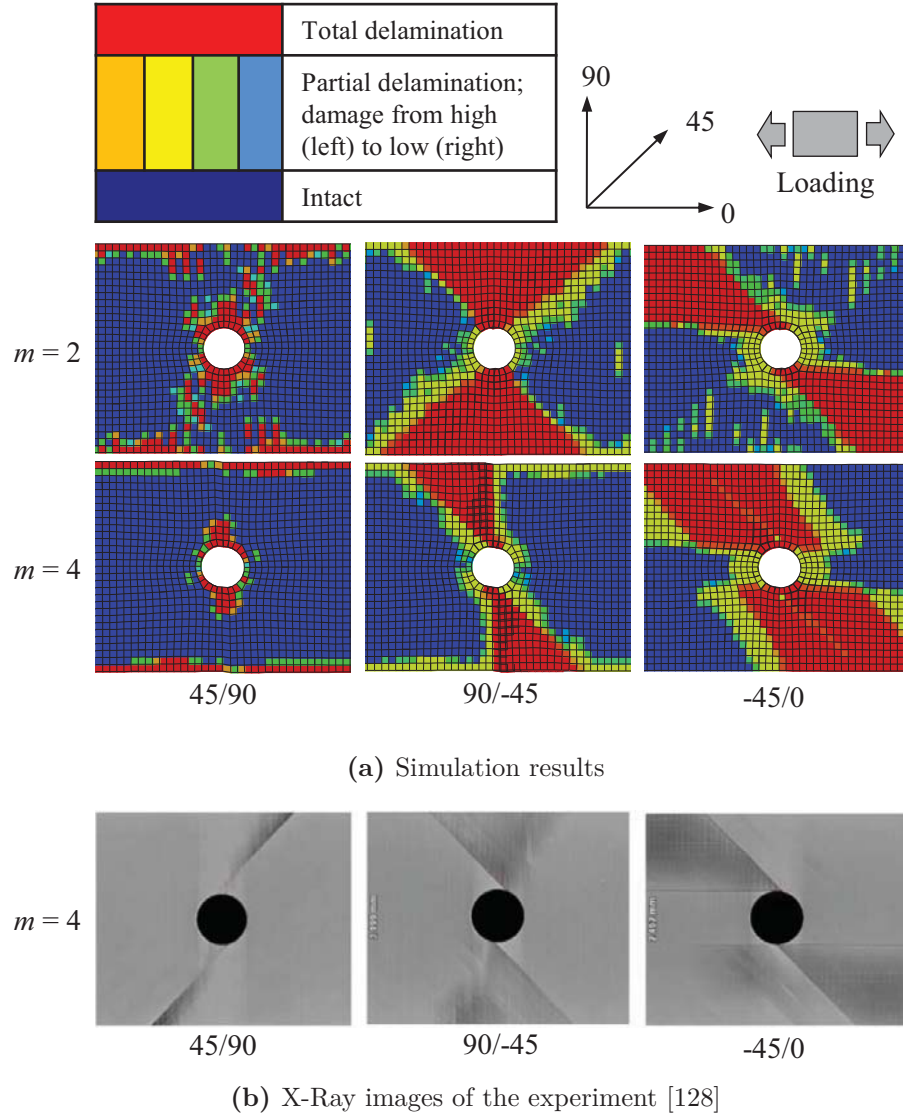


Figure 3.13: Comparison between simulation and experiment of the delamination patterns for the ply-scaled laminates of $d=3.175$ mm and of lay-up $[45_m/90_m/-45_m/0_m]_s$

3.4.3 In-plane size effect of sublaminated specimens

CSLI models were created for the two sublaminated 4 mm thick specimens with larger in-plane sizes as described in Table 3.4. The element size close to the hole is kept to be 0.57 mm to ensure that two cohesive elements are present in the cohesive zone in the area around the hole. Good agreement between simulations and experiments is obtained (Table 3.4). Since uniform and deterministic material properties are used in the numerical model, the predicted size effect is deterministic. This result supports the discussion in Section 3.2.1 which argues that the energetic size effect theory is the predominant theory for the size effect of open-hole composite laminates. The cohesive zone model is therefore an important component of constitutive models for composites.

Table 3.4: In-plane size effect of sublaminated specimens. Lay-up: $[45/90/-45/0]_{4s}$

Hole diameter	Strength (MPa)	
	Experiment	Simulation
d=3.175 mm	478 (cv, 3.09%)	499.4 (+4.48%)
d=6.35 mm	433 (cv, 2.03%)	432.9 (+0.02%)
d=12.7 mm	374 (cv, 1.01%)	357 (-4.5%)

3.4.4 In-plane size effect of ply-scaled specimens

The in-plane size effect of the ply-scaled specimens is also studied. Two ply-scaled 4 mm thick specimens with larger in-plane dimensions, as shown in Table 3.5, are modelled with the CSLI model. The element size is determined by Equation 3.20 for the entire area of the laminate. $h = 4 \times t$ for the area close to the hole and $h = 12 \times t$ for the area close to the grip.

The delamination failure mode is captured on all cases, where the load-drop is preceded by the extensive delamination on the -45/0 interfaces, similar as those shown in Figure 3.13. The strength predictions are summarized in Table 3.5. The predictions are in good agreement with the experiment; however, they show relative errors of around 10% on the two larger specimens, which are comparatively larger than those of the previous predictions (less than 5%). These errors could be caused by the fact that the stress concentration caused by the matrix cracks on the interfaces is not accurately captured by the smeared crack formulation used for matrix crack modelling in this study. In smeared crack formulations, matrix cracks are smeared out, so are the crack tips at the interfaces (Figure 3.14). Since the stress concentration caused by matrix crack tips at the interfaces is responsible for inducing delamination, smeared crack formulations have difficulties in reproducing this stress concentration accurately. Alternatively, it is possible to introduce the interaction into the material constitutive law for the cohesive elements as done in [2]. Another way of modelling the matrix crack/delamination interaction is to explicitly represent matrix cracks in the element formulation so that the stress concentration at the interface can be better modelled, such as the approaches used in [135, 146]. A nodal integration scheme was used on the cohesive elements in [146] to capture the matrix crack/delamination interaction in the situation of a major delamination [145]. The recent work of Fang et al. [49] has shown that it is important to not only explicitly represent the matrix cracks but also to represent the matrix crack tips at the interfaces to accurately capture the initiation of the local delamination. They have developed a breakable two-dimensional interface cohesive element that breaks into two parts when matrix cracks appear from adjacent plies. Chapter 4 proposes a novel method which models multiple plies and multiple cohesive cracks within one element, such that the element topology changes with respect to the development of cracks to accurately model the interaction between different cracks.

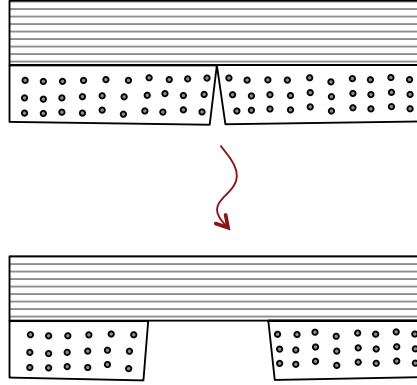
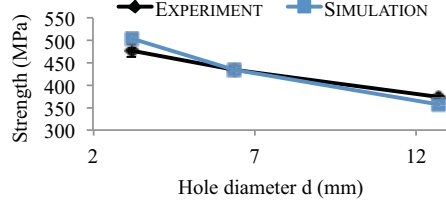


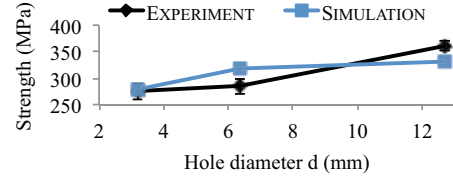
Figure 3.14: The smeared crack model approximates the sharp matrix crack tip into a blunted crack tip on the interface

Table 3.5: In-plane size effect of ply-scaled specimens. Lay-up: $[45_4/90_4/-45_4/0_4]_s$

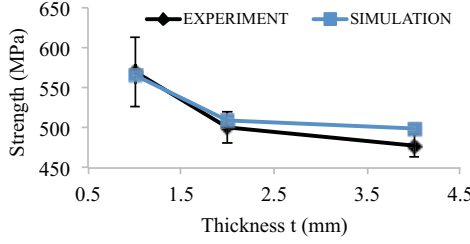
Hole diameter	Strength (MPa)	
	Experiment	Simulation
d=3.175 mm	275 (cv, 5.56%)	278.4 (+1.2%)
d=6.35 mm	285 (cv, 5.17%)	317.4 (+11.4%)
d=12.7 mm	362 (cv, 2.60%)	332.6 (-9.4%)



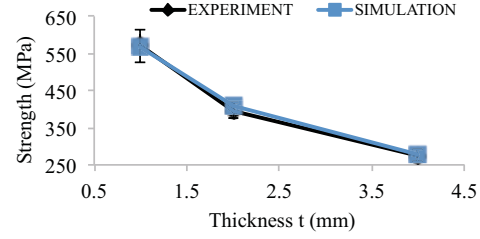
(a) Inplane size effect of sublaminatescaled cases



(b) Inplane size effect of ply-scaled cases



(c) Sublaminatescaling size effect



(d) Ply-scaling size effect

Figure 3.15: Summary of the predictions on strength size effects

3.4.5 Parametric sensitivity analysis

The majority of the specimens studied above fail in a pull-out or delamination failure mode, where the modelling of delamination is critical to the accurate prediction of the overall failure. In the in-plane tension loading, it is obvious that the shear strength τ_T^c ($=\tau_L^c$) and the mode II toughness G_{IIc} parameters of the interface are the most important parameters for delamination modelling using cohesive elements. Different values of these parameters have been used in the literature. Here we perform a parametric sensitivity analysis on both τ_T^c and G_{IIc} of the interface for two selected specimens, one failed by pulled-out and the other by delamination. Two different values are used for both τ_T^c and G_{IIc} in the simulation, and the results are summarized in Table 3.6. With different shear strength and toughness values, cohesive zone lengths are effectively different from those of the baseline models, and a new element size is calculated based on Equation 3.20 so as to allow a minimum to two elements in the cohesive zone for each case studied. A higher shear strength would reduce the cohesive zone length and therefore a finer mesh is needed to accurately capture the stress profile in the cohesive zone. A higher toughness, on the other hand,

would increase the cohesive zone and as a result a relatively coarser mesh can be employed. It is found that when this mesh requirement is satisfied, the failure pattern predictions remain unchanged as compared to those of the baseline models. It can be seen in Table 3.6 that the variation of these parameters have little influence on the strength prediction of the pulled-out failure mode, but have significant effect on that of the delamination failure mode. These results show that for specimens failed by pull-out, the CSLI model gives good strength prediction which is not sensitive to the interface properties (note that the modelling of delamination also ensures mesh-objectivity of the strength prediction, as analysed in Section 3.3). However, for specimens failed by delamination, both the strength and the toughness parameters of the cohesive elements are important for accurate predictions of the laminate strength. Established experimental procedures, such as the Double Cantilever Beam (DCB) test and Four point bend End-Notched Flexure (4ENF) test, exist for the measurement of interface fracture toughness parameters and the values used here are obtained from these tests. The interface strength parameters are however often assumed to be the typical strengths of the epoxy matrix material [56] or reduced to smaller values so as to relax the size constraints on cohesive elements [141] (which is the approach followed here). Therefore, the accuracy of the CSLI model on the quantitative strength predictions for the laminates failed by delamination (Section 3.4.2 and Section 3.4.4) is limited by the model's sensitivity to the interface shear strength parameter used.

It is worth-mentioning that the mesh-refinement with respect to the cohesive zone length according to Equation 3.20 only ensures the proper modelling of the cohesive zone and the associated amount of energy dissipation after the onset of delamination. The accurate prediction of delamination onset however may require a different level of mesh refinement; this could be the cause of the large over-predictions in the strength predictions of the delamination case. Theoretically, a further refined mesh would approximate better the sharp matrix crack, and therefore lead to a higher stress concentration on the interface, which would in turn trigger an earlier propagation of delamination and subsequently result in a lower laminate strength. However, further mesh refinement is computationally expensive and the right extent of refinement is difficult to be estimated. A more ac-

curate and efficient way of modelling would be to explicitly represent the sharp matrix crack as done in [135, 146] or by the method proposed in the next chapter.

Table 3.6: Parametric sensitivity analysis

Failure modes	Experimental strength (MPa)	Predicted strength (MPa) with		
		$\tau_T^c = 50$	$\tau_T^c = 90$	$\tau_T^c = 50$
		MPa $G_{IIc} = 0.631$ kJ/m ²	MPa $G_{IIc} = 0.631$ kJ/m ²	MPa $G_{IIc} = 1$ kJ/m ²
Pull-out	570 (cv, 7.69%)	566.6 (-0.6%)	589.9 (+3.5%)	591.2 (+3.7%)
Delamination	396 (cv, 5.18%)	410.8 (+3.7%)	490 (+23.7%)	514.8 (+30%)

3.5 Conclusion

This chapter presents a numerical study of the size effects observed on open-hole composite laminates. On the numerical side, it is shown that models without considering delamination give mesh-dependent and over-estimated strength predictions and therefore are not suitable for the failure modelling of composite laminates. The CSLI model, where the fibre breaking and matrix cracking are modelled by the smeared crack formulation and delamination is modelled by cohesive elements, gives accurate and mesh-independent strength predictions when a minimum of two cohesive elements are inside the numerical cohesive zone determined by the formula proposed in [58].

The thickness-dependence of the translaminar fracture toughness is accounted for in the numerical modelling, and the thickness size effect of the sublaminates is predicted by a deterministic model. Therefore, it is concluded that the thickness size effect is deterministic and the thickness-dependence of translaminar fracture toughness needs to be

taken into account in the failure modelling of composites. The in-plane size effect of the sublaminates scaled specimens is predicted with uniform input properties which shows that the in-plane size effect associated with fibre-breaking is deterministic. For the ply-scaled specimens which fail by delamination, good predictions on the failure mode and patterns are obtained with the CSLI model. However, the model has limitations on accurately representing the stress concentration caused by the matrix crack tip at the interface, and the laminate strength prediction is shown to be sensitive to the interface shear strength used for the cohesive elements. Therefore, the quantitative prediction by this model on the strengths of the laminates failed by delamination is limited.

A guideline of choosing appropriate numerical methods can be proposed:

- (a) if the laminate fails in a brittle manner with minimal delamination, then the CLT model can be used [29].
- (b) if there is delamination but the load-drop is still caused by ply failure (such as pull-out), then the CSLI model can be used.
- (c) if the laminate fails by widespread delamination between plies before all plies break, then the matrix crack/delamination interaction becomes critical. In this case, the CSLI model can give a qualitative analysis of the failure mode and the trend of the size effect, but have limitations on quantitative strength predictions. A model which better captures the matrix crack/delamination interaction such as the ones proposed in [2, 135, 146] and Chapter 4 can be used for higher accuracy.

In the case where the failure mode of the laminate cannot be estimated before analysis, the CSLI model can be firstly used to predict the failure mode of the specimen. If the predicted failure mode is brittle failure or pull-out, then the predicted strength can be considered reliable. However if the predicted failure mode is delamination, then the predicted strength is no longer accurate and a revised analysis with a more advanced model for matrix crack/delamination interaction is needed for a more reliable strength prediction. Furthermore, the predicted matrix crack directions by the smeared crack model can be biased by the mesh orientation. Although the effect of this limitation is not found to be significant in this study, it limits the applicability of the model in generic situations where the mesh

is not necessarily aligned with the critical cracks [43, 128, 144, 146]. In such cases, a method which explicitly represent matrix cracks in a mesh-independent manner is required. A new numerical method will be presented in the next chapter for these purposes.

3.6 Acknowledgement

The first author gratefully acknowledges the research scholarship from National University of Singapore and would like to thank Dr. M. Ridha and Dr. X. S. Sun for helpful and useful discussions on the topics of this chapter.

3.7 Publications

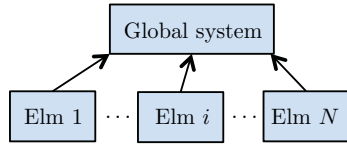
The content of this chapter has produced the following publications:

- B. Y. Chen, T. E. Tay, P. M. Baiz, and S. T. Pinho. Size effects in progressive damage of notched and holed composites. In *Proceedings of 18th international conference on composite materials, Jeju, Korea Aug, 2011*
- B. Y. Chen, T. E. Tay, P. M. Baiz, and S. T. Pinho. Numerical analysis of size effects on open-hole tensile composite laminates. *Composites Part A: Applied Science and Manufacturing*, 47(0):52 – 62, 2013

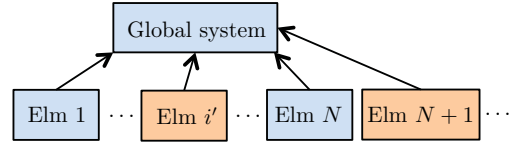
Chapter 4

Floating Node Method

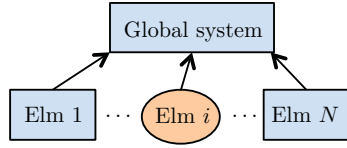
In this chapter, a new method is presented for the explicit modelling of discontinuities within a finite element, which is shown to possess some advantages in the modelling of complex damage and failure scenarios over existing methods (see Chapter 2). Figure 4.1 compares the computational architectures of the reviewed methods in Chapter 2 which explicitly represent the discontinuities, namely the remeshing method, XFEM and the PNM. The remeshing method is essentially a re-application of the standard FEM, according to an updated geometry and with carefully optimized element size and shape; the element formulation and assembly procedure remain the same as in FEM [4, 8, 9, 26, 74, 75, 77, 86, 93–95, 101, 122–124, 134]. XFEM [7, 21, 23, 51, 59, 63, 65, 73, 92, 97, 116, 117, 129, 130] enriches the element interpolation functions to explicitly represent the displacement fields in the presence of discontinuities, i.e., each element is changed to a different formulation when a crack propagates within its domain. The PNM [6, 33, 46, 49, 50, 57, 83, 89–91, 113, 118, 126, 127, 143, 146, 147] introduces one more level of hierarchy in the assembly architecture; each damaged element is replaced with two or more sub-elements, whose system matrices are firstly assembled within the original element system equation, before being assembled together as one element into the global system. In summary, remeshing models crack propagation globally (i.e., in the main programme) through updating the domain geometry while XFEM and PNM do it locally (i.e., in the local element subroutine) through modifying the element formulation. With the latter approach, the global quantities (ge-



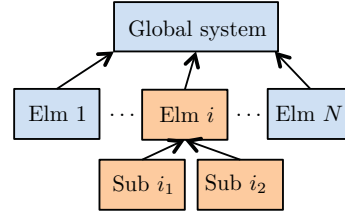
(a) Standard FEM architecture; local system matrices of each element is directly assembled to the global system.



(b) Remeshing architecture; damaged element (Elm i) is replaced with new elements (Elm i' , Elm $N + 1$, ...).



(c) XFEM architecture; XFEM enriches the displacement interpolation of the element.



(d) PNM architecture; damaged element (Elm i) is replaced with new (sub-)elements (Sub i_1 , Sub i_2); local system matrices of sub-elements are first assembled to the original element (Elm i) system equation, and then to the global system.

Figure 4.1: Comparison of assembly architectures of different methods

ometry, mesh and element connectivities) remain unchanged as opposed to the general case in remeshing. Extra databases used in remeshing for the storage, update and transfer (from the old mesh to the new one) of solution dependent variables are not needed as they can be directly updated in the local element. The local approach of XFEM and PNM is particularly advantageous in the modelling of composite structures where the number of cracks is generally large and the geometries of crack networks can be very complex. In such cases, updating the crack geometries and the mesh in the main programme (as done in remeshing) is not an efficient approach of modelling and its capability can be limited; while with XFEM and PNM, more cracks can be effectively modelled with more enriched or partitioned local elements, whose calculations are standardized and remain relatively simple. These advantages of XFEM and PNM have lead to several applications of these methods in the progressive failure modelling of composites [50, 65, 73, 83, 135, 143, 146, 147].

We propose a new method, a Floating Node Method (FNM), which is similar to the PNM in terms of computational architecture, where discontinuities are modelled locally within an element by partitioning the original element into sub-elements (Figure 4.1d). The FNM has the following advantages over the PNM:

- the integration of sub-elements is simpler, as it does not require numerical integration over *only part* of a domain after transforming to the natural space;
- it can incorporate weak discontinuities and cohesive cracks readily;
- it is ideally suited for the representation of multiple and complex networks of discontinuities, such as the multiple T-shaped cracks in Section 4.4;
- it does not introduce an error on the crack geometry when mapping from physical to natural coordinates; and
- it is conceptually simpler.

In terms of solution, the FNM is equivalent to local remeshing within the cracked element; but crucially, the FNM:

- is structurally different in computational architecture;
- is computationally more efficient than remeshing (eg. no modification of global quantities such as initial geometry, mesh and element connectivities) when modelling a large number of cracks and complex crack networks, such as those typically found in composite laminates;
- can be readily implemented in relatively closed FE codes (using user-defined elements); and
- allows sub-elements within an original element to readily share information.

When applying to the progressive failure modelling of composites, the FNM:

- can model multiple plies and interfaces of a composite laminate within a user-defined element;
- can readily share information between different plies and interfaces within the local element;
- can model both matrix cracking and delamination within an element; and
- can predict the interaction between matrix cracking and delamination accurately.

Due to the similarities of the proposed FNM with the existing PNM, we review the PNM in Section 4.1, before introducing the proposed FNM in Section 4.2. Validation examples are presented in Section 4.3, and applications are shown in Section 4.4, before concluding in Section 4.5.

4.1 Overview of the Phantom Node Method

4.1.1 Introduction

The static equilibrium of a body with volume Ω under body forces with density $\mathbf{f}$ (acting on Ω) and traction $\mathbf{t}$ acting on the boundary Γ_Ω can be

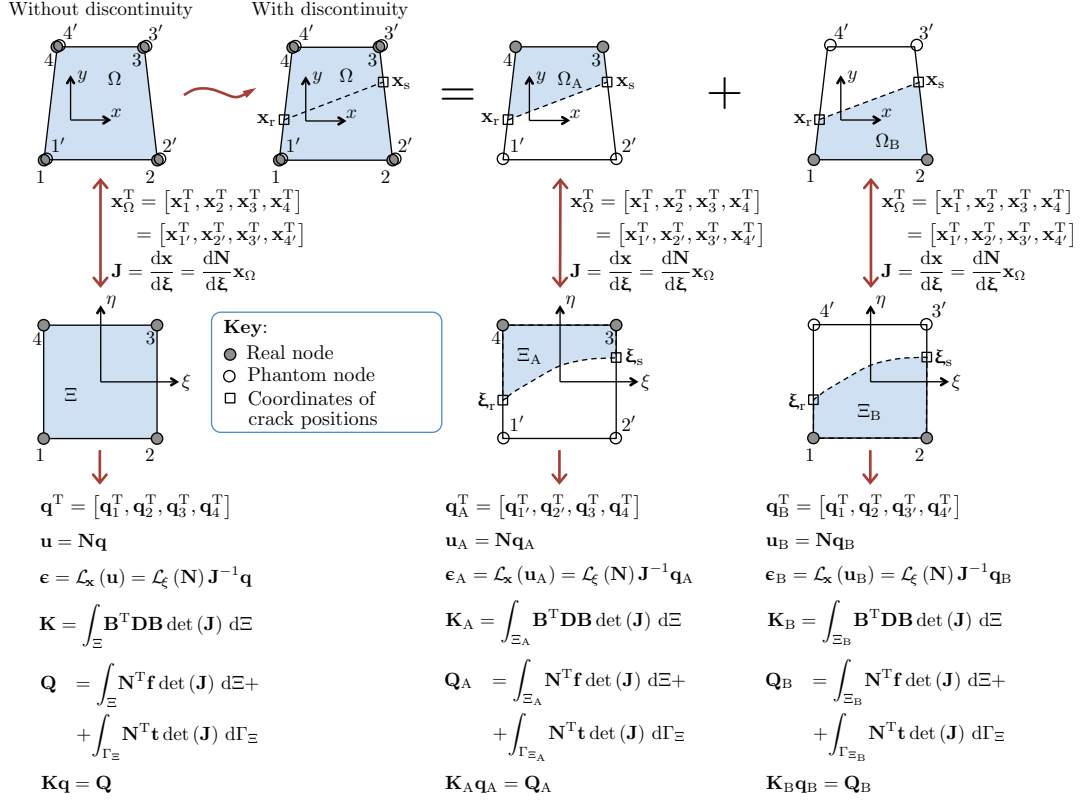


Figure 4.2: Phantom Node Method

expressed in the weak form as:

$$\int_{\Omega} \boldsymbol{\epsilon}^T(\mathbf{v}) \boldsymbol{\sigma}(\mathbf{u}) d\Omega = \int_{\Omega} \mathbf{v}^T \mathbf{f} d\Omega + \int_{\Gamma_{\Omega}} \mathbf{v}^T \mathbf{t} d\Gamma_{\Omega} \quad (4.1)$$

where $\mathbf{u}$ is the displacement vector; $\mathbf{v}$ is the test function; $\boldsymbol{\epsilon}$ is the strain tensor, related to $\mathbf{u}$ through the differential operator relative to Cartesian coordinates $\mathcal{L}_{\mathbf{x}}$ as $\boldsymbol{\epsilon} = \mathcal{L}_{\mathbf{x}}(\mathbf{u})$; and $\boldsymbol{\sigma}$ is the stress tensor, related to the strains through Hook's law as $\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon}$, with $\mathbf{D}$ being the constitutive tensor.

In the PNM, each real node i (characterised by its nodal coordinates $\mathbf{x}_i$ and associated DoF $\mathbf{q}_i$) is accompanied by a phantom node i' (characterised by the same nodal coordinates $\mathbf{x}_{i'}$, i.e. $\mathbf{x}_{i'} \equiv \mathbf{x}_i$, and associated DoF $\mathbf{q}_{i'}$; in general, $\mathbf{q}_{i'} \neq \mathbf{q}_i$). Figure 4.2 shows on the top left an example of a finite element suitable for the PNM; the positions of the real nodes ($[\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4]$) coincide with the positions of the phantom nodes ($[\mathbf{x}_{1'}, \mathbf{x}_{2'}, \mathbf{x}_{3'}, \mathbf{x}_{4'}]$).

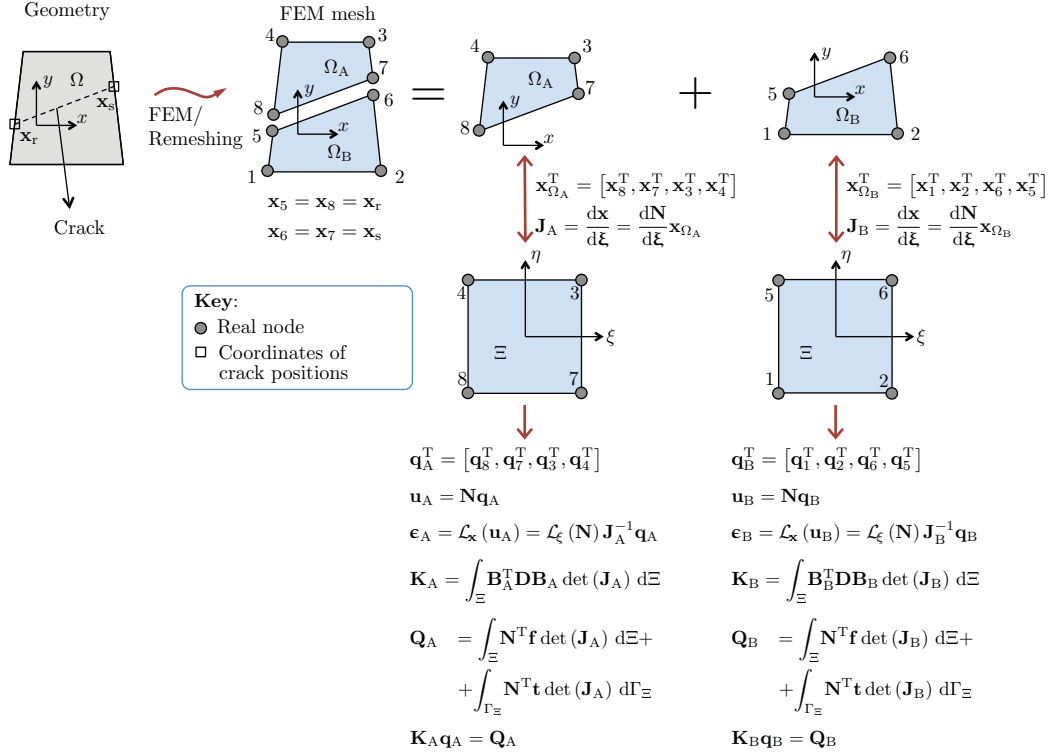


Figure 4.3: FEM or remeshing

4.1.2 Without a discontinuity

If there is no discontinuity to be modelled by the element (e.g. before failure), the element is simply a standard finite element. The vector of nodal coordinates is given as $\mathbf{x}_{\Omega}$. In the case of Figure 4.2,

$$\mathbf{x}_{\Omega}^T = [\mathbf{x}_1^T, \mathbf{x}_2^T, \mathbf{x}_3^T, \mathbf{x}_4^T] = [\mathbf{x}_{1'}^T, \mathbf{x}_{2'}^T, \mathbf{x}_{3'}^T, \mathbf{x}_{4'}^T]. \quad (4.2)$$

The Jacobian of the transformation from physical coordinates $\mathbf{x}$ to natural coordinates $\boldsymbol{\xi}$ is:

$$\mathbf{J} = \frac{d\mathbf{x}}{d\boldsymbol{\xi}} = \frac{d\mathbf{N}}{d\boldsymbol{\xi}} \mathbf{x}_{\Omega}, \quad (4.3)$$

where $\mathbf{N}$ is a standard matrix of shape functions. Assuming an isoparametric formulation, the displacement field $\mathbf{u}$ is related to the *real* DoF $\mathbf{q}$ through the same matrix $\mathbf{N}$, i.e.:

$$\mathbf{u} = \mathbf{N} \mathbf{q}. \quad (4.4)$$

In the case of Figure 4.2, $\mathbf{q}^T = [\mathbf{q}_1^T, \mathbf{q}_2^T, \mathbf{q}_3^T, \mathbf{q}_4^T]$. The strains can also be expressed in terms of the real DoF $\mathbf{q}$ as:

$$\boldsymbol{\epsilon} = \mathcal{L}_{\mathbf{x}}(\mathbf{u}) = \mathcal{L}_{\xi}(\mathbf{N}) \mathbf{J}^{-1} \mathbf{q} = \mathbf{B} \mathbf{q} \quad \text{with} \quad \mathbf{B} = \mathcal{L}_{\xi}(\mathbf{N}) \mathbf{J}^{-1}, \quad (4.5)$$

leading to the stiffness matrix:

$$\mathbf{K} = \int_{\Xi} \mathbf{B}^T \mathbf{D} \mathbf{B} \det(\mathbf{J}) \, d\Xi, \quad (4.6)$$

where Ξ is the integration domain (in natural coordinates) corresponding to Ω (in physical coordinates), and to the vector of nodal forces:

$$\mathbf{Q} = \int_{\Xi} \mathbf{N}^T \mathbf{f} \det(\mathbf{J}) \, d\Xi + \int_{\Gamma_{\Xi}} \mathbf{N}^T \mathbf{t} \det(\mathbf{J}) \, d\Gamma_{\Xi} \quad (4.7)$$

where Γ_{Ξ} is similarly the boundary corresponding to Γ_{Ω} . The weak form of equilibrium, Equation 4.1, becomes then:

$$\mathbf{K} \mathbf{q} = \mathbf{Q}. \quad (4.8)$$

Equation 4.8 involves only the real DoF; for completeness, the phantom DoF can be defined using constraint equations so that they coincide with the real DoF; otherwise, they can be removed from the system of equations during the assembly process.

4.1.3 With a strong discontinuity

A discontinuity in the element can be predicted through generic stress-based criteria or otherwise; Section 4.2.9 delves in more detail into propagation models and criteria. Once a discontinuity is predicted, the points at which it crosses the element are defined (points with coordinates $\mathbf{x}_r$ and $\mathbf{x}_s$ in Figure 4.2). In this case, the vector of nodal coordinates $\mathbf{x}_{\Omega}$ is still given by Equation 4.2, and the Jacobian $\mathbf{J}$ is still given by Equation 4.3. However, the element is split in two sub-elements Ω_A and Ω_B , as indicated in Figure 4.2, and the displacement $\mathbf{u}$ is no longer related to the real DoF $\mathbf{q}$ by Equation 4.4. Instead, the displacements $\mathbf{u}_A$ and $\mathbf{u}_B$, in sub-elements Ω_A and Ω_B respectively, are interpolated separately from the respective

DoF:

$$\mathbf{u}_A = \mathbf{N}\mathbf{q}_A \quad \text{and} \quad \mathbf{u}_B = \mathbf{N}\mathbf{q}_B. \quad (4.9)$$

In the case of Figure 4.2, $\mathbf{q}_A^T = [\mathbf{q}_1^T, \mathbf{q}_2^T, \mathbf{q}_3^T, \mathbf{q}_4^T]$ and $\mathbf{q}_B^T = [\mathbf{q}_1^T, \mathbf{q}_2^T, \mathbf{q}_3^T, \mathbf{q}_4^T]$. The strains then become:

$$\boldsymbol{\epsilon}_A = \mathcal{L}_x(\mathbf{u}_A) = \mathcal{L}_x(\mathbf{N}) \mathbf{J}^{-1} \mathbf{q}_A = \mathbf{B} \mathbf{q}_A, \quad \mathbf{B} = \mathcal{L}_x(\mathbf{N}) \mathbf{J}^{-1} \quad (4.10)$$

$$\boldsymbol{\epsilon}_B = \mathcal{L}_x(\mathbf{u}_B) = \mathcal{L}_x(\mathbf{N}) \mathbf{J}^{-1} \mathbf{q}_B = \mathbf{B} \mathbf{q}_B. \quad (4.11)$$

Note that the $\mathbf{B}$ matrix (Equations 4.10 and 4.11) is the same for both sub-elements Ω_A and Ω_B ; their respective stiffness matrices are obtained by integrating over the corresponding domains in the natural space, i.e.:

$$\mathbf{K}_A = \int_{\Xi_A} \mathbf{B}^T \mathbf{D} \mathbf{B} \det(\mathbf{J}) \, d\Xi \quad \text{and} \quad \mathbf{K}_B = \int_{\Xi_B} \mathbf{B}^T \mathbf{D} \mathbf{B} \det(\mathbf{J}) \, d\Xi. \quad (4.12)$$

Note that the integrands for sub-elements Ω_A and Ω_B in Equation 4.12 are the same; only the integration domains (Ξ_A and Ξ_B respectively) are different. Similarly, the force vectors are:

$$\mathbf{Q}_A = \int_{\Xi_A} \mathbf{N}^T \mathbf{f} \det(\mathbf{J}) \, d\Xi + \int_{\Gamma_{\Xi_A}} \mathbf{N}^T \mathbf{t} \det(\mathbf{J}) \, d\Gamma_{\Xi} \quad \text{and} \quad (4.13)$$

$$\mathbf{Q}_B = \int_{\Xi_B} \mathbf{N}^T \mathbf{f} \det(\mathbf{J}) \, d\Xi + \int_{\Gamma_{\Xi_B}} \mathbf{N}^T \mathbf{t} \det(\mathbf{J}) \, d\Gamma_{\Xi}, \quad (4.14)$$

where again the integrands are the same for both sub-elements but the integration domains differ. From Equation 4.1, the equations of equilibrium for the two sub-elements are:

$$\mathbf{K}_A \mathbf{q}_A = \mathbf{Q}_A \quad \text{and} \quad \mathbf{K}_B \mathbf{q}_B = \mathbf{Q}_B. \quad (4.15)$$

Finally, the equation of equilibrium of the Phantom Node (PN) element is the assembly of those of the two sub-elements:

$$\mathbf{K} \mathbf{q} = \mathbf{Q}. \quad (4.16)$$

where

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_A & \\ & \mathbf{K}_B \end{bmatrix}, \quad (4.17)$$

and

$$\mathbf{q}^T = [\mathbf{q}_A^T, \mathbf{q}_B^T], \quad (4.18)$$

and

$$\mathbf{Q}^T = [\mathbf{Q}_A^T, \mathbf{Q}_B^T]. \quad (4.19)$$

4.1.4 With other types of discontinuities and scenarios

Weak discontinuities can be modelled with the PNM, for instance, by using appropriate constraints between the displacements of the two sub-elements along the discontinuous interface Γ :

$$\mathbf{u}_A(\Gamma) = \mathbf{u}_B(\Gamma) \quad \Rightarrow \quad \mathbf{N}(\Gamma)(\mathbf{q}_A - \mathbf{q}_B) = 0. \quad (4.20)$$

The above constraint equation involves all the DoF, i.e. both the real DoF and the phantom DoF (note that the shape function interpolations in the PNM along the discontinuity interface are all non-zero (Figure 4.2), thus all terms in $\mathbf{q}_A$ and $\mathbf{q}_B$ are used to interpolate the displacements of the two sub-elements along that surface). The number of DoF for modelling weak discontinuities is therefore the same as that for modelling strong discontinuities.

Similarly, cohesive cracks can be modelled for instance by introducing a suitable non-linear forcing function between $\mathbf{u}_A(\Gamma)$ and $\mathbf{u}_B(\Gamma)$ of Equation 4.20, which would then also depend on both the real and the phantom DoF sets as in the case for weak discontinuity. Therefore, in the case of Figure 4.2, the cohesive traction directly affects eight DoF sets; while in the FEM/remeshing method, the cohesive crack is explicitly represented by a linear cohesive element which contains only four DoF sets. As a result, more coupling terms are present in the eventual stiffness matrix (which is less favourable for calculation) in the PNM than in the FEM/remeshing

method.

While the PNM is well suited for modelling an individual discontinuity within an element, extensions which allow modelling more complex networks of discontinuities (such as multiple intersecting cohesive cracks) become considerably more challenging.

4.1.5 Comparison with other methods

Comparison with XFEM

The PNM has been shown to be equivalent to XFEM with Heaviside enrichment [5, 127]. The PNM displacement solution over a cracked element can be expressed as:

$$\mathbf{u}(\mathbf{x}) = \begin{cases} \mathbf{u}_A(\mathbf{x}) & \text{if } \mathbf{x} \in \Omega_A; \\ \mathbf{u}_B(\mathbf{x}) & \text{if } \mathbf{x} \in \Omega_B, \end{cases} \quad (4.21)$$

which is equivalent to:

$$\mathbf{u} = H \mathbf{u}_A + (1 - H) \mathbf{u}_B, \quad (4.22)$$

where

$$H(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_A; \\ 0 & \text{if } \mathbf{x} \in \Omega_B. \end{cases} \quad (4.23)$$

After rearrangement, $\mathbf{u}$ becomes:

$$\mathbf{u} = \mathbf{u}_B + H(\mathbf{u}_A - \mathbf{u}_B). \quad (4.24)$$

Since the two superposing sub-elements in the PNM are identical in shape in physical coordinates (Figure 4.2), they share the same interpolation functions in physical coordinates, i.e., $\mathbf{u}_A(\mathbf{x}) = \mathbf{\Phi}(\mathbf{x}) \mathbf{q}_A$ and $\mathbf{u}_B(\mathbf{x}) = \mathbf{\Phi}(\mathbf{x}) \mathbf{q}_B$, where $\mathbf{\Phi}$ is the shape function matrix in physical coor-

dinates; thus, $\mathbf{u}$ can be expressed as:

$$\mathbf{u}(\mathbf{x}) = \Phi(\mathbf{x}) \mathbf{q}_B + H(\mathbf{x}) \Phi(\mathbf{x}) (\mathbf{q}_A - \mathbf{q}_B) \quad (4.25)$$

Denoting $\mathbf{q}^\circ = \mathbf{q}_B$ and $\mathbf{q}^* = \mathbf{q}_A - \mathbf{q}_B$, we obtain the standard XFEM expression of the displacement solution [21, 92]:

$$\mathbf{u}(\mathbf{x}) = \Phi(\mathbf{x}) \mathbf{q}^\circ + H(\mathbf{x}) \Phi(\mathbf{x}) \mathbf{q}^* \quad (4.26)$$

where $\mathbf{q}^\circ$ and $\mathbf{q}^*$ correspond to the standard and enriched DoF, respectively. Therefore, the PNM and the XFEM with Heaviside enrichment are equivalent variations of each other in terms of displacement approximation. However, the two methods differ in terms of methodology and implementation in that the PNM duplicates the number of nodes to form an assembly of two superposing sub-elements, while XFEM keeps the number of nodes constant; and that the PNM uses only standard FEM shape functions for the displacement approximation, while XFEM employs extra enrichment functions for it; and that the PNM associates only the displacement DoF with each node, while XFEM assigns extra DoF at each node. The PNM is easier to be implemented in existing FEM packages than XFEM as it is more compatible with the conventional FEM architecture, where it is easier to duplicate an element than introducing enrichment functions into its formulation.

Comparison with FEM/remeshing

The equations of equilibrium of the PNM (Equation 4.15, see also Figure 4.2) are fundamentally different from those of an equivalent finite element mesh representing the same discontinuity explicitly (Figure 4.3). As shown in Figure 4.3, the two elements on Ω_A and Ω_B in the FEM or remeshing are transformed to the natural coordinates for integration through different Jacobians and full integration is performed on the element domain, as opposed to the approach in the PNM (Figure 4.2). Furthermore, the two elements on Ω_A and Ω_B in remeshing are created according to the global geometry and are assembled directly into the global system equation; those in the PNM are created according to the local element's damage status

and are assembled locally within the PN element (c.f. Figure 4.1b and Figure 4.1d). In this way, the original mesh and the element connectivities remain unchanged in the PNM and no geometrical update and mesh regeneration is needed, as opposed to in remeshing [4, 8, 9, 26, 74, 75, 77, 86, 93–95, 101, 122–124, 134].

Figure 4.2 highlights that the PNM introduces in general an error in the representation of the geometry of discontinuities, when mapping the discontinuities from the physical to the natural space. For instance, a straight crack will in general not be straight in the natural space; this is due to the fact that the shape functions of a quadrilateral element contain the term $\xi\eta$, making the interpolation nonlinear within the element domain. In the actual calculation, a curved crack is often approximated as a straight crack so as to form polygonal sub-domains for numerical integrations [51]. This is also mentioned in [113] and demonstrated with an example in Appendix B. The integration over the partial domains Ξ_A and Ξ_B (Figure 4.2) constitutes another source of complexity. It should be noted that as XFEM is equivalent to the PNM, the same error and complexity exist in XFEM in the transformation and integration of cracked elements [51]; however, they are not associated with the FEM or remeshing method as the crack coincides with edges of the elements along which the interpolation is always linear, and integration is always performed on the full element domain (Figure 4.3).

4.2 Floating Node Method

4.2.1 Overview of the approach

Section 4.1 shows that, in the PNM, phantom DoF do not need an associated position vector before the element is split (Section 4.1.2). After the element is split, the nodal position associated with the phantom DoF is not the most suitable in terms of transformation to the natural coordinate system nor in terms of integration (Section 4.1.4). In fact, when a crack needs to be propagated within an element, the nodal coordinates of new crack tips are calculated from the failure theory, and thus they become directly available to define the two material domains separated by the

crack, which are the actual integration domains of the two sub-elements in Figure 4.2. Transformation and integration directly performed on these material domains, rather than on the partial domains of the PN element, would constitute a more straightforward choice of calculation.

This observation forms the basis of the FNM, which we detail in this section. In the FNM, see Figure 4.4, each *real* node i is characterised by its nodal coordinates $\mathbf{x}_i$ and associated DoF $\mathbf{q}_i$; in addition, the element contains a suitable number of *floating* DoF, i.e., DoF without pre-defined associated nodal position vectors. For implementation in existing FE codes, DoF cannot in general be defined independently without associated nodal coordinates. Therefore, in the current implementation, floating DoF are defined in the form of floating nodes where any position vector, such as the origin, can be associated to their DoF (thus the name *floating nodes*); these position vectors are not used in the analysis. This will be detailed in Section 4.2.9. Figure 4.4 shows an example of such an element with four real nodes (1 to 4, defining the geometry through $\mathbf{x}_1, \dots, \mathbf{x}_4$ and with associated DoF sets $\mathbf{q}_1, \dots, \mathbf{q}_4$) and four further floating nodes with associated sets of floating DoF ($\mathbf{q}_5, \dots, \mathbf{q}_8$) required to represent a strong discontinuity.

4.2.2 Without a discontinuity

If there is no discontinuity to be modelled by the element (e.g. before failure), the formulation is the same as in the standard FEM or the PNM. The vector of nodal coordinates is given as $\mathbf{x}_\Omega$. In the case of Figure 4.4, $\mathbf{x}_\Omega$ is given by the same equation as in the PNM, Equation 4.2. The Jacobian of the transformation is equally given by Equation 4.3. Assuming an isoparametric formulation, the displacement field $\mathbf{u}$ is also related to the real DoF $\mathbf{q}$ through Equation 4.4. The strains are given by Equation 4.5, and the stiffness matrix and vector of nodal forces by Equations 4.6 and 4.7 respectively, thus leading to the equilibrium equations expressed in Equation 4.8.

4.2.3 With a strong discontinuity

Once a discontinuity in the element is predicted (see Section 4.2.9), and thus the coordinates of the points which define the intersection between

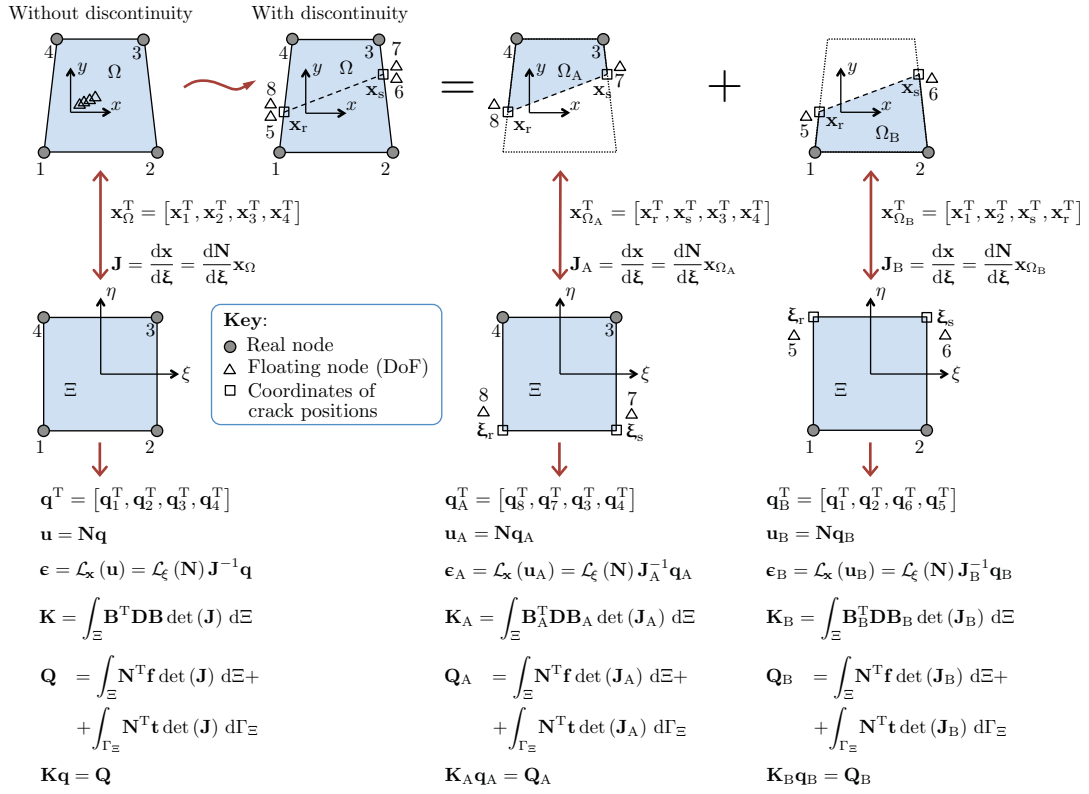


Figure 4.4: Floating Node Method

the discontinuity and the element ($\mathbf{x}_r$ and $\mathbf{x}_s$ in Figure 4.4) defined, the proposed formulation becomes different from the PNM presented in Section 4.1.3. As the discontinuity is defined, the element is split in two or more sub-elements (depending on the discontinuity). Without loss of generality, we will illustrate in detail firstly the most typical situation in which the element is split in two sub-elements, Ω_A and Ω_B (Figure 4.4).

Unlike in the PNM, we make direct use of the coordinates of crack boundaries, $\mathbf{x}_r$ and $\mathbf{x}_s$, and define a vector of nodal coordinates for each sub-element Ω_A and Ω_B , $\mathbf{x}_{\Omega_A}$ and $\mathbf{x}_{\Omega_B}$ respectively. For the case in Figure 4.4, these would be (c.f. Equation 4.2):

$$\mathbf{x}_{\Omega_A}^T = [\mathbf{x}_r^T, \mathbf{x}_s^T, \mathbf{x}_3^T, \mathbf{x}_4^T] \quad \text{and} \quad \mathbf{x}_{\Omega_B}^T = [\mathbf{x}_1^T, \mathbf{x}_2^T, \mathbf{x}_s^T, \mathbf{x}_r^T]. \quad (4.27)$$

Unlike in the PNM, each sub-element has then a separate Jacobian (c.f. Equation 4.3),

$$\mathbf{J}_A = \frac{d\mathbf{x}}{d\xi} = \frac{d\mathbf{N}}{d\xi} \mathbf{x}_{\Omega_A} \quad \text{and} \quad \mathbf{J}_B = \frac{d\mathbf{x}}{d\xi} = \frac{d\mathbf{N}}{d\xi} \mathbf{x}_{\Omega_B}. \quad (4.28)$$

The displacements $\mathbf{u}_A$ and $\mathbf{u}_B$, in sub-elements Ω_A and Ω_B respectively, are interpolated separately from the respective DoF. Assuming an isoparametric formulation,

$$\mathbf{u}_A = \mathbf{N}\mathbf{q}_A \quad \text{and} \quad \mathbf{u}_B = \mathbf{N}\mathbf{q}_B. \quad (4.29)$$

$\mathbf{q}_A$ and $\mathbf{q}_B$ are formed by using a suitable number of floating DoF. In the case of Figure 4.4, which represents a strong discontinuity, $\mathbf{q}_A^T = [\mathbf{q}_8^T, \mathbf{q}_7^T, \mathbf{q}_3^T, \mathbf{q}_4^T]$ and $\mathbf{q}_B^T = [\mathbf{q}_1^T, \mathbf{q}_2^T, \mathbf{q}_6^T, \mathbf{q}_5^T]$ which employs four sets of floating DoF: $\mathbf{q}_5$, $\mathbf{q}_6$, $\mathbf{q}_7$ and $\mathbf{q}_8$. Note that no initial coordinates need to be associated with these floating DoF; they are not needed in the analysis. If, for instance, a weak discontinuity were to be modelled, only two sets of floating DoF would be included in the element, i.e., $\mathbf{q}_5$ and $\mathbf{q}_6$ would coincide with $\mathbf{q}_7$ and $\mathbf{q}_8$ (Figure 4.5a).

The strains then become (c.f. Equations 4.10 and 4.11):

$$\boldsymbol{\epsilon}_A = \mathcal{L}_{\mathbf{x}}(\mathbf{u}_A) = \mathcal{L}_{\xi}(\mathbf{N}) \mathbf{J}_A^{-1} \mathbf{q}_A = \mathbf{B}_A \mathbf{q}_A, \quad \mathbf{B}_A = \mathcal{L}_{\xi}(\mathbf{N}) \mathbf{J}_A^{-1}, \quad (4.30)$$

$$\boldsymbol{\epsilon}_B = \mathcal{L}_{\mathbf{x}}(\mathbf{u}_B) = \mathcal{L}_{\xi}(\mathbf{N}) \mathbf{J}_B^{-1} \mathbf{q}_B = \mathbf{B}_B \mathbf{q}_B, \quad \mathbf{B}_B = \mathcal{L}_{\xi}(\mathbf{N}) \mathbf{J}_B^{-1}. \quad (4.31)$$

The stiffness matrices for sub-elements Ω_A and Ω_B are (c.f. Equation 4.12):

$$\mathbf{K}_A = \int_{\Xi} \mathbf{B}_A^T \mathbf{D} \mathbf{B}_A \det(\mathbf{J}_A) d\Xi \quad \text{and} \quad \mathbf{K}_B = \int_{\Xi} \mathbf{B}_B^T \mathbf{D} \mathbf{B}_B \det(\mathbf{J}_B) d\Xi. \quad (4.32)$$

Note that the integrands for sub-elements Ω_A and Ω_B in Equation 4.32 are different; full integration on the natural domain Ξ is performed for both sub-elements and standard Gauss integration procedures can be employed. Similarly, the force vectors are (c.f. Equations 4.13 and 4.14):

$$\begin{aligned} \mathbf{Q}_A &= \int_{\Xi} \mathbf{N}^T \mathbf{f} \det(\mathbf{J}_A) d\Xi + \int_{\Gamma_{\Xi}} \mathbf{N}^T \mathbf{t} \det(\mathbf{J}_A) d\Gamma_{\Xi} \quad \text{and} \\ \mathbf{Q}_B &= \int_{\Xi} \mathbf{N}^T \mathbf{f} \det(\mathbf{J}_B) d\Xi + \int_{\Gamma_{\Xi}} \mathbf{N}^T \mathbf{t} \det(\mathbf{J}_B) d\Gamma_{\Xi}. \end{aligned} \quad (4.33)$$

From Equation 4.1, the equations of equilibrium for the two sub-elements are:

$$\mathbf{K}_A \mathbf{q}_A = \mathbf{Q}_A \quad \text{and} \quad \mathbf{K}_B \mathbf{q}_B = \mathbf{Q}_B. \quad (4.34)$$

Finally, in the same way as the PNM, the equation of equilibrium of the Floating Node (FN) element is the assembly of those of the two sub-elements:

$$\mathbf{K} \mathbf{q} = \mathbf{Q}. \quad (4.35)$$

where

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_A & \\ & \mathbf{K}_B \end{bmatrix}, \quad (4.36)$$

and

$$\mathbf{q}^T = [\mathbf{q}_A^T, \mathbf{q}_B^T], \quad (4.37)$$

and

$$\mathbf{Q}^T = [\mathbf{Q}_A^T, \mathbf{Q}_B^T]. \quad (4.38)$$

Comparing Figure 4.4 with Figure 4.3, it is easy to verify that unlike in the PNM, the system equations of a sub-element in the FNM are the same as those of a standard finite element defined on the same do-

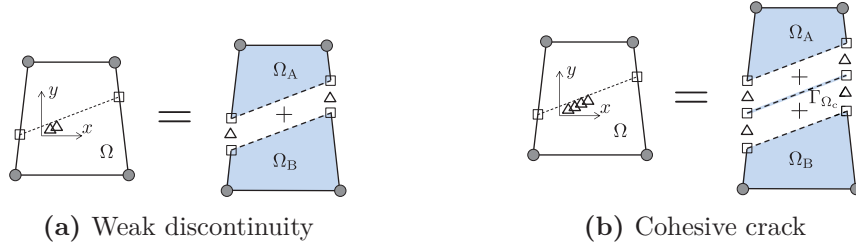


Figure 4.5: Weak discontinuity and cohesive crack

main. Therefore, in terms of the solution, the FN element after splitting (Figure 4.4) is equivalent to a FEM mesh representing the same cracking scenario (Figure 4.3), and thus the error in the representation of the discontinuity geometry associated with the PNM (Section 4.1.5) is mitigated in this case.

4.2.4 With weak discontinuities and cohesive cracks

The representation of weak discontinuities is analogous to the representation of strong discontinuities, requiring only fewer floating DoF, the latter being shared (through the assembly process) between both sub-elements (Figure 4.5a). This is simpler to implement and involves less DoF, than in the PNM where the modelling of a weak discontinuity requires constraints imposed on generally all the DoF (Section 4.1.4).

Cohesive cracks (Figure 4.5b) can be represented by partitioning the integration domain Ω into two quadrilaterals (Ω_A and Ω_B) and a surface (Γ_{Ω_c}). Consider an element as in Figure 4.5b with integration domain Ω partitioned into two sub-elements with integration domains Ω_A and Ω_B , as well as a cohesive sub-element with integration domain Γ_{Ω_c} . The left hand side of the equilibrium equation (Equation 4.1) can be expanded from the integration domain Ω to the three integration domains Ω_A , Ω_B and Γ_{Ω_c} :

$$\int_{\Omega} \boldsymbol{\epsilon}^T(\mathbf{v}) \boldsymbol{\sigma}(\mathbf{u}) \, d\Omega = \int_{\Omega_A} \boldsymbol{\epsilon}^T(\mathbf{v}) \boldsymbol{\sigma}(\mathbf{u}) \, d\Omega + \int_{\Omega_B} \boldsymbol{\epsilon}^T(\mathbf{v}) \boldsymbol{\sigma}(\mathbf{u}) \, d\Omega + \int_{\Gamma_{\Omega_c}} \llbracket \mathbf{v} \rrbracket^T \boldsymbol{\tau}_c(\llbracket \mathbf{u} \rrbracket) \, d\Gamma_{\Omega_c}. \quad (4.39)$$

where $\llbracket \bullet \rrbracket$ represents the jump of a function between the top and bottom surfaces of the cohesive element, and $\boldsymbol{\tau}_c$ is the traction vector acting on the surfaces of the cohesive sub-element. In this section, we are interested in using the last term on the right hand side of Equation 4.39 to define the

stiffness matrix for the cohesive sub-element.

The cohesive element topology can be defined through the vector of nodal coordinates $\mathbf{x}_{\Gamma_{\Omega_c}}$ which defines the discontinuity; in the case of Figure 4.5b, and using the notation in Figure 4.4, $\mathbf{x}_{\Gamma_{\Omega_c}}^T = [\mathbf{x}_r^T, \mathbf{x}_s^T]$. The physical coordinates of a point can be defined from the natural coordinates using suitable shape functions, as $\mathbf{x} = \mathbf{N}_{CE} \mathbf{x}_{\Gamma_{\Omega_c}}$, leading to the Jacobian of the transformation:

$$\mathbf{J}_{CE} = \frac{d\mathbf{x}}{d\xi} = \frac{d\mathbf{N}_{CE}}{d\xi} \mathbf{x}_{\Gamma_{\Omega_c}}. \quad (4.40)$$

The displacement jump $[[\mathbf{u}]]$ (and $[[\mathbf{v}]]$) can be obtained from the DoF as:

$$[[\mathbf{u}]] = [\mathbf{N}_{CE} \quad -\mathbf{N}_{CE}] \mathbf{q}_{CE}. \quad (4.41)$$

In the case of Figure 4.5b, and using the notation in Figure 4.4, $\mathbf{q}_{CE}^T = [\mathbf{q}_5^T, \mathbf{q}_6^T, \mathbf{q}_7^T, \mathbf{q}_8^T]$, i.e., the DoF vector of the cohesive sub-element is entirely formed by floating DoF. The traction vector $\boldsymbol{\tau}_c$ is typically related to the displacement jump through a constitutive relation of the form:

$$\boldsymbol{\tau}_c = \mathbf{D}_{CE} [[\mathbf{u}]] = \mathbf{D}_{CE} [\mathbf{N}_{CE} \quad -\mathbf{N}_{CE}] \mathbf{q}_{CE}. \quad (4.42)$$

where $\mathbf{D}_{CE}$ embodies the cohesive constitutive law. Replacing Equations 4.41 and 4.42 in the last term on the right hand side of Equation 4.39 leads to the definition of the stiffness matrix for the cohesive sub-element:

$$\mathbf{K}_{CE} = \int_{\Gamma_{\Xi_c}} [\mathbf{N}_{CE} \quad -\mathbf{N}_{CE}]^T \mathbf{D}_{CE} [\mathbf{N}_{CE} \quad -\mathbf{N}_{CE}] \mathbf{J}_{CE} d\Gamma_{\Xi_c}. \quad (4.43)$$

Therefore, with the FNM, the floating DoF sets can directly interpolate the displacement jump across the cohesive interface. The stiffness matrix of the cohesive sub-element is locally assembled to the stiffness matrix of the FN element, together with those of Ω_A and Ω_B . It is easy to verify that the formulation of the cohesive sub-element is the same as that of a standard cohesive element. In the case of a 2D linear cohesive sub-element, the cohesive traction directly affects four DoF sets, as opposed to in the PNM where a double of that is directly involved (Section 4.1.4). Therefore, the resulted stiffness matrix of the FN element after partitioning (which

is the same as that of a FEM mesh of two quadrilateral elements and one cohesive elements representing the same cracking scenario) has less coupling terms than that of the PNM (Section 4.1.4).

It should be noted that the partitioning of the FN element only forms the cohesive sub-element along the line of propagation; the eventual fracture of the cohesive surface is determined by the cohesive law associated with the cohesive sub-element. Therefore, two criteria, i.e., one to initiate the partitioning of the FN element and one to determine the failure of the cohesive sub-element, are needed for the eventual propagation of a cohesive crack within a FN element (see details in Section 4.2.9).

4.2.5 Different geometries for the discontinuities and integration

Figure 4.4 shows one particular case for the intersection between a quadrilateral element and a strong discontinuity. This case leads naturally to the partition of the domain Ω into two quadrilateral sub-domains Ω_A and Ω_B ; for integration, each sub-domain then maps trivially into a quadrilateral element in the natural space Ξ (see Figure 4.4). Another possibility is shown in Figure 4.6a, where a linear triangular sub-element is formed over the partition Ω_A with nodal coordinates defined by the crack tip and two real nodes, and the crack tip DoF represented by a floating DoF set (Figure 4.7). Following the procedures presented in Section 4.2.3, it can be shown that the resulting equilibrium equation of the triangular sub-element is the same as that of a standard linear triangular element (Figure 4.7); thus, standard Gauss integration method can be employed. Finally, other possibilities for the intersection may lead to the partition of Ω into two triangular sub-elements (Figure 4.6b) or one triangular sub-element and one pentagon; the pentagon is then partitioned into three triangular sub-elements for integration (Figure 4.6c). Depending on the actual intersection between the discontinuity and the element, accurate integration may require a more complex integration scheme, such as the one shown in Figure 4.6d, where the pentagon is integrated by decomposing it into five triangular sub-elements, using for instance Delaunay triangulation. The pentagons in Figure 4.6c and Figure 4.6d are equivalent to a FEM mesh of

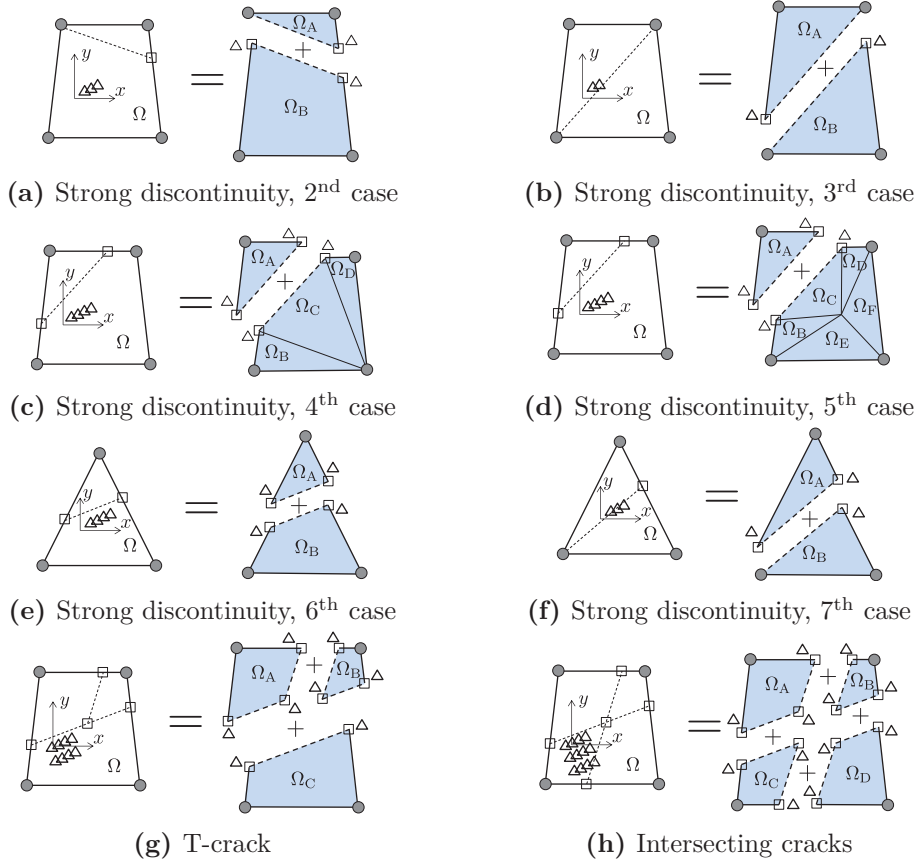


Figure 4.6: Examples of different geometries of elements, partitions and discontinuities that can be modelled by the FNM(see key in Figure 4.4).

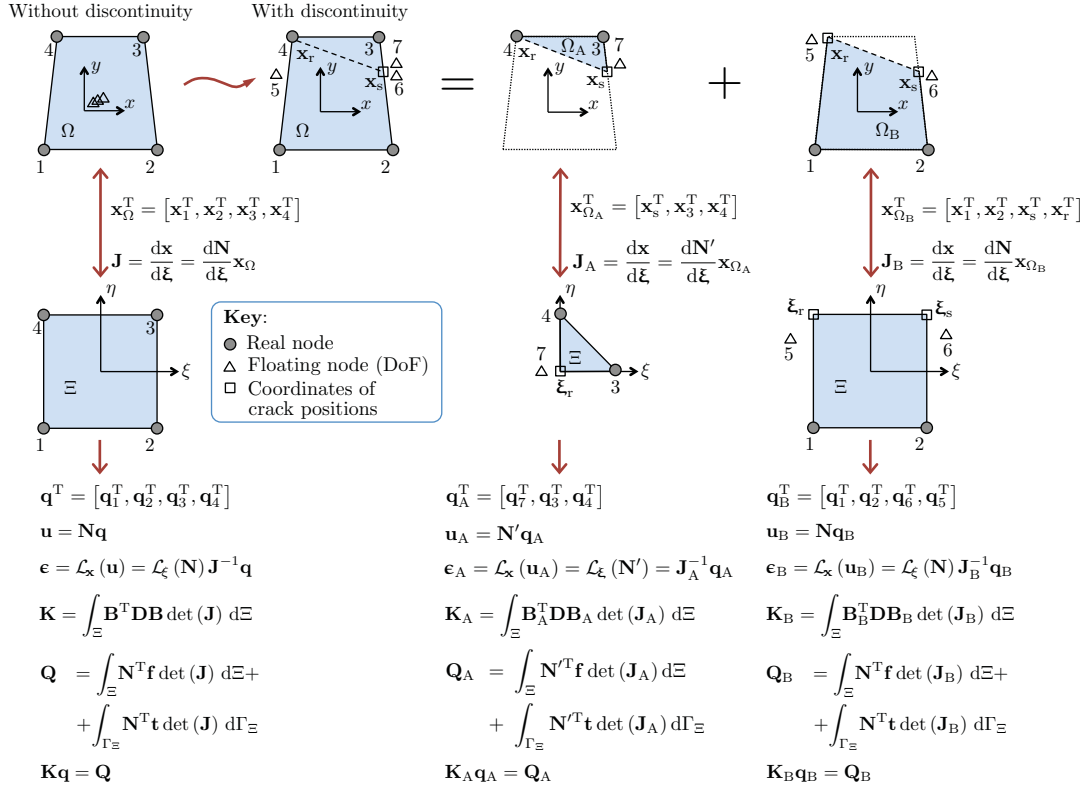


Figure 4.7: FNM with one triangular partition and one quadrilateral partition; $\mathbf{N}'$ is the standard linear triangular element shape function matrix.

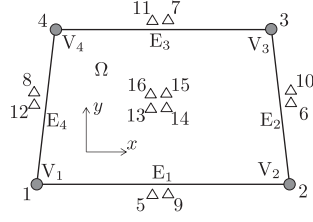
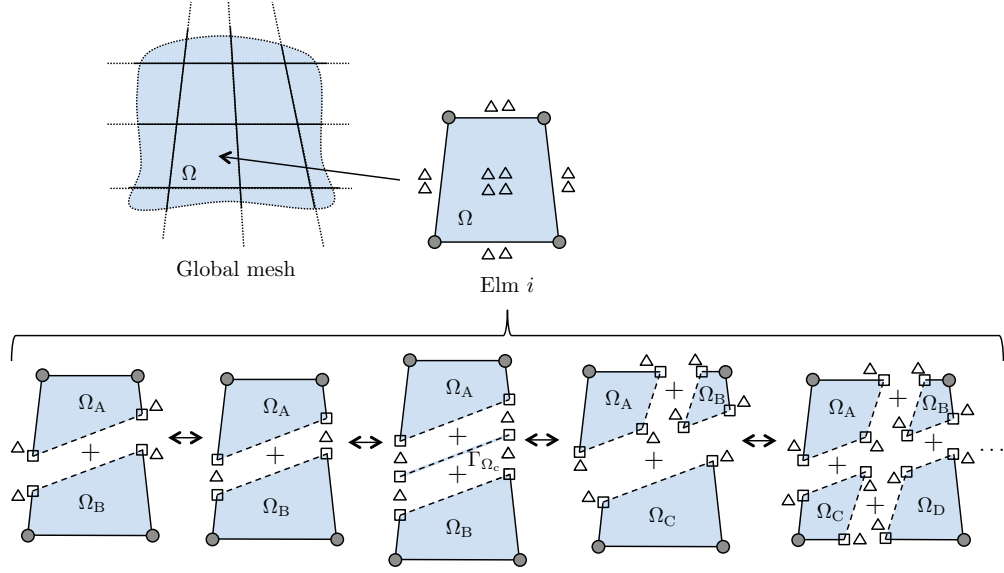


Figure 4.8: Example of local DoF, vertices and edges numbering for the FN element in Figure 4.6h. To facilitate assembly, a pair of floating DoF are associated with each edge, and four floating DoF are internal.

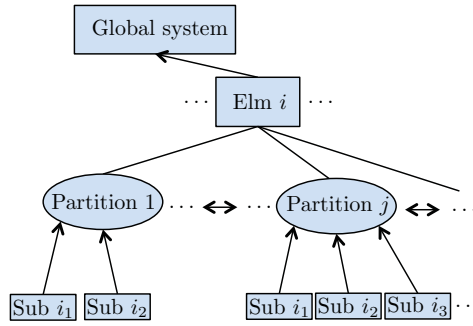
three and five linear triangular elements respectively, and standard Gauss integration procedures can be employed for their stiffness integration (note that the DoF set associated with the centre vertex of triangulation in Figure 4.6d can be calculated from those of the other vertices through static condensation and therefore is actually not needed).

The method is not limited to quadrilateral elements. A crack propagating through a triangular element may cut the domain into one quadrilateral and one triangle (Figure 4.6e) or into two triangles (Figure 4.6f); both cases can be modelled by a FNM triangular element with a suitable number of floating DoF. Before failure initiation, the FNM triangular element would behave like a standard linear triangular element and the floating nodes are not used. When a discontinuity is to be modelled within its domain, the triangular element is partitioned as shown in Figure 4.6e or Figure 4.6f; a suitable number of floating DoF are used to form the displacement vector of each sub-element, and the calculations of each sub-element follow from the standard procedures presented Section 4.2.3, Figure 4.4 or Figure 4.7.

More complex scenarios for the topology of the discontinuity can be represented, requiring only that a suitable number of floating DoF are included in the element formulation. Figure 4.6g shows how a T-shaped crack leads to partitioning the integration domain Ω into three quadrilaterals; Figure 4.6h shows that two intersecting cracks lead to the partition of Ω into four quadrilaterals.



(a) The FN element in Figure 4.8 can represent a wide range of cracking scenarios locally within its formulation.



(b) Schematic drawing of the calculation and assembly procedures.

Figure 4.9: A FN element can represent a wide range of cracking scenarios by adopting different partitioning strategies to form sub-elements for integration; the change from one scenario to another can be achieved by switching between the different partitioning strategies within the local element calculation.

4.2.6 Element topology and assembly

Consider, as a relatively generic example, the FN element shown in Figure 4.8 suitable to modelling two intersecting strong discontinuities. The vertices, edges, real DoF and floating DoF are clearly labelled with their local index numbers. The full definition of a completely generic FN element with global index i_{elem} can be represented as:

$$i_{\text{elem}}, \quad \text{Node}(1 : N_{\text{node}}), \quad \text{Edge}(1 : N_{\text{edge}}), \quad \text{rDOF}(1 : N_{\text{rDOF}}), \quad (4.44) \\ \text{fDOF}_j(1 : N_{\text{fDOF}_j}) \text{ with } j = 1 \text{ to } N_{\text{edge}}, \quad \text{fDOF}_D(1 : N_{\text{fDOF}_D})$$

where the arrays Node, Edge, rDOF, fDOF_j and rDOF_D contain respectively the global indices for the N_{node} nodes, for the N_{edge} edges, for the N_{rDOF} real DoF, for the N_{fDOF_j} floating DoF on edge j and for the N_{fDOF_D} DoF inside the original element domain (i.e. those not at the edges), all for element i_{elem} .

Some floating DoF are topologically associated to the edges of the element (to represent the DoF of crack tips) which are generally shared by neighbouring elements. As such, floating DoF must be duly assembled. To exemplify the assembly process, we consider again the FN element shown in Figure 4.8. During integration, the domain of this element is partitioned as detailed previously in Figure 4.6h (see also Section 4.2.5). Each of the floating DoF is associated either to an edge (DoF sets 9 to 12) or to the inner domain of the element (DoF sets 13 to 16). Shared floating DoF associated with common edges of neighbouring elements are assembled to the same positions in the global DoF vector according to their global DoF indices, in the same way as the assembly of the shared real nodes.

A wide range of possible cracking scenarios can be modelled by such a FN element (Figure 4.9b), and the solutions are the same as those of the FEM meshes which represent the cracking scenarios explicitly. The change from one scenario to another, e.g., from a single discontinuity (Figure 4.4) to a T-crack network (Figure 4.6g) and then further to an intersecting network (Figure 4.6h), requires only the change of the domain partitioning strategy for stiffness integration and activating the right sets of floating DoF to form the nodal DoF vectors for the new sub-elements; all these can be directly implemented within the local FN element subroutine; no global

update of geometry and subsequent remeshing are needed; the original mesh and element connectivities are preserved.

4.2.7 Comparison with other methods

Comparison with PNM

Similarly to the PNM, the modelling of a crack by the FNM is essentially realised by partitioning the original element into sub-elements to represent the cracking scenario; all DoF of the sub-elements are only displacement DoF and the interpolation only uses standard shape functions, making the implementation of the FNM in existing FEM packages straightforward. However, the differences between the PNM and the FNM are that:

- the FNM makes direct use of the available crack boundary coordinates for the definition and transformation of the sub-elements (Section 4.2.3), for which the PNM uses the coordinates of the phantom nodes instead (Section 4.1.3); the coordinates of the floating nodes in the FNM are not used in the calculation and effectively they can be anywhere (Section 4.2.3), while those of the phantom nodes need to coincide with the coordinates of the real nodes in the PNM;
- in the FNM, the sub-elements are transformed onto the natural space for integration through different Jacobians, and the integrands for the calculation of system matrices of the two sub-elements are different (Section 4.2.3); while in the PNM, the Jacobians and the integrands of the two sub-elements are the same (Section 4.1.3); the integration in the FNM is performed on the full domain where the standard Gauss integration can be employed (Figure 4.4); the integration over the partial domains Ξ_A and Ξ_B in the PNM (Figure 4.2) and XFEM [51] is not required in the FNM;
- unlike the PNM and XFEM (see Section 4.1.5), the FNM leads to equations of equilibrium (Equation 4.34, see also Figure 4.4) which coincide exactly with those that would correspond to an equivalent FE mesh representing the same discontinuity explicitly (Figure 4.3), and thus the respective numerical solutions will always coincide; the

error in the representation of discontinuities in the natural coordinates associated with the PNM and XFEM (see Section 4.1.5) is mitigated in the FNM (Figure 4.4);

- the FNM is naturally well suited for representing weak discontinuities and cohesive cracks (Section 4.2.4); unlike the PNM, weak discontinuities in the FNM lead directly to fewer DoF than strong discontinuities; cohesive cracks can be readily represented exactly with cohesive sub-elements, as if cohesive elements were used directly, thus simplifying considerably its calculation and assembly procedures as compared to the case with the PNM (Section 4.2.4);
- while modelling more types of cracking scenarios in the PNM is challenging, this can be achieved with the FNM by simply adopting the appropriate partitioning strategies and activating the correct sets of floating DoF topologically associated with the FN element (Section 4.2.6);
- by avoiding the initial choice of requiring that the extra nodes be coincident with the real nodes (as is the case in the PNM), the FNM is conceptually simpler to understand, as the floating DoF have a direct meaning: the displacement at a defined material point on the surface of the discontinuity.

Comparison with remeshing

In terms of the final solution, the FNM effectively implements a local remeshing within the cracked element. However, the FNM differs from remeshing [4, 8, 9, 26, 74, 75, 77, 86, 93–95, 101, 122–124, 134] in terms of the methodology. Remeshing models discontinuities by essentially re-applying the FEM over a new geometry description according to a new mesh requirement; element formulation and assembly architecture remain unchanged. Similarly as in the PNM, the FNM models discontinuities within a user-defined element through modifying the original element's formulation into an assembly of sub-elements, and the sub-element system equations are assembled locally within the element. In this way, the

global quantities (geometry, mesh and element connectivities) remain unchanged in the FNM as opposed to the general case in remeshing. Extra databases used in remeshing for the storage, update and transfer (from the old mesh to the new one) of solution dependent variables are not needed in the FNM; the variables are stored and updated locally within each FN element. In remeshing, the elements (either new or old) are generally independent from each other in their calculation, and a dedicated algorithm would be required to manage and exchange information between them; while in the FNM, the local sub-elements are within a single element and all their information is shared (which facilitates the modelling of multiple intersecting cohesive cracks, see Section 4.4). Remeshing with an appropriate error estimator usually leads to an optimal mesh with minimal element distortion [9, 26, 93, 101, 122]; however, in the FNM (or XFEM and PNM), the crack may partition the element domain into distorted shapes or very small sub-domains which lead to difficulties in integration and calculation [51].

Comparison with XFEM

The FNM is not equivalent to XFEM as opposed to the PNM (see Section 4.1.5). This is due to the fact that the shapes of the two FNM sub-elements in physical space are different from each other (Figure 4.4), and their corresponding interpolation functions in physical coordinates (which are mapped from the same shape functions in natural coordinates through different transformations) are different from each other, i.e., $\mathbf{u}_A(\mathbf{x}) = \Phi_A(\mathbf{x}) \mathbf{q}_A$ and $\mathbf{u}_B(\mathbf{x}) = \Phi_B(\mathbf{x}) \mathbf{q}_B$ and Equation 4.25 cannot be obtained (note that the same arguments hold for the comparison between XFEM and FEM/remeshing (Figure 4.3)).

4.2.8 Formulation of FN elements for composite laminates

In composite laminates, matrix cracking often triggers the subsequent delamination from the matrix crack tips [137, 138]. Existing computational methods for composites usually model each ply and each interface separately, and the damage/failure status of a ply element is generally not

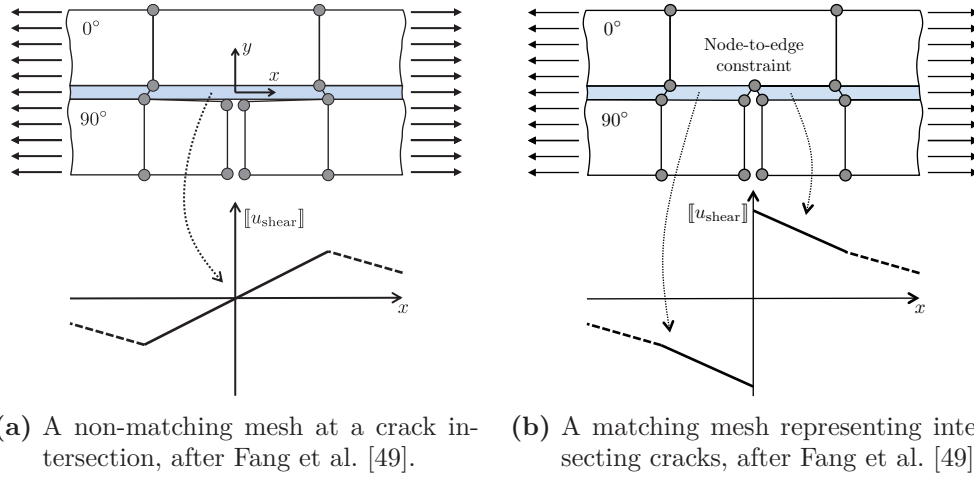


Figure 4.10: Modelling the intersection between matrix cracks and delamination. Non-matching mesh fails to capture the displacement jump across the crack tip (a); matching mesh captures the displacement jump across the crack tip (b).

available to its adjacent interface element. When a matrix crack occurs in the ply element (e.g., using the PNM), the cohesive element at the interface does not reflect this discontinuity, leading to a non-matching mesh at the matrix crack tip (Figure 4.10a). Fang et al. [49] showed that using non-matching meshes at a crack intersection leads to an inaccurate representation of the displacement jump (and hence of the cohesive traction) at the interface (Figure 4.10a); capturing correctly the displacement jump requires further DoF at the intersection between cracks (Figure 4.10b). They then developed a breakable cohesive element using the Augmented FEM (an approach very similar to the PNM), such that the cohesive element breaks into two sub-elements when the adjacent ply elements break; such an approach requires passing damage variables among different elements across the thickness of the laminate which is non-trivial to implement within existing FEM packages.

With the capability of modelling complex crack networks shown in Section 4.2.5, the FNM can be applied to formulate specific elements for composite laminates to model accurately the network of matrix crack and delamination in an integrated manner. Without loss of generality, the formulation of a FN element for cross-ply laminates is shown as an example. The cross-ply FN element is formed with both real nodes and floating nodes,

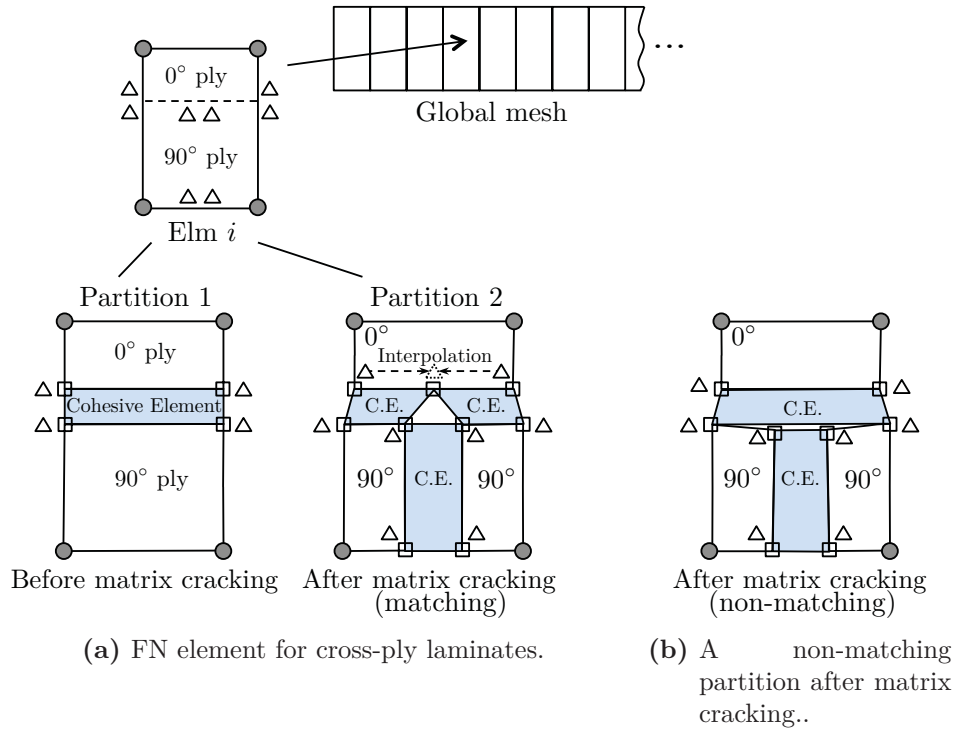


Figure 4.11: The FN element formulated for cross-ply composites with a matching partition at the interface after matrix cracking (a); and the non-matching partition after matrix cracking (b).

as shown in Figure 4.11a. The entire laminate across the thickness is modelled within such an element. A pair of floating nodes is topologically associated with the left, right and bottom edge, as well as the internal 0/90 interface of the FN element. The FN element makes use of the known position of the interface, so that the interface is not seeded with real nodes; instead, it is represented by a cohesive sub-element formed with floating nodes (Figure 4.11a 'Partition 1'). In this way, minimum seeding is required during preprocessing. The advantage of modelling all the plies and interfaces within a local element is that all the information (stress, strain, failure, etc.) across the thickness of the laminate is shared, thus allowing the straightforward representation of any interacting scenario across the thickness. For example, a matching representation at the intersection of the matrix crack and the interface (Figure 4.10b) can be implemented within the FN element by having the cohesive sub-element at the interface break into two sub-elements as soon as a matrix crack is judged to appear in the 90-ply sub-element (Figure 4.11a 'Partition 2'). The FN element in Figure 4.11a behaves in such a way that:

- (a) initially, the element is partitioned into three sub-elements as shown in Figure 4.11a 'Partition 1'; each ply is represented by a quadrilateral sub-element and the interface by a cohesive sub-element;
- (b) upon satisfaction of the failure criterion in the 90° ply (Equation 4.48), the element switches to the partition as shown in Figure 4.11a 'Partition 2'; the 90° ply is split into two quadrilateral sub-elements and one cohesive sub-element, and the interface is split into two cohesive sub-elements, such that the intersection of the matrix crack and the interface can be accurately represented;
- (c) if energetically favourable, the cohesive sub-elements then propagate the matrix crack and eventually also the delamination, according to their individual failure propagation theories.

A non-matching partition (Figure 4.11b) uses only one cohesive sub-element to model the delamination along the interface, rather than two. A FN element with this partition corresponds very closely to a model in which the matrix crack in the 90° ply is modelled with the PNM or XFEM,

and the delamination is modelled independently using a cohesive element. Such a non-matching model will be shown to have erroneous prediction of the delamination propagation and the matrix crack density saturation (see Section 4.4).

4.2.9 Other details

Closing the crack tip

Considering Figure 4.8, suppose that the strong discontinuity terminates at edge E_2 , i.e., the element to the right of E_2 has not failed. The existence of a crack tip at edge E_2 implies that $\mathbf{q}_6 = \mathbf{q}_{10}$. There are several ways in which this can be imposed, as discussed below.

Internal assembly of the floating DoF When a suitable criterion predicts that a discontinuity should be created inside an element, the discontinuity will intersect two edges of the element. The equations of equilibrium for the partitions of the FN element (Equation 4.34) can be assembled (to form the equations of equilibrium for the FN element) such that the two sets of floating DoF at the crack tip are assembled into one single set; in other words, only one set of floating DoF is actually used. For instance, if the FN element in Figure 4.8 partitions into Ω_A and Ω_B as shown in Figure 4.2 and the crack tip lies at edge E_2 , then during assembly of the stiffness matrix for the FN element, terms corresponding to floating DoF set $\mathbf{q}_6$ (of domain Ω_B) and $\mathbf{q}_{10}$ (of domain Ω_A) are assembled into the same set of slots in the element's stiffness matrix (e.g., that of $\mathbf{q}_6$), and only one of the two DoF sets (in this case, $\mathbf{q}_6$) is actually used.

Constraint equations The equilibrium of an element, prior to the existence of a discontinuity, is given (for the real DoF) in Equation 4.8. Constraint equations can be employed to ensure that the two sets of floating DoF on a non-failed edge coincide (and hence that the crack tip is closed). For instance, considering the ordering of DoF in Figure 4.8, this gives:

$$k \left(\mathbf{q}_i - \mathbf{q}_{i+N_{\text{edge}}} \right) = 0, \quad (4.45)$$

where k is a penalty stiffness, N_{edge} is the number of edges (in the case of Figure 4.8, $N_{\text{edge}} = 4$) and $i = 5, \dots, 8$. Therefore, the equilibrium of non-failed elements is represented by Equation 4.8 (for the real DoF) and Equation 4.45 (for the floating DoF). Together with the assembly of adjacent elements, this leads naturally to the representation of crack tips.

Improving the representation of the crack tip

Considering Figures 4.12a and 4.12b, it is clear that closing the crack tip is not sufficient for an accurate representation of the crack tip; in fact; the transition from a strong discontinuity to no discontinuity can lead to an artifact whereby the element at the crack tip (see Figures 4.12a and 4.12b) does not have the adequate topology for connecting to the adjacent uncracked element (lack of compatibility). Figures 4.12c and 4.12d show that this can be mitigated to an extent using constraint equations which define the DoF at the crack tip through interpolation of suitable DoF of the element at the crack tip.

A better solution is shown in Figures 4.12e and 4.12f; the lack of compatibility can be avoided by using a transition element at the crack tip (in fact, the code required for doing Figures 4.12e is exactly the same used to represent weak discontinuities within the FNM). The partition line can run from the crack tip to one of the opposing vertices of the transition element as shown in Figures 4.12e or to a point within the domain (whose DoF can be removed from static condensation) as shown in Figures 4.12f, regardless of the exact orientation of the crack. While Figures 4.12e and 4.12f show two possibilities for element partitioning, other possibilities could be considered, such as joining the crack tip with the two opposing vertices to divide the transition element domain into three triangular elements.

Figure 4.12g shows that refinement elements can be used as well, leading to further improvements in the representation of the crack tip. If at least one refinement element is used (Figure 4.12g), then the virtual crack closure technique (VCCT) can be used directly to model propagation of strong discontinuities, which will be detailed in section 4.2.9.

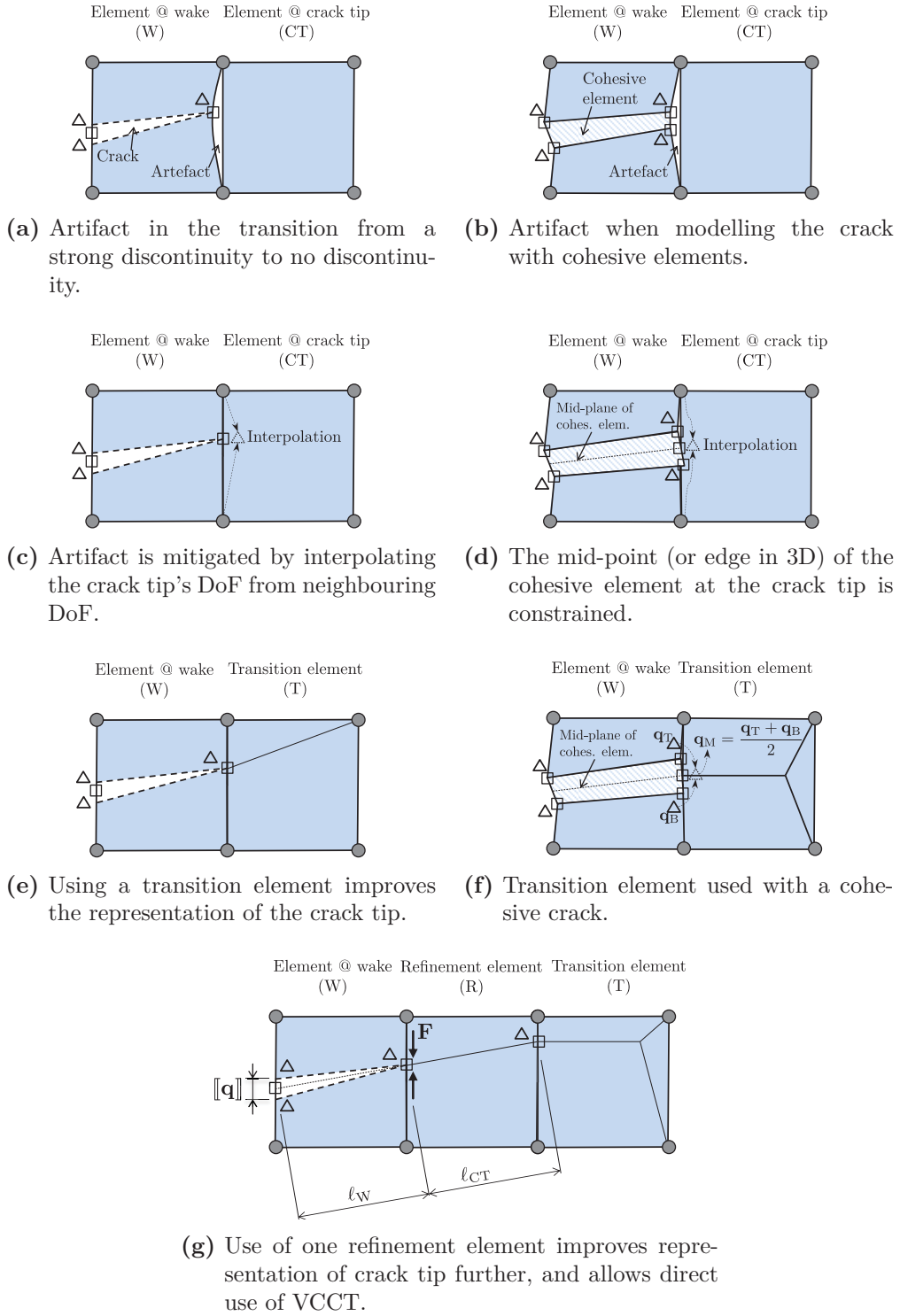


Figure 4.12: Using local refinement elements and transition elements to represent the crack tip more accurately (see key in Figure 4.4).

Representation of discontinuities

Representing a discontinuity across an entire mesh implies passing information which defines this discontinuity to each element. With this information, the element will follow the appropriate integration procedure (see Figure 4.6). The level set method is for instance well suited for this purpose [22]. We present below an alternative to the level set method which offers some advantages for the representation of complex crack networks.

We firstly define an edge status variable μ for each edge with global index j :

$$\mu(j) = \begin{cases} -1, & \text{if there is no discontinuity at edge } E_j; \\ 0, & \text{if } E_j \text{ corresponds to the tip of a discontinuity;} \\ 1, & \text{if a discontinuity crosses through edge } E_j. \end{cases} \quad (4.46)$$

and then define a dataset for each edge j :

$$\mu(j), \quad \xi(j) \quad (4.47)$$

where $\xi(j)$ contains the natural coordinates of the intersection of a discontinuity with edge j , if such intersection exists. The dataset is then made available to all elements that include edge j . The initial status of a discontinuity (prior to an analysis) can be defined through an array of initial values for the dataset in Equation 4.47. The following explains the application of the edge status variable on the modelling of a crack propagating through an element (with topology shown in Figure 4.8) in a mesh:

- if no discontinuity is present in the element or on its edges, $\mu(j) = -1$ for all the edges and $\xi(j)$ is empty;
- if a crack propagates until an edge (e.g. E_4 in Figure 4.8) of the element but not going through (μ of E_4 changes from -1 to 0), the element would partition its domain and change its stiffness integration to become a transition element as in Figure 4.12e or 4.12f so as to improve the representation of the field variables ahead of a crack tip (Section 4.2.9);
- if the crack is judged to propagate in the element (by a propagation

criterion, see Section 4.2.9) and terminate on one of its edge (e.g. E_2 in Figure 4.8), the element would partition its domain and update its stiffness matrix to represent this crack and close the tip on its edge as shown in Figure 4.12c or 4.12d (Section 4.2.9), and modify the datasets associated with its edges according to Equation 4.46 ($\mu = 1$ on E_4 and $\mu = 0$ on E_2); the neighbouring element ahead of the tip (which shares edge E_2) would then change to a transition element upon accessing its edge status variables;

- if μ is changed from 0 to 1 on the edge with the tip (E_2), indicating that the neighbouring transition element has propagated the crack, the element would then repartition its domain and update its stiffness matrix to represent the crack through the element as shown in Figure 4.4 or 4.5b, depending on its type;
- if another crack were to go through the element from another edge (e.g. E_3), the element may be repartitioned to represent a T-crack as shown in Figure 4.6g; the discontinuity between Ω_A and Ω_B in Figure 4.6g could be weak (if $\mu = 0$ for E_3) to improve the representation of the tip, or strong/cohesive (if $\mu = 1$ for E_3) to propagate the crack in the element and form the T-crack network.

Therefore, elements only need to check and update their associated edge status variables, and do the necessary partitioning/repartitioning accordingly, to represent the propagation of discontinuities within the mesh. Note that with this approach, each element only accesses a fixed amount of information (the status variables of its edges); the tracking of all the cracks by every element in the mesh is not needed.

Propagation of discontinuities

Discontinuities can propagate according to different laws or criteria, depending on the type of discontinuity. In principle, any method suitable for FE implementation can be used with the FNM. When the discontinuities represent cracks, there are two approaches of particular interest; these are detailed below.

Stress based criteria with cohesive formulations When a crack in a FN element is represented through a cohesive model, with the latter having a stress-based failure initiation criterion embedded in its formulation, it is natural to also use an identical stress-based failure criterion to decide on the partitioning of the FN element to form a cohesive sub-element along the cohesive surface (Section 4.2.4). If the failure criterion for the cohesive sub-element is:

$$f(\boldsymbol{\sigma}) = 0, \quad (4.48)$$

then the partitioning criterion of the FN element can be:

$$f(\boldsymbol{\sigma}) = \epsilon, \quad \text{with } -1 \leq \epsilon \leq 0, \quad (4.49)$$

where ϵ is a user-defined non-positive number such that the cohesive sub-element is activated before the actual propagation is due to occur. A more negative value of ϵ implies an earlier allocation of the cohesive sub-element. In the limit, if ϵ equals -1 , then the crack tips are pre-determined and cohesive sub-elements are effectively pre-allocated before analysis.

For the initiation of a crack, Equation 4.49 can be applied to the stresses at the integration points of any non-failed element. For the propagation of a crack through a transition element as shown in Figure 4.12f, Equation 4.49 can be applied to both the stresses at the integration points of the transition element and the cohesive traction at the crack tip (calculated using the displacement jump between $\mathbf{q}_T$ and $\mathbf{q}_B$ in Figure 4.12f). Once the partitioning criterion in Equation 4.49 is verified, the edge status variables are updated and the integration is carried out accordingly, as detailed in Sections 4.2.9, 4.2.9 and 4.2.9. The eventual propagation of the cohesive crack is then determined by the failure criterion of the cohesive sub-element (Equation 4.48).

Virtual crack closure technique It is possible to propagate cracks without using stress measures directly, i.e. using fracture mechanics concepts only. For this, VCCT [121] is particularly well suited for using with the FNM. An extensive review of VCCT is provided in [76]. Traditionally, VCCT is used to obtain energy release rates in cases where the crack path is known beforehand and can be aligned with the elements' edges, such as for

delamination. Combining VCCT with the FNM enables the energy release rates to be obtained for cracks having an arbitrary orientation relative to the mesh.

Consider Figure 4.12g, where the element at the crack tip (refinement element labelled R) has not failed yet, but contains a weak discontinuity. The weak discontinuity is introduced to better represent the crack tip. It is equivalent to a local mesh refinement, and can be extended to several elements ahead of the crack tip. Let $\mathbf{F}$ be the internal force vector at the crack tip and $[[\mathbf{q}]]$ be the displacement jump at the opposing edge in the wake element. Also, let A_W be the area of the crack surface in the element on the wake of the crack (for a 2 dimensional problem, $A_W = \ell_W b$, where ℓ_W is indicated in Figure 4.12g and b is the thickness) and A_{CT} be the area of the crack surface in the element at the crack tip (for a 2 dimensional problem, $A_{CT} = \ell_{CT} b$, where ℓ_{CT} is indicated in Figure 4.12g). Then, the energy release rates for modes I and II can be calculated as [76]:

$$G_I = \frac{1}{2A_W} F_n [[q_n]] \left(\frac{A_W}{A_{CT}} \right)^{1/2} \quad (4.50)$$

$$G_{II} = \frac{1}{2A_W} F_t [[q_t]] \left(\frac{A_W}{A_{CT}} \right)^{1/2} \quad (4.51)$$

where F_n and F_t are respectively the components of $\mathbf{F}$ in the normal and tangential directions to the crack, and $[[q_n]]$ and $[[q_t]]$ are respectively the components of $[[\mathbf{q}]]$ in the normal and tangential directions to the crack. Once the energy release rates are known, a criterion of the form:

$$f(G_I, G_{II}) = 0 \quad (4.52)$$

can be used to decide on the eventual propagation of a crack through the element and on the direction of the crack. Once the failure criterion is verified, the edge status variables are updated and the integration is carried out accordingly, as detailed in Sections 4.2.9, 4.2.9 and 4.2.9.

Calculating the energy release rates in Equations 4.50 and 4.51 requires that the element at a crack tip has access to the displacement jump at the opposing edge in the element in the wake of the crack (i.e., in Figure 4.12g, the refinement element (R) needs to know $[[\mathbf{q}]]$ at element (W).) This can

be achieved by expanding the dataset for each edge j (Equation 4.47) with more information to become:

$$\mu(j), \quad \xi(j), \quad \llbracket \mathbf{q} \rrbracket(\bar{j}), \quad \mathbf{n}(\bar{j}), \quad A(\bar{j}) \quad (4.53)$$

where $\llbracket \mathbf{q} \rrbracket(\bar{j})$ is the displacement jump at the other (opposing) edge of the element in the wake of the crack (W in Figure 4.12g), $\mathbf{n}(\bar{j})$ is the normal to the crack direction in element W, and $A(\bar{j})$ is the crack area in element W. The neighbouring element which shares edge j would then have access to all the data necessary for the implementation of VCCT.

Implementation in existing FE codes

In most existing FE codes, commercially available or otherwise, DoF are defined in an analysis as part of the definition of nodes. It is therefore convenient to define all DoF as being associated with a nodal position. The difficulty of course is that the floating DoF express displacements at positions which are not necessarily known at the beginning of the analysis. However, as in the PNM, the floating DoF in the FNM can be allocated to the element by defining nodes, in this case *floating* nodes, which belong to the element's connectivity. The actual coordinates of these floating nodes, defined when discretising the domain, are in fact not used by the method. For simplicity, the floating nodes can be all given nil physical coordinates. Using the FN element in Figure 4.8 as an example, the element can be defined to have 16 nodes (4 real nodes plus 12 floating nodes); the floating nodes (with nil coordinates) can be created and added to the FN element's connectivity during analysis if the main programme is accessible; otherwise, they can be created and allocated before analysis as part of the initial element connectivities.

Since, in most existing FE codes, the edges of an element are not attributed global indices, the dataset in Equation 4.47 or 4.53 (required for the representation of the crack; of the crack tip; and, in the case of Eq 4.53, for modelling propagation) can be defined in terms of floating nodes instead. Taking a strong discontinuity as an example, an edge will have two floating nodes on it; the dataset associated with the edge can therefore be associated with one of the floating DoF (or both) on that edge. The edge

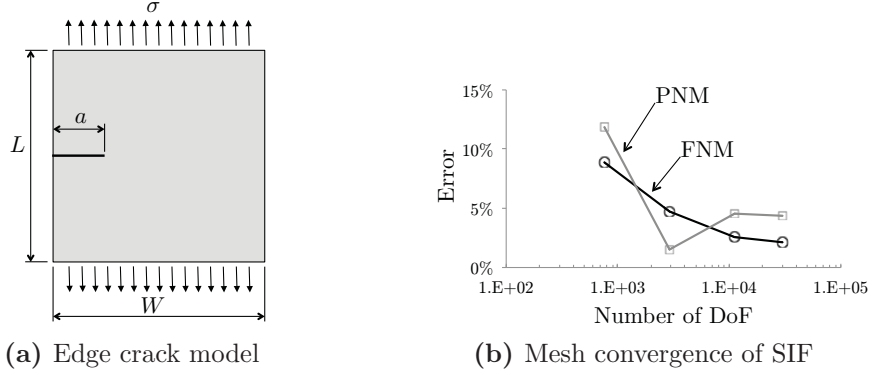


Figure 4.13: For this edge crack model, the FNM converges monotonically, unlike the PNM.

status variable becomes therefore a floating node status variable which can be stored in a global array indexed by the global floating node index i_{node} . If propagation is modelled using cohesive elements, each entry of the global array will be:

$$i_{\text{node}}, \quad \mu(i_{\text{node}}), \quad \xi(i_{\text{node}}), \quad (4.54)$$

while, if propagation is modelled with VCCT, each entry of the global array will be:

$$i_{\text{node}}, \quad \mu(i_{\text{node}}), \quad \xi(i_{\text{node}}), \quad \llbracket \mathbf{q} \rrbracket(\bar{i}_{\text{node}}), \quad \mathbf{n}(\bar{i}_{\text{node}}), \quad A(\bar{i}_{\text{node}}). \quad (4.55)$$

Note that the dataset will be the same for each pair of DoF topologically associated with a given edge, hence it needs only to be defined for one of them.

The validation and application examples below were obtained through an implementation of the FNM in an User Element (UEL) subroutine in the commercial code Abaqus [1]. Floating nodes are preallocated to the elements before analysis.

4.3 Validation

4.3.1 Convergence of Stress Intensity Factors for an edge crack

In this section, we compare the mesh convergence of the PNM and FNM in the evaluation of Stress Intensity Factors (SIF) for an edge crack propagating in mode I (Figure 4.13a). The numerical evaluations for the FNM are performed with VCCT as presented in Section 4.2.9, and for PNM with VCCT as implemented in the commercial software Abaqus [1].

The model has dimensions $W = 9$ mm, $L = 37$ mm and $a = 3$ mm. The Young's modulus is $E = 200$ GPa and the Poisson's ratio is $\nu = 0.3$. The applied stress is $\sigma = 1$ MPa. The plate is discretised uniformly with n_W and n_L four-noded plain stress elements in the width and height directions respectively. The four meshes created have $(n_W, n_L) = (9, 37); (18, 75); (36, 149)$ and $(42, 173)$. Under plain stress conditions, the energy release rates G_I and G_{II} obtained by Equation 4.50 and 4.51 relate to K_I and K_{II} by:

$$K_I = \sqrt{G_I E} \quad (4.56)$$

$$K_{II} = \sqrt{G_{II} E} \quad (4.57)$$

The results are summarized in Figure 4.13b. The FNM can be seen to converge monotonically and more rapidly than the PNM implementation used for comparison.

4.3.2 Evaluation of Stress Intensity Factors for a centre slant crack

In this section, we evaluate the SIF for a centre slant crack (Figure 4.14a) obtained by the FNM against the corresponding analytical solutions [136] in mode I (K_I) and mode II (K_{II}), for different orientations θ of the crack. The numerical evaluations for the FNM are performed with VCCT as presented in Section 4.2.9. For FNM with VCCT, when the crack separates the original element domain into a triangle and a pentagon, both the partitions

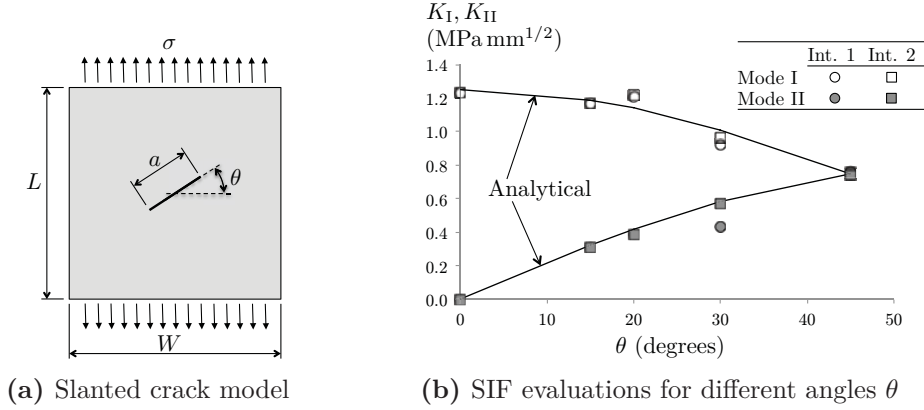


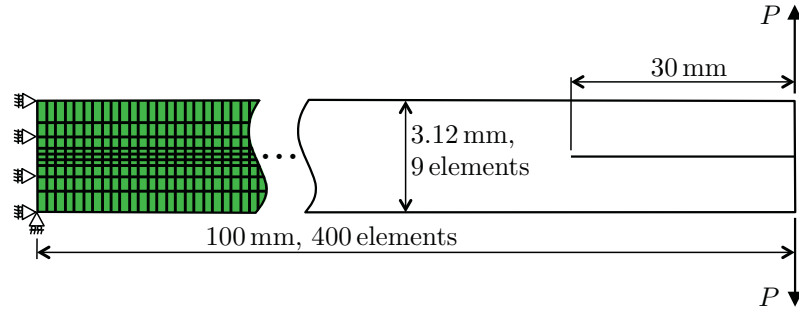
Figure 4.14: For this slant crack model, the FNM captures the SIF well in modes I and II for different angles θ .

shown in Figure 4.6c and that shown in Figure 4.6d are employed. The model has dimensions $L = W = 10 \text{ mm}$ and the horizontal projection of the crack is $a \cos \theta = 0.1W$. The Young's modulus is $E = 200 \text{ GPa}$ and the Poisson's ratio is $\nu = 0.3$. The applied stress is $\sigma = 1 \text{ MPa}$. The plate is discretised uniformly with a coarse mesh (80 elements across the width and 81 elements across the height). Plain stress conditions are assumed; G_I and G_{II} relate to K_I and K_{II} by (4.56) and (4.57).

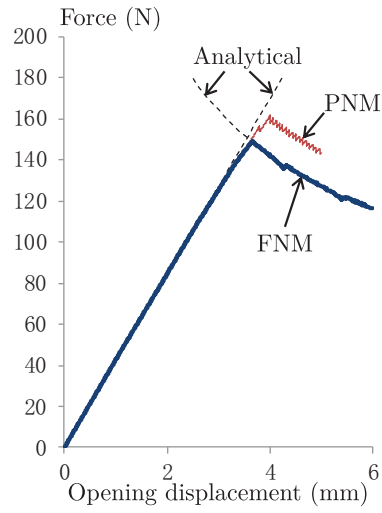
The results are summarized in Figure 4.14b. The data-points labelled 'Int. 1' are obtained with FNM-VCCT using the partition in Figure 4.6c, and the data-points labelled 'Int. 2' are obtained with FNM-VCCT using the partition in Figure 4.6d. For K_I evaluations, the FNM with both integration methods achieves good agreement with the analytical solution for all orientations θ of the crack. For K_{II} evaluations, the FNM with a more refined integration (Int. 2, Figure 4.6d) gives in general slightly more accurate predictions.

4.3.3 Double Cantilever Beam bending test

In this section, a Double Cantilever Beam (DCB) test is used to simulate a propagating crack for a case in which the analytical solution (using corrected beam theory [115]) is known. The material is representative of carbon/PEEK fibre reinforced composite (Table 4.1) and is modelled as an orthotropic material. The geometry, boundary conditions and mesh are de-



(a) DCB model



(b) Predictions

Figure 4.15: DCB validation case, showing that, for the same mesh seeding, the FNM predicts accurately the force (P) vs. displacement curve while the PNM overpredicts the force.

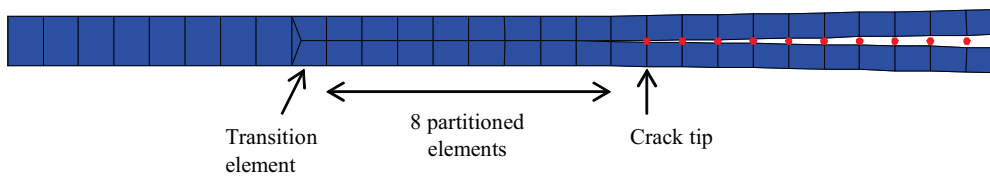


Figure 4.16: DCB simulation using FNM. 8 partitioned elements are present between the crack tip and the transition element. The red dots indicate failure at the corresponding integration points.

Table 4.1: Mechanical properties representative of carbon/PEEK [143] used for the DCB test.

Property	Value
Longitudinal Young's modulus E_1 (GPa)	122.7
Poisson's ratio ν	0.25
Transverse tensile strength Y_t (MPa)	80
Mode I critical energy release rate G_{Ic} (kJ/m ²)	0.969

scribed in Figure 4.15a; the width of the specimen is 25.4 mm; the elements are 0.25 mm in length and the most refined elements near the mid-section are 0.124 mm in height. The cohesive zone approach with a standard bi-linear law and a stress-based criterion as described in Section 4.2.9 is employed to determine the initiation and propagation of a crack. Since the loading is purely in Mode I, the choice of failure initiation criterion and mixed-mode damage propagation law are irrelevant; we therefore use the typical quadratic interaction criteria for both initiation and propagation.

This case is analysed with both the FNM and the PNM (the latter implemented in the commercial software Abaqus [1]). For the FNM, the transition element shown in Figure 4.12f is employed for the element in front of the crack tip. The repartitioning of the transition element into an element containing a cohesive sub-element (Figure 4.5b) is carried out as detailed in Section 4.2.9 with $\epsilon = 0$. The propagation of the crack is then determined by the failure of the cohesive sub-elements. The results are shown in Figure 4.15b. Both methods are able to model the propagation of the crack. A significant over-prediction of the force during propagation is observed for the PNM, suggesting that the high stress concentration in front of the crack tip is not sufficiently captured, resulting in a delayed onset of crack propagation and thus over-prediction of the applied force. For the FNM, the predicted force vs. opening displacement are in good agreement with the analytical solution, showing that the transition element captures well the stress concentration ahead of the crack tip. Figure 4.16 shows the

element topology, deformation and failure prediction of the FNM in the region close to the crack tip.

4.4 Applications

In this section, we analyse the problem of interaction between matrix cracks and delamination in cross-ply laminates, using the FN element presented in Section 4.2.8. Two experimental work in the literature are studied. The first experiment was on a toughened glass/epoxy laminate. It was tested in tension by Joffe and Varna [68], who reported the crack density vs. applied strain. The second work was on three AS4/3501-6 carbon/epoxy laminates of different thickness of 90 plies. This set of experiments was performed by Nairn et al. [99] who reported the crack density vs. applied stress on the laminates.

Failure occurs essentially under tension (mode I) in the 90° ply and under shear (mode II) for the delamination. The choice of failure initiation criterion and mixed-mode damage propagation law are therefore largely irrelevant in this case. We therefore use the typical quadratic interaction criteria for both initiation and propagation. Thermal residual stresses are not considered in the current study and this constitutes a limitation of the model; they would in principle promote the onset of delamination due to the fact that the 90 ply would be in tension. However, the gradual moisture absorption relaxes the thermal stress after curing and thus it is difficult to accurately estimate the residual stresses in the specimen immediately before testing. As the current study focuses on the method development rather than the application, thermal residual stresses are neglected for simplicity; they can however be included as detailed in [56, 146] to improve the accuracy of the approach.

4.4.1 Crack density evolution in a toughened glass/epoxy cross-ply laminate

In this section, the numerical study of the experiment by Joffe and Varna [68] using the FNM is presented. The laminate is of the lay-up $[0_2/90_4]_S$. The total thickness of the $[0_2]$ sub-laminate is $h_1 = 0.288$ mm and the total

thickness of the $[90_8]$ sublaminates is $h_2 = 1.152$ mm. All material properties are given in Table 4.2. Material strength and toughness properties related to matrix cracking and delamination are not all directly available. The transverse tensile strength Y_t is calculated from the transverse Young's modulus E_2 and the experimental transverse tensile failure strain ϵ_2^f [69] by $Y_t = E_2 \epsilon_2^f$. The Mode I critical energy release rate was obtained through Monte-Carlo simulation in [69]. The shear strength and Mode II critical energy release rate of the toughened Glass-Epoxy material used in [68] are not found by the authors; those of other Glass-Epoxy materials that are available in [54] are employed. The delamination strengths and toughness values are assumed to be the same as those for matrix cracking. Since the loading is uniform in tension, a 10% reduction on transverse tensile strength and mode I critical energy release rate is introduced in a selected element at the centre so as to initiate failure at the centre of the model. The model, shown in Figure 4.17a, represents half of the laminate, with symmetric boundary conditions applied on the bottom surface of the 90° plies.

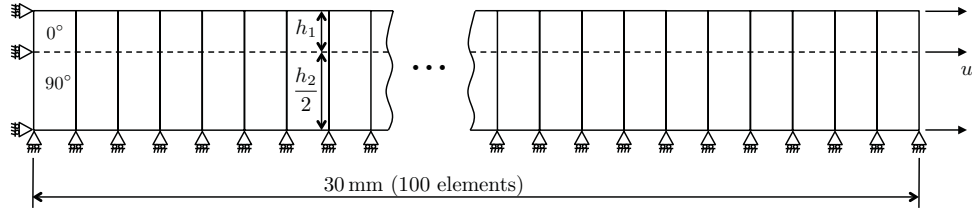
Figures 4.17b and 4.17c show the failure pattern predictions for the laminate when using the FN element from Figure 4.11a. It can be seen that the interaction between matrix cracks and delamination is clearly captured. In the simulation, it is observed that matrix cracks develop first; their density increases and gradually saturates. Delamination starts then to develop and T-shaped crack networks start to form.

To demonstrate how crucial it is to have matching meshes at the crack intersections, a second model was created, differing only in that only one cohesive sub-element is used to model the delamination in each FN element (as in Figure 4.11b), rather than two. The resulting crack pattern, at the same level of strain as in Figures 4.17b and 4.17c, is shown in Figures 4.17d and 4.17e. The second model predicts significantly less delamination than the FNM model. The simulation shows that delamination does not start from the element containing the matrix crack; instead, it occurs firstly in the elements next to the cracked element. This non-physical sequence of delamination propagation is expected in the stress analysis of the non-matching meshes (Figure 4.10a).

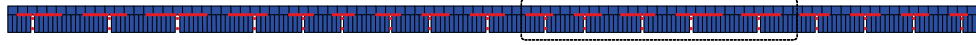
Figure 4.18 shows the crack density vs. applied strain predictions. While

Table 4.2: Mechanical properties of toughened glass/epoxy composite

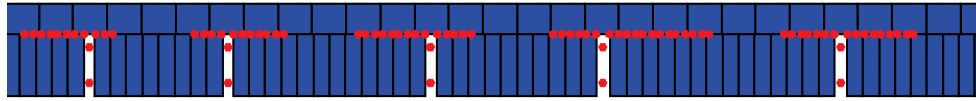
Property	Value	Reference
Longitudinal Young's modulus E_1 (GPa)	44.73	[68]
Transverse Young's modulus E_2 (GPa)	12.76	[68]
Poisson's ratio ν	0.297	[68]
Shear modulus G_{12} (GPa)	5.8	[68]
Transverse tensile failure strain ϵ_2^f	0.78%	[69]
Transverse tensile strength $Y_t = E_2\epsilon_2^f$ (MPa)	100	Calculated
Mode I critical energy release rate G_{Ic} (kJ/m ²)	0.584	[69]
Shear strength S (MPa)	75	[54]
Mode II critical energy release rate G_{IIc} (kJ/m ²)	0.9	[54]



(a) FNM model. Note that there is only one FN element along the height (see Figure 4.11a).



(b) Accurate modelling of the transition from matrix cracking to delamination when using the FN element from Figure 4.11a.



(c) Zoom of FNM mesh in Figure 4.17b.



(d) Model with non-matching mesh at the intersection (using the element partition in Figure 4.11b), showing that some transition to delamination is not correctly captured.



(e) Zoom of non-matching mesh in Figure 4.17d.

Figure 4.17: Modelling the transition from matrix cracking to delamination in a cross-ply composite specimen. The red dots indicate failure at the corresponding integration points.

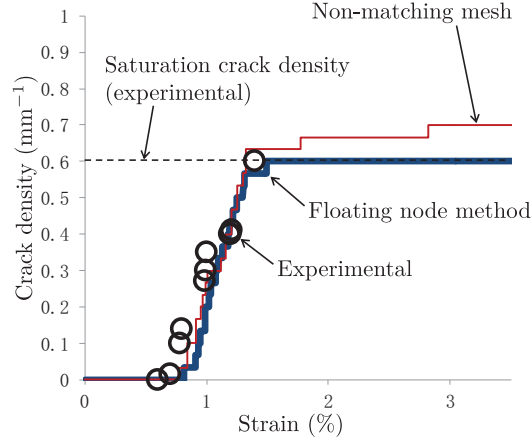


Figure 4.18: The saturation crack density is correctly captured using the FN element from Figure 4.11a. FN element with non-matching partition (Figure 4.11b) results in over-prediction of this density. Experimental data is from Joffe and Varna [68].

both models are able to capture the growth of crack density with applied strain, only the first model (with matching mesh at the crack intersections) is able to predict saturation accurately; the second model continues to predict an increase in crack density, albeit at a lower growth rate, after saturation should have occurred.

4.4.2 Crack density evolution in AS4/3501-6 carbon/epoxy cross-ply laminates

This section presents the FNM modelling of the experiment by Nairn et al. [99] on three cross-ply AS4/3501-6 carbon/epoxy laminates of lay-ups $[0/90]_S$, $[0/90_2]_S$ and $[0/90_4]_S$, respectively. Each ply is roughly 0.125 mm thick. It has been observed that the saturation crack density is inversely proportional to the thickness of the 90 ply-block, and that the tendency for delamination increases with increasing thickness of the 90 ply-block [85, 98, 99]. Figure 4.19 shows the experimental data of matrix crack density at different loading stresses, as well as theoretical predictions based on variational mechanics. The theoretical model gives good prediction of the trend of the effect of 90 plies' thickness on the crack density evolution,

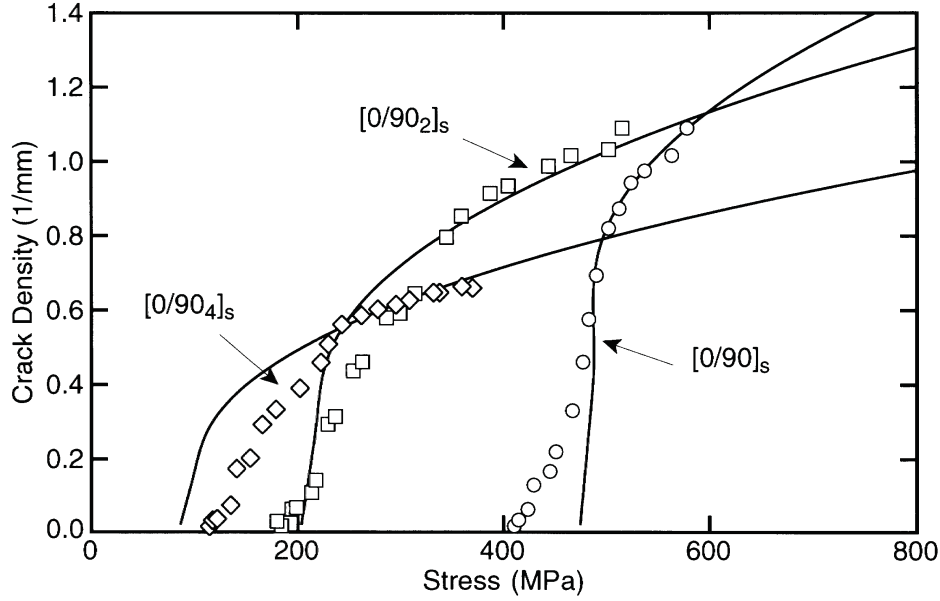


Figure 4.19: Matrix crack density with respect to applied stress for three cross-ply AS4/3501-6 carbon/epoxy composites of different 90 ply block thicknesses. Symbols represent experimental data and solid lines are theoretical predictions based on variational mechanics [98].

but does not provide a clear answer for the saturation crack densities (Figure 4.19). The same FN element in Section 4.4.1 is employed to model this set of experiments to see if the crack density evolution, the saturation of crack density, and the effects of 90 plies' thickness can be predicted.

All the material properties used are summarized in Table 4.3. It has been shown that the transverse strengths of a ply within a laminate are higher than those of a unidirectional lamina [40, 149]. For the problem in this study, the cross-ply laminates are under uniform tension and matrix cracks in 90 plies are essentially caused by Mode I fracture; hence, only the in situ effect of the transverse tensile strength is considered. A formula proposed by Pinho et al. [108, 109] is used to calculate the in situ transverse tensile strength Y_t^{ist} of the 90 plies embedded in the laminates from the unidirectional lamina's transverse tensile strength Y_t^{UD} as follows:

$$Y_t^{\text{ist}} = 1.12 \sqrt{2} Y_t^{\text{UD}} \quad (4.58)$$

Table 4.3: Mechanical properties of AS4/3501-6 carbon/epoxy composite

Property	Value	Reference
Longitudinal Young's modulus E_1 (GPa)	130	[85]
Transverse Young's modulus E_2 (GPa)	9.7	[85]
Poisson's ratio ν	0.3	[85]
Shear modulus G_{12} (GPa)	3.6	[85]
Unidirectional lamina's transverse tensile strength/delamination normal strength Y_t^{UD} (MPa)	53	[96]
In situ transverse tensile strength Y_t^{ist} (MPa)	84 (mode)	Calculated [108]
Matrix cracking G_{Ic} (kJ/m ²)	0.1 (mean); 0.069 (mode)	[87, 114]
Delamination G_{Ic} (kJ/m ²)	0.069	[114]
Matrix cracking/delamination shear strength S (MPa)	42	[96]
Matrix cracking/delamination G_{IIc} (kJ/m ²)	0.56	[114]

The Mode I and II fracture toughness for delamination are available in [114]. It is generally assumed that the fracture toughness for matrix cracking are the same as those for delamination, as the fracture in both cases occur within the epoxy resin or at the fibre/matrix interface.

In the problem of matrix crack density evolution of cross-ply, both the non-uniform stress-state (caused by the first matrix crack and continuously modified by the occurrence of subsequent matrix cracks) and the variations in material properties determine the critical regions of the 90 plies. For a material with a relatively uniform material property distribution without significant variation, the matrix crack initiates at the weakest point and the non-uniform stress state would subsequently lead to the next occurrence of matrix crack at the location of highest transverse tensile stress. In the previous example of the glass/epoxy laminate [69], it is found that introducing a weak point in the domain suffices to trigger an evolution of matrix crack density which agrees very well with the experimental data, suggesting that the material has a relatively uniform property distribution and the non-uniform stress in the domain determines the next occurrence of matrix crack. Therefore, no further statistical variations of material properties are introduced. On the other hand, the AS4/3501-6 material studied [98] has been found to have large variations on the material properties in the literature [87], suggesting that the statistical variation needs to be considered in the modeling; this is also the approach followed by other numerical studies on the same problem in the literature [98, 147]. It is shown in [87] that the value of G_{Ic} in the literature varies from $\approx 0.06 \text{ kJ/m}^2$ to $\approx 0.14 \text{ kJ/m}^2$. For the problem in this study, transverse tensile properties are directly involved in the calculation of matrix cracking; their statistical variations can be expected to have a strong influence on the progression of the crack density accumulation. For this reason, Y_t^{ist} and G_{Ic} for matrix cracking are assumed to follow a log-normal distribution of the same scaling factor. In a log-normal distribution, the mode is the point with the maximum probability density and it is always smaller than the mean. In the DCB tests for the determination of G_{Ic} , the entire interface is fractured. Therefore, the measured G_{Ic} value represents the mean G_{Ic} for different parts of the interface. A mean G_{Ic} value of 0.1 kJ/m^2 is used, obtained from averaging the minimum and maximum values in [87]. The mode G_{Ic} value is assumed

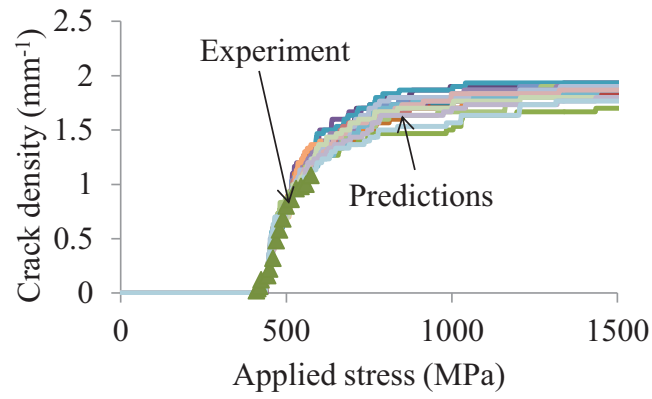
to be 0.069 kJ/m², corresponding to the experimental data in [114]. In transverse tensile tests, only the weakest transverse plane of the lamina is fractured; hence, the experimental Y_t^{UD} represents the weakest strength, rather than the average one, of different parts of the lamina. For this reason, the Y_t^{ist} (calculated from the experimental Y_t^{UD}) is assumed to be the mode, rather than the mean, of the log-normal distribution. In practice, the G_{Ic} of an element is generated according to the log-normal distribution with the mean and mode G_{Ic} values in Table 4.3; the Y_t^{ist} of the element is calculated as follows:

$$Y_t^{ist} \text{ of element } i = \text{mode } Y_t^{ist} \times \frac{G_{Ic} \text{ of element } i}{\text{mode } G_{Ic}} \quad (4.59)$$

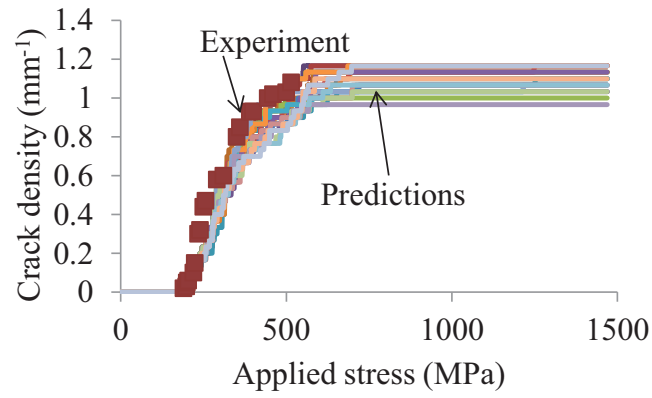
The delamination in this problem is caused by the high stress concentration at the matrix crack tips; the rest part of the interface is subject to a much lower interfacial stress. Therefore, it is assumed that the crack tip stress concentration is the dominant factor in determining the extent of delamination and the role of material property variation is secondary. In addition, the study intends to focus on the delamination induced by matrix crack rather than by initial weakness of the interface. For these reasons, statistical variation of strength and toughness is not included for delamination; however, it can be employed in the same way as for matrix cracking for better completeness and accuracy.

The FEM model is the same as the one shown in Figure 4.17a. The simulation on each of the three laminates was performed twenty times to investigate the effect of the introduced statistical variation, and to obtain a set of meaningful average predictions.

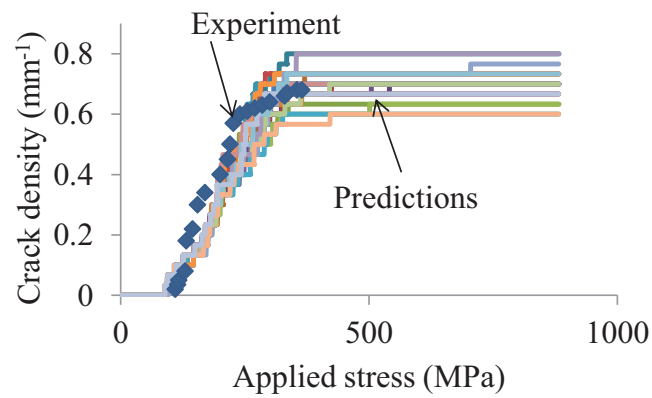
Figure 4.20 shows the comparisons of experimental data with the predictions of all simulation runs on the three cases. It can be seen that the scatters caused by the statistical variations of transverse tensile properties are within a reasonable band which covers the experimental data. The average of the twenty predictions are compared against experimental data in Figure 4.21. Very good correlations are obtained for the laminates of lay-ups $[0/90_4]_S$ and $[0/90_2]_S$, where both the initiation and the final saturation densities of the two laminates are accurately predicted. For the $[0/90]_S$ laminate, the initial part of the curve correlates very well with the experi-



(a) [0/90]_S



(b) [0/90₂]_S



(c) [0/90₄]_S

Figure 4.20: FNM Simulation results of all twenty runs vs. experimental data

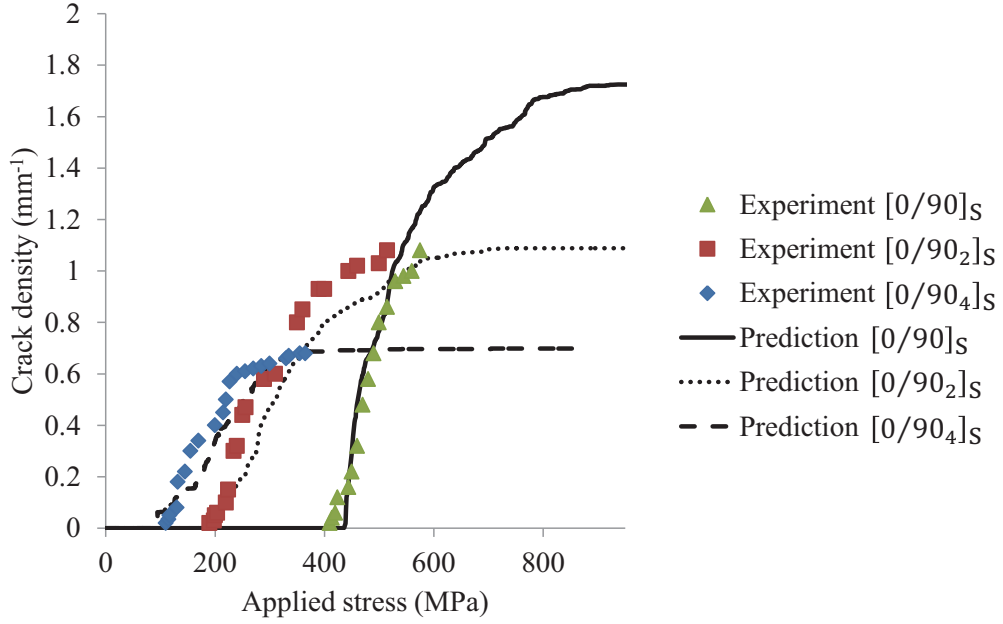


Figure 4.21: Average of the twenty simulation results vs. experimental data

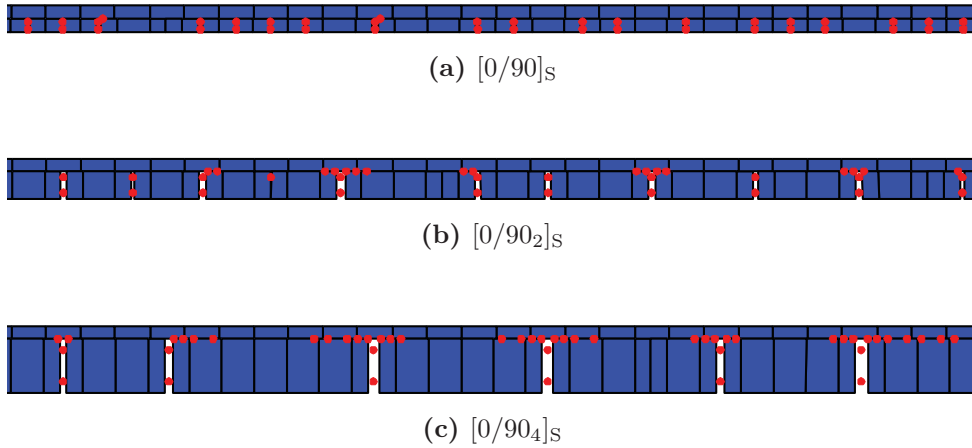


Figure 4.22: Failure pattern predictions for the same section of all three laminates

mental data; however, the predicted saturation density is much higher than the largest recorded experimental crack density. It is also noted that the maximum experimental crack densities in $[0/90_2]_S$ and $[0/90]_S$ laminates are almost the same, which contradicts the general trend observed in the experiment that a cross-ply laminate with thinner 90 plies tends to have a higher saturation crack density [85, 98, 99]. Looking at the end slope of the experimental data for the $[0/90]_S$ laminate, it is clear that the slope at the end is still rising, indicating a potentially premature stoppage of the experiment. In [99], it was written that *"The specimens were continually reloaded into the MTS frame and tested to higher displacements until the end tabs failed, the specimen broke, or delamination began"*, which provides some possible causes of the potentially early termination of experimental recording for the $[0/90]_S$ laminate.

The failure pattern predictions are shown in Figure 4.22 for the same horizontal section of the three laminates. The experimentally-observed thickness effect of the 90 plies in [85, 98, 99] can be clearly observed. The area of delamination increases, and the crack density decreases, with respect to the increasing thickness of the 90 plies from the $[0/90]_S$ laminate to the $[0/90_4]_S$ laminate.

This example, together with the example in Section 4.4.1, demonstrates the capability of the FNM in constructing elements for the modelling of specific geometries and complex crack networks, and in capturing accurately the interactions between matrix cracking and delamination.

4.5 Discussion and conclusion

The essential idea that gives rise to the FNM is that the DoF and the nodal coordinates are regarded as two independent entities, which can be created separately at different stages of the analysis and only become associated with each other when required. A DoF vector can be defined initially without being associated with nodal coordinates, or for the ease of implementation, with random coordinates, and be used to represent the DoF of a chosen material point at a later stage of the analysis; a DoF vector is therefore freed from being constrained to a fixed point in the domain after creation.

In summary, a novel numerical method, the FNM, is proposed for the modelling of complex networks of discontinuities in solids, especially in composites. It is shown that the FNM has a similar computational architecture as that of the PNM (which is equivalent in solution to XFEM with Heaviside enrichment), but offers several advantages over the latter in that:

- the integration is simple, as it does not require numerical integration over *only part* of a domain after transforming to the natural space;
- it can incorporate weak discontinuities and cohesive cracks readily;
- it is ideally suited for the representation of multiple and complex networks of discontinuities, such as the multiple T-shaped cracks in Section 4.4;
- it does not introduce an error on the crack geometry when mapping from physical to natural coordinates; and
- it is conceptually simpler.

It is also demonstrated that the FNM effectively implements local remeshing in terms of solution, but crucially:

- it is computationally more efficient than remeshing (eg. no modification of initial mesh and geometry, and the element connectivities are not changed during the analysis);
- it can be readily implemented in relatively closed FE codes (using user elements); and
- it allows (sub-)elements near a crack tip to readily share information.

Furthermore, the FNM is shown to possess the following capabilities:

- it can incorporate transition elements at crack tips easily;
- it can be readily combined with VCCT;
- it can provide accurate predictions for stress intensity factors under generic mode ratios;

- it can be used to construct a specific element according to the lay-up of the composite laminate so as to model multiple plies and interfaces, and both matrix cracking and delamination, all within such a user-defined element, allowing the immediate exchange of information between different parts of the laminate; and
- it can accurately predict certain interactions between matrix cracking and delamination, by correctly representing the kinematics of the crack intersection.

4.6 Acknowledgement

The author greatly acknowledges the research scholarship from National University of Singapore.

The work in this chapter comes from some earlier study on the PNM which now becomes obsolete; it is presented in Appendix A for reference.

4.7 Publications

The work of this chapter has produced the following publications:

- B. Y. Chen, S. T. Pinho, N. V. De Carvalho, P. M. Baiz, and T. E. Tay. A floating node method for the modelling of discontinuities in composites. *Engineering Fracture Mechanics*, 2014. Submitted
- S. T. Pinho, B. Y. Chen, N. V. De Carvalho, P. M. Baiz, and T. E. Tay. A floating node method for the modelling of discontinuities within a finite element. In *Proceedings of 19th international conference on composite materials, Montréal, Canada*, 2013
- N. V. De Carvalho, B. Y. Chen, S. T. Pinho, P. M. Baiz, and T. E. Tay. Floating node method and virtual crack closure technique for modelling matrix cracking–delamination migration. In *Proceedings of 19th international conference on composite materials, Montréal, Canada*, 2013

- P. M. Baiz, B. Y. Chen, S. T. Pinho, N. V. De Carvalho, and T. E. Tay. Modelling multiple discontinuities within an element for applications in composite fracture. In *Preceedings of UK national conference on computational mechanics in engineering*, 2013

Chapter 5

Conclusions

This research aims to find an accurate model for composites which is capable of capturing the critical failure mechanisms and their interactions in composite laminates, such as those shown in the open-hole tension experiments in Chapter 3 and the matrix crack/delamination interaction in Chapter 4 Section 4.4.

Chapter 3 examines the capabilities and limitations of some commonly-used numerical methods on the open-hole tension problem in [53], using existing failure theories; it also employs the recently-discovered thickness dependence of translaminar fracture toughness [79] in the failure predictions. The work shows that the thickness-dependence of translaminar fracture toughness is an important parameter that needs to be taken into account in the failure modelling of composites; it is used as a novel input in the numerical model for the size effect studies of open-hole tension composite laminates, which helps to accurately predict the thickness size effect of the sublaminate-scaled specimens by a deterministic model.

It is also shown in Chapter 3 that numerical models without considering delamination give mesh-dependent and over-estimated strength predictions and therefore, they are not suitable for the failure modelling of composite laminates. The CSLI model, where the fibre breaking and matrix cracking are modelled by the smeared crack formulation and delamination modelled by cohesive elements, gives generally accurate and mesh-independent predictions when a minimum of two cohesive elements are present within the numerical cohesive zone, the length of which is determined by the for-

mula proposed in [58]. However, for the ply-scaled specimens which fail by delamination, the CSLI model gives good predictions on the failure mode and patterns but exhibits limitations on accurate strength predictions. The predicted strength is shown to be sensitive to the interface shear strength used for the cohesive elements. The smeared crack formulation in the CSLI model smears the matrix crack and its tips. This simplification of the geometrical representation of the crack limits the ability of the method to accurately represent the stress concentration caused by the matrix crack tip at the interface.

A novel numerical method, the FNM, is established in Chapter 4 for the modelling of multiple discontinuities and their interactions within one finite element. The concepts of floating DoFs and floating nodes are introduced. Floating DoFs are used to represent the displacements of the discontinuities; they are not associated with any initial position vector. For implementation in existing FE packages, floating DoFs are created in the form of floating nodes, where random initial coordinates are assigned to these nodes and they are not used throughout the analysis; only the DoF vectors are actually needed to represent the DoFs of the discontinuities. The floating nodes do not coincide with the real nodes as opposed to the case in the PNM; for simplicity, they are assigned nil coordinates, i.e., the floating nodes are positioned at the origin. The FNM shares similarities with the PNM in that both methods model a discontinuity through partitioning the original element into sub-elements. In the PNM, only part of the domain of the sub-element is active, and the DoFs of the discontinuities are interpolated from both the real and the phantom DoFs through shape functions. In the FNM, the sub-elements are defined directly on the material domain partitions, and the floating DoFs directly represent the DoFs of discontinuities. The latter approach possesses several advantages in that:

- the integration is simple, as it does not require numerical integration over *only part* of a domain as in the PNM;
- it leads to exactly the same solution as a finite element mesh where the discontinuity is represented explicitly;
- it can incorporate weak discontinuities and cohesive cracks readily;

- it is ideally suited for the representation of multiple and complex networks of (weak, strong and cohesive) discontinuities;
- unlike XFEM and PNM, it does not introduce an error on the crack geometry when mapping from physical to natural coordinates;
- unlike remeshing, it can be readily implemented in relatively closed FE codes (using user elements);
- it allows (sub-)elements near a crack tip to readily share information.
- it can incorporate transition elements at crack tips easily;
- it can be readily combined with VCCT; and
- it can provide accurate predictions for stress intensity factors under generic mode ratios.

The biggest advantage of the method is its flexibility in modelling multiple and complex networks of different types of discontinuities within a single user-defined element. A FN element can be designed to have a range of different partitions, such that the assembly of the sub-elements in each partition is equivalent to a different FE mesh which solves the problem of a different cracking scenario. In another word, a FN element can be designed to perform the tasks of several different FE meshes, and changing from one task to another requires only local modifications of sub-element domains within the FN element subroutine. When applied to the failure modelling of composites, a special FN element can be constructed for a laminate of a certain lay-up such that:

- it can model multiple plies and interfaces through the thickness of the composite laminate, and both matrix cracking and delamination, within its local element subroutine; and
- it can readily share information between different plies and interfaces, allowing the straightforward implementation of laws and criteria involving the information across the thickness of the laminate.

The cross-ply FN element can accurately predict certain interactions between matrix cracking and delamination which have been shown to be difficult by the conventional approach of modelling (Chapter 4 Section 4.4). The FNM is still in the early stage of development and its potential has not been fully explored. In the next chapter, some possible future work on the FNM will be presented.

Chapter 6

Future work

6.1 Developing a Multi-scale FNM element for laminates

It has been proposed in the conclusions of Chapter 3 that when the failure of the laminate is dominated by brittle fracture, CLT model can be used for the failure modelling of the laminate; and that when the failure of the laminate is pull-out, CSLI model should be used instead. It is only when the failure mode is delamination (such that the explicit representation of matrix cracks may be critical), that the CSLI model becomes insufficient for high-fidelity predictions of laminate strength. A more advanced method (such as the FNM element in Chapter 4 Section 4.4) then needs to be employed.

By exploiting the flexibility of the FNM in the in-situ modification of the element topology through changing partition strategies, the above three models could all be written within the formulation of one FN element. The general FN element would ideally behave as follows:

- initially, its formulation is based on CLT, i.e., linear interpolation of displacement across the thickness of the laminate is used and no delamination is considered;
- when the state of the above element reaches a certain criterion (e.g., delamination initiation or delamination front reaching one of the edges), floating nodes are introduced and the element is partitioned

into a mesh of sub-elements, such that each ply is represented by a single element individually, and each interface is represented by a cohesive element, as in the case of the CSLI model;

- when the state of the above element reaches another criterion which indicates the necessity of having matrix cracks explicitly represented, the element can be repartitioned such that each matrix crack tips are embedded in the cohesive elements at the interface, as shown in Chapter 4 Section 4.2.8. This element topology is denoted by the name ‘FNM’ in Figure 6.1.

In this way, the mesh of FNM elements would automatically have CLT formulation for non-critical areas of the laminate, the FNM formulation for critical areas of the laminate involving significant interaction between matrix cracking and delamination, and the CSLI formulation for areas having delamination passing through or having delamination front at an edge (Figure 6.1). This strategy is expected to achieve both accuracy and economy in the calculation, at the expense of building more complexity in the partitioning strategies of the FN element. Implementation details such as the accuracy in the connection between neighbouring FN elements of different formulations and the accuracy of the CLT formulation for thick laminates, etc., need to be addressed.

6.2 Extension of the FNM to higher-order and 3D elements

Currently, the FNM is implemented only for 2D linear elements. Practical application of this method would require that it is implemented for higher-order and 3D elements. The extensions of the FNM into second-order elements and 3D elements do not have any theoretical barrier. However, implementation issues with respect to the geometrical partitioning of the element domain (when cut by a discontinuity) need to be considered. In 3D situation, a crack may cut an element domain into sub-domains of complex geometries. The partitioning of these sub-domains into regular 3D sub-elements may require complex algorithms.

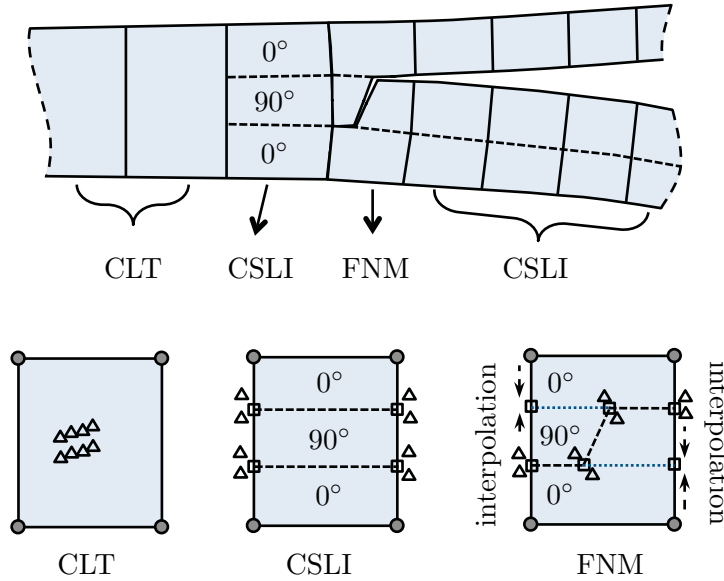


Figure 6.1: A multi-scale FN element which can incorporate three different topologies for the modelling of failure at different scales, e.g., on the problem of 2D delamination migration in a cross-ply laminate (see key in Figure 4.4)

6.3 Reliable strength predictions of the ply-scaled open-hole tension laminates

In Chapter 3 Section 3.4.5, it has been shown that the CSLI model has limitations in the accurate strength prediction of the ply-scaled open-hole tension specimens (which fail in the delamination failure mode) due to the model's sensitivity on the value of interfacial shear strength. This limitation could be due to the smearing of matrix cracks associated with the smeared crack approach in the CSLI model. It was discussed in Chapter 3 Section 3.4.4, and later shown in Chapter 4 Section 4.4 that the smeared crack approach for matrix cracking has difficulties capturing the stress concentration at the intersections of matrix cracks and interfaces. The FNM has been shown to predict accurately the matrix crack/delamination prediction in a cross-ply laminate in Chapter 4 Section 4.4. The FNM could be applied to the ply-scaling open-hole tension problem to see if more reliable strength predictions can be obtained compared to those by the CSLI model. A FN element could be constructed for the quasi-isotropic lami-

nate, in a way similar to that presented in Chapter 4 Section 4.4, such that the intersections of matrix cracks and interfaces are explicitly represented to allow for more accurate predictions of the stress concentrations in those regions.

6.4 Modelling of delamination migration in composite laminates

Delamination migration is the phenomenon of sudden jump of propagation plane for delamination within a laminate (Figure 6.2). This phenomenon typically happens in the DCB or ENF tests of the laminate. A clear mechanism of interaction between the delamination and the matrix cracking is visible at the point of migration. Accurate modelling of this migration phenomenon requires physically-based migration criteria to accurately determine the conditions of migration, and adequate numerical representation of the kinematics of the migration. The FNM, with its flexibility in the representation of complex crack networks, is particularly well-suited for this problem. A specific FN element could be constructed for the laminate in the same way as shown in Chapter 4 Section 4.4, so that the geometry of the migration crack path could be accurately represented within the FN element. A particular advantage of the FN element on the migration problem is that all the information through the thickness of the laminate is directly available within each element (i.e., within one UEL subroutine), such that complex migration criteria (which may need to simultaneously access information across different plies and interfaces) could be easily implemented. This advantage may also be useful for other problems involving bimaterial interfaces or trimaterial junctions, where the application of a constitutive theory may require the simultaneous access of information across several material domains and their interfaces.

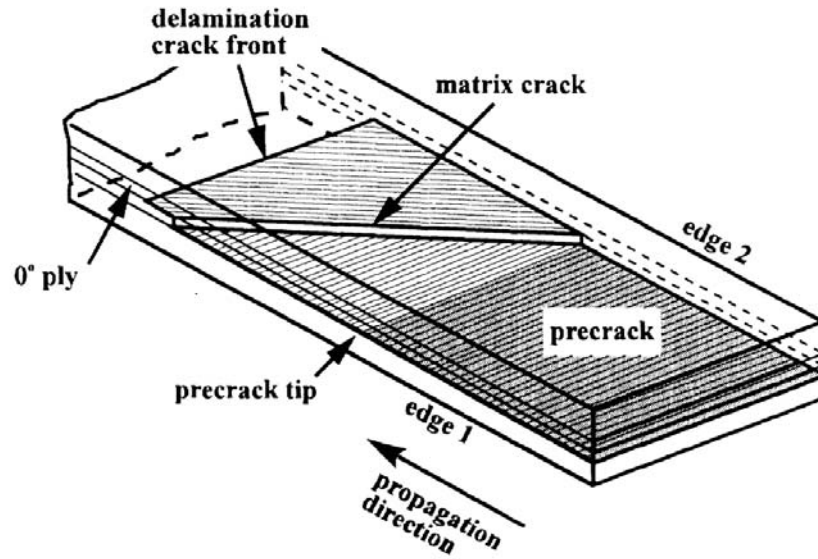


Figure 6.2: Delamination migration in a multi-directional composite laminate from Tao and Sun [138]

6.5 Evaluation of the edge status variable approach of the FNM in representing a large number of discontinuities

The progressive failure of composite materials typically involves a large number of matrix cracks in different plies. The FNM edge status variable approach for the representation of discontinuities presented in Chapter 4 Section 4.2.9 could be particularly advantageous for the modelling of complex crack networks involving a large number of discontinuities. In such situations, rather than tracing the crack tips of all the new and existing cracks, each element only needs to keep updated its own edge status variables to represent the discontinuities reaching its boundaries or passing through its domain. The continuity of discontinuities is naturally guaranteed by the sharing of edge status variables between neighbouring elements. In this way, the number of sets of information that each element accesses to keep track of the evolution of an arbitrary number of discontinuities is equal to the number of edges the element has (under the assumption that

each edge can only crack once). A comparative study could be performed to evaluate the computational time that the edge status variable approach would consume when modelling an increasingly large number of discontinuities, compared to that by the conventional methods used in XFEM or PNM which typically track the global coordinates of all crack fronts.

6.6 Error estimation and adaptivity

The adaptive FEM employs the a-posteriori error estimation to calculate the error of the current FE solution with respect to an approximated real solution (the so called recovered solution), and adaptivity refers to the process of iteratively modifying the element formulation or the mesh until this estimated error reduces within an acceptable tolerance [159]. Different types of adaptivity exist, such as the H-adaptivity, P-adaptivity, and R-adaptivity, etc. The a-posteriori error estimation and adaptivity have also been developed for XFEM in the so called E-adaptivity, where elements are adaptively enriched with XFEM enrichment functions according to the specific error estimator, so as to better approximate the stress field in the domain [47]. Error estimation and adaptivity are areas of research targeting at improving the smoothness of the FE solution, not directly associated with the kinematic modelling of crack propagation; they can be used together with methods such as remeshing or XFEM to improve the solution accuracy along with the modelling of crack propagation. A-posteriori error estimation and adaptivity have not been considered in this work. However, they can be studied in the future work of the FNM, where a suitable error estimator and an adaptivity scheme could be implemented to improve the solution accuracy in the domain.

6.7 Strain smoothing in distorted sub-elements

In the FNM, distorted sub-elements may arise since the crack may cut through the element in an arbitrary angle. At the moment, no special treatment has been performed to improve the accuracy of the distorted

sub-elements in the FNM. Recently, a Smoothed Finite Element Method (S-FEM) has been introduced to allow accurate solutions to be obtained on heavily distorted elements [84]. The method uses the strain smoothing technique, so that the stiffness integration can be performed in the physical space directly on the shape functions using line integrals, rather than on the derivatives of shape functions using volume integrals [25, 67]. Strain smoothing could be implemented in the FNM for the stiffness integration of the distorted sub-elements to improve the solution accuracy of the method.

Bibliography

- [1] Dassault Systèmes Simulia Corp. *Abaqus 6.10 documentation*. Dassault Systèmes, Providence, RI, USA, 2010.
- [2] E. Abisset, F. Daghia, and P. Ladevèze. On the validation of a damage mesomodel for laminated composites by means of open-hole tensile tests on quasi-isotropic laminates. *Composites Part A: Applied Science and Manufacturing*, 42(10):1515 – 1524, 2011.
- [3] G. Alfano. On the influence of the shape of the interface law on the application of cohesive-zone models. *Composites Science and Technology*, 66(6):723–730, 2006.
- [4] F. M. Andrade Pires, E. A. de Souza Neto, and D. R. J. Owen. On the finite element prediction of damage growth and fracture initiation in finitely deforming ductile materials. *Computer Methods in Applied Mechanics and Engineering*, 193(48):5223–5256, 2004.
- [5] P. M. A. Areias and T. Belytschko. A comment on the article ”A finite element method for simulation of strong and weak discontinuities in solid mechanics” By A. Hansbo and P. Hansbo [Comput. methods Appl. Mech. Engrg. 193 (2004) 3523-3540]. *Computer Methods in Applied Mechanics and Engineering*, 195(9-12):1275–1276, 2006.
- [6] Pedro M. A. Areias, J. H. Song, and Ted Belytschko. Analysis of fracture in thin shells by overlapping paired elements. *Computer Methods in Applied Mechanics and Engineering*, 195(41-43):5343–5360, 2006.
- [7] S. E. Ashari and S. Mohammadi. Delamination analysis of composites by new orthotropic bimaterial extended finite element method.

- International Journal for Numerical Methods in Engineering*, 86(13): 1507–1543, 2011.
- [8] H. Askes and L. J. Sluys. Remeshing strategies for adaptive ALE analysis of strain localisation. *European Journal of Mechanics A-Solids*, 19(3):447–467, 2000.
 - [9] H. Azadi and A. R. Khoei. Numerical simulation of multiple crack growth in brittle materials with adaptive remeshing. *International Journal for Numerical Methods in Engineering*, 85:1017–1048, 2011.
 - [10] I. Babuška and B. Q. Guo. The h, p and h-p version of the finite element method: basis theory and applications. *Advances in Engineering Software*, 15(3-4):159–174, November 1992.
 - [11] P. M. Baiz, B. Y. Chen, S. T. Pinho, N. V. De Carvalho, and T. E. Tay. Modelling multiple discontinuities within an element for applications in composite fracture. In *Preceedings of UK national conference on computational mechanics in engineering*, 2013.
 - [12] G. I. Barenblatt. The mathematical theory of equilibrium cracks in brittle fracture. volume 7 of *Advances in Applied Mechanics*, pages 55 – 129. Elsevier, 1962.
 - [13] Z. P. Bažant. Instability, ductility, and size effect in strain-softening concrete. *Journal of the Engineering Mechanics Division-ASCE*, 102(2):331–344, 1976.
 - [14] Z. P. Bažant. Scaling laws in mechanics of failure. *Journal of Engineering Mechanics-ASCE*, 119(9):1828–1844, 1993.
 - [15] Z. P. Bažant. Scaling of quasibrittle fracture: Asymptotic analysis. *International Journal of Fracture*, 83(1):19–40, 1997.
 - [16] Z. P. Bažant and B. H. Oh. Crack band theory for fracture of concrete. *Matériaux et Construction*, 16(3):155–177, 1983.
 - [17] Z. P. Bažant and Y. P. Xi. Statistical size effect in quasi-brittle structures .2. nonlocal theory. *Journal of Engineering Mechanics-ASCE*, 117(11):2623–2640, 1991.

- [18] Z. P. Bažant and Q. Yu. Universal size effect law and effect of crack depth on quasi-brittle structure strength. *Journal of Engineering Mechanics-ASCE*, 135(2):78–84, 2009.
- [19] Z. P. Bažant, Y. P. Xi, and S. G. Reid. Statistical size effect in quasi-brittle structures .1. is weibull theory applicable. *Journal of Engineering Mechanics-ASCE*, 117(11):2609–2622, 1991.
- [20] Z. P. Bažant, J. J. H. Kim, I. M. Daniel, E. Becq-Giraudon, and G. Zi. Size effect on compression strength of fiber composites failing by kink band propagation. *International Journal of Fracture*, 95(1-4): 103–141, 1999.
- [21] T. Belytschko and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45(5):601–620, 1999.
- [22] T. Belytschko, N. Moes, S. Usui, and C. Parimi. Arbitrary discontinuities in finite elements. *International Journal for Numerical Methods in Engineering*, 50(4):993–1013, 2001.
- [23] Ted Belytschko, Robert Gracie, and Giulio Ventura. A review of extended/generalized finite element methods for material modeling. *Modelling and Simulation in Materials Science and Engineering*, 17 (4), 2009.
- [24] M. L. Benzeggagh and M. Kenane. Measurement of mixed-mode delamination fracture toughness of unidirectional glass/epoxy composites with mixed-mode bending apparatus. *Composites Science and Technology*, 56(4):439–449, 1996.
- [25] Stéphane Bordas, Timon Rabczuk, Nguyen-Xuan Hung, Vinh Phu Nguyen, Sundararajan Natarajan, Tino Bog, Do Minh Quan, and Nguyen Vinh Hiep. Strain smoothing in fem and xfem. *Computers & structures*, 88(23):1419–1443, 2010.
- [26] H. Borouchaki, P. Laug, A. Cherouat, and K. Saanouni. Adaptive remeshing in large plastic strain with damage. *International Journal for Numerical Methods in Engineering*, 63(1):1–36, 2005.

- [27] A. J. Brunner, B. R. K. Blackman, and Peter Davies. A status report on delamination resistance testing of polymer-matrix composites. *Engineering Fracture Mechanics*, 75(9):2779–2794, 2008.
- [28] P. P. Camanho, C. G. Dávila, and M. F. de Moura. Numerical simulation of mixed-mode progressive delamination in composite materials. *Journal of Composite Materials*, 37(16):1415–1438, 2003.
- [29] P. P. Camanho, P. Maimi, and C. G. Dávila. Prediction of size effects in notched laminates using continuum damage mechanics. *Composites Science and Technology*, 67(13):2715–2727, 2007.
- [30] Luis Pablo Canal, Carlos González, Javier Segurado, and Javier LLorca. Intraply fracture of fiber-reinforced composites: Microscopic mechanisms and modeling. *Composites Science and Technology*, 72(11):1223–1232, 2012.
- [31] N. Carrere, F. Laurin, and J-F Maire. Micromechanical-based hybrid mesoscopic 3D approach for non-linear progressive failure analysis of composite structures. *Journal of Composite Materials*, 46:2389–2415, 2012.
- [32] G. Catalanotti, P. P. Camanho, J. Xavier, C. G. Dávila, and A. T. Marques. Measurement of resistance curves in the longitudinal failure of composites using digital image correlation. *Composites Science and Technology*, 70(13):1986–1993, 2010.
- [33] Thanh Chau-Dinh, Goangseup Zi, Phill-Seung Lee, Timon Rabczuk, and Jeong-Hoon Song. Phantom-node method for shell models with arbitrary cracks. *Computers & Structures*, 92-93(0):242 – 256, 2012.
- [34] B. Y. Chen, T. E. Tay, P. M. Baiz, and S. T. Pinho. Size effects in progressive damage of notched and holed composites. In *Proceedings of 18th international conference on composite materials, Jeju, Korea Aug*, 2011.
- [35] B. Y. Chen, P. M. Baiz, S. T. Pinho, and T. E. Tay. An extended phantom node method for crack interactions in composites. In *Proceedings of the 20th UK Conference of the Association for Compu-*

tational Mechanics in Engineering, the University of Manchester, Manchester, UK, March 2012.

- [36] B. Y. Chen, S. T. Pinho, P. M. Baiz, and T. E. Tay. An extended phantom node method for crack interactions in composites. In *Proceedings of the 10th World Congress on Computational Mechanics, São Paulo, Brazil*, number 19582, July 2012.
- [37] B. Y. Chen, T. E. Tay, P. M. Baiz, and S. T. Pinho. Numerical analysis of size effects on open-hole tensile composite laminates. *Composites Part A: Applied Science and Manufacturing*, 47(0):52 – 62, 2013.
- [38] B. Y. Chen, S. T. Pinho, N. V. De Carvalho, P. M. Baiz, and T. E. Tay. A floating node method for the modelling of discontinuities in composites. *Engineering Fracture Mechanics*, 2014. Submitted.
- [39] B. Cox and Q. D. Yang. In quest of virtual tests for structural composites. *Science*, 314(5802):1102–1107, 2006.
- [40] F. W. Crossman and A. S. D. Wang. The dependence of transverse cracking and delamination on ply thickness in graphite/epoxy laminates. *Damage in Composite Materials, ASTM STP*, 775:118–139, 1982.
- [41] C. G. Dávila, P. P. Camanho, and C. A. Rose. Failure criteria for FRP laminates. *Journal of Composite Materials*, 39(4):323–345, 2005.
- [42] C. G. Dávila, C. A. Rose, and P. P. Camanho. A procedure for superposing linear cohesive laws to represent multiple damage mechanisms in the fracture of composites. *International Journal of Fracture*, 158(2):211–223, 2009.
- [43] R. de Borst. Numerical aspects of cohesive-zone models. *Engineering Fracture Mechanics*, 70(14):1743–1757, 2003.
- [44] N. V. De Carvalho, B. Y. Chen, S. T. Pinho, P. M. Baiz, and T. E. Tay. Floating node method and virtual crack closure technique for modelling matrix cracking–delamination migration. In *Proceedings*

of 19th international conference on composite materials, Montréal, Canada, 2013.

- [45] H. Matthias Deuschle and Bernd-H Kroeplin. Finite element implementation of Puck’s failure theory for fibre-reinforced composites under three-dimensional stress. *Journal of Composite Materials*, 46: 2485–2513, 2012.
- [46] Qinglin Duan, Jeong-Hoon Song, Thomas Menouillard, and Ted Belytschko. Element-local level set method for three-dimensional dynamic crack growth. *International Journal for Numerical Methods in Engineering*, 80(12):1520–1543, 2009.
- [47] Marc Duflot and Stéphane Bordas. A posteriori error estimation for extended finite elements by an extended global recovery. *International Journal for Numerical Methods in Engineering*, 76(8):1123–1138, 2008.
- [48] D. S. Dugdale. Yielding of steel sheets containing slits. *Journal of the Mechanics and Physics of Solids*, 8(2):100 – 104, 1960.
- [49] X. J. Fang, Q. D. Yang, B. N. Cox, and Z. Q. Zhou. An augmented cohesive zone element for arbitrary crack coalescence and bifurcation in heterogeneous materials. *International Journal for Numerical Methods in Engineering*, 88:841–861, 2011.
- [50] X. J. Fang, Z. Q. Zhou, B. N. Cox, and Q. D. Yang. High-fidelity simulations of multiple fracture processes in a laminated composite in tension. *Journal of the Mechanics and Physics of Solids*, 59(7): 1355–1373, 2011.
- [51] Thomas-Peter Fries and Ted Belytschko. The extended/generalized finite element method: An overview of the method and its applications. *International Journal for Numerical Methods in Engineering*, 84(3):253–304, 2010.
- [52] YF Gao and AF Bower. A simple technique for avoiding convergence problems in finite element simulations of crack nucleation and growth

- on cohesive interfaces. *Modelling and Simulation in Materials Science and Engineering*, 12(3):453, 2004.
- [53] B. G. Green, M. R. Wisnom, and S. R. Hallett. An experimental investigation into the tensile strength scaling of notched composites. *Composites Part A-Applied Science and Manufacturing*, 38(3):867–878, 2007.
 - [54] S. R. Hallett and M. R. Wisnom. Numerical investigation of progressive damage and the effect of layup in notched tensile tests. *Journal of Composite Materials*, 40(14):1229–1245, 2006.
 - [55] S. R. Hallett, W. G. Jiang, B. Khan, and M. R. Wisnom. Modelling the interaction between matrix cracks and delamination damage in scaled quasi-isotropic specimens. *Composites Science and Technology*, 68(1):80–89, 2008.
 - [56] S. R. Hallett, B. G. Green, W. G. Jiang, and M. R. Wisnom. An experimental and numerical investigation into the damage mechanisms in notched composites. *Composites Part A-Applied Science and Manufacturing*, 40(5):613–624, 2009.
 - [57] A. Hansbo and P. Hansbo. A finite element method for the simulation of strong and weak discontinuities in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 193(33-35):3523–3540, 2004.
 - [58] P. W. Harper and S. R. Hallett. Cohesive zone length in numerical simulations of composite delamination. *Engineering Fracture Mechanics*, 75(16):4774–4792, 2008.
 - [59] T. Hettich, A. Hund, and E. Ramm. Modeling of failure in composites by X-FEM and level sets within a multiscale framework. *Computer Methods in Applied Mechanics and Engineering*, 197(5):414–424, 2008.
 - [60] A. Hillerborg, M. Mod  er, and P.-E. Petersson. Analysis of crack formation and crack growth in concrete by means of fracture mechanics

- and finite elements. *Cement and Concrete Research*, 6(6):773 – 781, 1976.
- [61] M. J. Hinton and A. S. Kaddour. Triaxial test results for fibre-reinforced composites: The Second World-Wide Failure Exercise benchmark data. *Journal of Composite Materials*, 47(6-7):653–678, 2013.
 - [62] M. J. Hinton, A. S. Kaddour, and P. D. Soden. A comparison of the predictive capabilities of current failure theories for composite laminates, judged against experimental evidence. *Composites Science and Technology*, 62(12-13):1725–1797, 2002.
 - [63] D. B. P. Huynh and T. Belytschko. The extended finite element method for fracture in composite materials. *International Journal for Numerical Methods in Engineering*, 77(2):214–239, 2009.
 - [64] E. V. Iarve. Mesh independent modelling of cracks by using higher order shape functions. *International Journal for Numerical Methods in Engineering*, 56(6):869–882, 2003.
 - [65] Endel V. Iarve, Mark R. Gurvich, David H. Mollenhauer, Cheryl A. Rose, and Carlos G. Dávila. Mesh-independent matrix cracking and delamination modeling in laminated composites. *International Journal for Numerical Methods in Engineering*, 88(8):749–773, 2011.
 - [66] W. G. Jiang, S. R. Hallett, B. G. Green, and M. R. Wisnom. A concise interface constitutive law for analysis of delamination and splitting in composite materials and its application to scaled notched tensile specimens. *International Journal for Numerical Methods in Engineering*, 69(9):1982–1995, 2007.
 - [67] Y Jiang, TE Tay, L Chen, and XS Sun. An edge-based smoothed xfem for fracture in composite materials. *International Journal of Fracture*, 179(1-2):179–199, 2013.
 - [68] R. Joffe and J. Varna. Analytical modeling of stiffness reduction in symmetric and balanced laminates due to cracks in 90° layers. *Composites Science and Technology*, 59(11):1641 – 1652, 1999.

- [69] R. Joffe, A. Krasnikovs, and J. Varna. COD-based simulation of transverse cracking and stiffness reduction in [S/90n]s laminates. *Composites Science and Technology*, 61(5):637 – 656, 2001.
- [70] A. S. Kaddour and M. J. Hinton. Benchmarking of triaxial failure criteria for composite laminates: Comparison between models of ‘Part (A)’ of ‘WWFE-II’. *Journal of Composite Materials*, 46(19-20):2595–2634, 2012.
- [71] A. S. Kaddour and M. J. Hinton. Maturity of 3D failure criteria for fibre-reinforced composites: Comparison between theories and experiments: Part B of WWFE-II. *Journal of Composite Materials*, 47(6-7):925–966, 2013.
- [72] A. S. Kaddour, M. J. Hinton, S. Li, and P. Smith. Instructions to Contributors of the Third World-Wide Failure Exercise (WWFE-III): Part (A). 2008.
- [73] Luiz F. Kawashita, Alexandre Bedos, and Stephen R. Hallett. Modelling mesh independent transverse cracks in laminated composites with a simplified cohesive segment method. *Computers, Materials & Continua*, 32(2):133–158, 2012.
- [74] A.R. Khoei, M. Eghbalian, H. Moslemi, and H. Azadi. Crack growth modeling via 3d automatic adaptive mesh refinement based on modified-spr technique. *Applied Mathematical Modelling*, 37:357 – 383, 2013.
- [75] C. Koenke, R. Harte, W. B. Krätzig, and O. Rosenstein. On adaptive remeshing techniques for crack simulation problems. *Engineering Computations*, 15(1):74–88, 1998.
- [76] R. Krueger. Virtual crack closure technique: history, approach and applications. *Applied Mechanics Reviews*, 57:109–143, 2004.
- [77] Roman Lackner and Herbert A. Mang. Adaptivity in computational mechanics of concrete structures. *International Journal for Numerical and Analytical Methods in Geomechanics*, 25(7):711–739, 2001.

- [78] M. J. Laffan, S. T. Pinho, P. Robinson, and L. Iannucci. Measurement of the in situ ply fracture toughness associated with mode I fibre tensile failure in FRP. Part I: Data reduction. *Composites Science and Technology*, 70(4):606–613, 2010.
- [79] M. J. Laffan, S. T. Pinho, P. Robinson, and L. Iannucci. Measurement of the in situ ply fracture toughness associated with mode I fibre tensile failure in FRP. Part II: Size and lay-up effects. *Composites Science and Technology*, 70(4):614–621, 2010.
- [80] M. J. Laffan, S. T. Pinho, P. Robinson, and A. J. McMillan. Translaminar fracture toughness: The critical notch tip radius of 0° plies in CFRP. *Composites Science and Technology*, 72(1):97 – 102, 2011.
- [81] X. Li, S. R. Hallett, and M. R. Wisnom. Predicting the effect of through-thickness compressive stress on delamination using interface elements. *Composites Part A-Applied Science and Manufacturing*, 39(2):218–230, 2008.
- [82] X. Q. Li, S. R. Hallett, M. R. Wisnom, N. Zobeiry, R. Vaziri, and A. Poursartip. Experimental study of damage propagation in over-height compact tension tests. *Composites Part A-Applied Science and Manufacturing*, 40(12):1891–1899, 2009.
- [83] D. S. Ling, Q. D. Yang, and B. Cox. An augmented finite element method for modeling arbitrary discontinuities in composite materials. *International Journal of Fracture*, 156(1):53–73, 2009.
- [84] GR Liu, KY Dai, and TT Nguyen. A smoothed finite element method for mechanics problems. *Computational Mechanics*, 39(6):859–877, 2007.
- [85] Siulie Liu and John A. Nairn. The formation and propagation of matrix microcracks in cross-ply laminates during static loading. *Journal of Reinforced Plastics and Composites*, 11(2):158–178, 1992.

- [86] S. H. Lo and C. K. Lee. Solving crack problems by an adaptive refinement procedure. *Engineering Fracture Mechanics*, 43(2):147–163, 1992.
- [87] Mary J. Mathews and Stephen R. Swanson. Characterization of the interlaminar fracture toughness of a laminated carbon/epoxy composite. *Composites Science and Technology*, 67(7):1489–1498, 2007.
- [88] J. M. Melenk and I. Babuška. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):289–314, 1996.
- [89] J. Mergheim. A variational multiscale method to model crack propagation at finite strains. *International Journal for Numerical Methods in Engineering*, 80(3):269–289, 2009.
- [90] J. Mergheim, E. Kuhl, and P. Steinmann. A finite element method for the computational modelling of cohesive cracks. *International Journal for Numerical Methods in Engineering*, 63(2):276–289, 2005.
- [91] J. Mergheim, E. Kuhl, and P. Steinmann. Towards the algorithmic treatment of 3D strong discontinuities. *Communications in Numerical Methods in Engineering*, 23(2):97–108, 2007.
- [92] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46(1):131–150, 1999.
- [93] H. Moslemi and A. R. Khoei. 3d adaptive finite element modeling of non-planar curved crack growth using the weighted superconvergent patch recovery method. *Engineering Fracture Mechanics*, 76:1703 – 1728, 2009.
- [94] K. S. R. K. Murthy and M. Mukhopadhyay. Adaptive finite element analysis of mixed-mode crack problems with automatic mesh generator. *International Journal for Numerical Methods in Engineering*, 49(8):1087–1100, 2000.

- [95] K. S. R. K. Murthy and M. Mukhopadhyay. Adaptive finite element analysis of mixed-mode fracture problems containing multiple crack-tips with an automatic mesh generator. *International Journal of Fracture*, 108(3):251–274, 2001.
- [96] M Naderi and AR Maligno. Fatigue life prediction of carbon/epoxy laminates by stochastic numerical simulation. *Composite Structures*, 94(3):1052–1059, 2012.
- [97] T. Nagashima and H. Suemasu. X-FEM analyses of a thin-walled composite shell structure with a delamination. *Computers & Structures*, 88(9-10):549–557, 2010.
- [98] John A Nairn. Matrix microcracking in composites. *Comprehensive Composite Materials*, 2(12):403–432, 2000.
- [99] John A Nairn, Shoufeng Hu, and Jong Seok Bark. A critical evaluation of theories for predicting microcracking in composite laminates. *Journal of Materials Science*, 28(18):5099–5111, 1993.
- [100] J. Oliver. A consistent characteristic length for smeared cracking models. *International Journal for Numerical Methods in Engineering*, 28:461–474, 1989.
- [101] Borek Patzák and Milan Jirásek. Adaptive resolution of localized damage in quasi-brittle materials. *Journal of Engineering Mechanics*, 130(6):720–732, 2004.
- [102] Soraia Pimenta and Silvestre T. Pinho. Hierarchical scaling law for the strength of composite fibre bundles. *Journal of the Mechanics and Physics of Solids*, 61(6):1337–1356, 2013.
- [103] S. T. Pinho, L. Iannucci, and P. Robinson. Formulation and implementation of decohesion elements in an explicit finite element code. *Composites Part A-Applied Science and Manufacturing*, 37(5):778–789, 2006.
- [104] S. T. Pinho, L. Iannucci, and P. Robinson. Physically-based failure models and criteria for laminated fibre-reinforced composites with

- emphasis on fibre kinking: Part I: Development. *Composites Part A-Applied Science and Manufacturing*, 37(1):63–73, 2006.
- [105] S. T. Pinho, L. Iannucci, and P. Robinson. Physically based failure models and criteria for laminated fibre-reinforced composites with emphasis on fibre kinking. Part II: FE implementation. *Composites Part A-Applied Science and Manufacturing*, 37(5):766–777, 2006.
- [106] S. T. Pinho, R. Darvizeh, P. Robinson, C. Schuecker, and P. P. Camanho. Material and structural response of polymer-matrix fibre-reinforced composites. *Journal of Composite Materials*, 46:2313–2341, 2012.
- [107] S. T. Pinho, B. Y. Chen, N. V. De Carvalho, P. M. Baiz, and T. E. Tay. A floating node method for the modelling of discontinuities within a finite element. In *Proceedings of 19th international conference on composite materials, Montréal, Canada*, 2013.
- [108] S. T. Pinho, G. M. Vyas, and P. Robinson. Material and structural response of polymer-matrix fibre-reinforced composites: Part b. *Journal of Composite Materials*, 47(6-7):679–696, 2013.
- [109] Silvestre T. Pinho, Carlos G. Dàvila, Pedro P. Camanho, Lorenzo Iannucci, and Paul Robinson. Failure models and criteria for FRP under in-plane or three-dimensional stress states including shear non-linearity. *NASA STI Report*, 2005.
- [110] J. Planas, M. Elices, G. V. Guinea, F. J. Gomez, D. A. Cendon, and I. Arbilla. Generalizations and specializations of cohesive crack models. *Engineering Fracture Mechanics*, 70(14):1759–1776, 2003.
- [111] A. Puck and H. Schurmann. Failure analysis of FRP laminates by means of physically based phenomenological models. *Composites Science and Technology*, 58(7):1045–1067, 1998.
- [112] Y. Qiu, M. A. Crisfield, and G. Alfano. An interface element formulation for the simulation of delamination with buckling. *Engineering Fracture Mechanics*, 68(16):1755–1776, 2001.

- [113] Timon Rabczuk, Goangseup Zi, Axel Gerstenberger, and Wolfgang A. Wall. A new crack tip element for the phantom-node method with arbitrary cohesive cracks. *International Journal for Numerical Methods in Engineering*, 75(5):577–599, 2008.
- [114] James R. Reeder. A bilinear failure criterion for mixed-mode delamination. *Composite Materials: Testing and Design (Eleventh Volume), ASTM STP 1206*, pages 303–322, 1993.
- [115] James R. Reeder, Kevin Demarco, and Karen S. Whitley. The use of doubler reinforcement in delamination toughness testing. *Composites Part A: Applied Science and Manufacturing*, 35(11):1337 – 1344, 2004.
- [116] J. J. C. Remmers, R. de Borst, and A. Needleman. A cohesive segments method for the simulation of crack growth. *Computational Mechanics*, 31(1-2):69–77, 2003.
- [117] Joris J. C. Remmers, René de Borst, and Alan Needleman. The simulation of dynamic crack propagation using the cohesive segments method. *Journal of the Mechanics and Physics of Solids*, 56(1):70–92, 2008.
- [118] Casey L. Richardson, Jan Hegemann, Eftychios Sifakis, Jeffrey Hellrung, and Joseph M. Teran. An XFEM method for modeling geometrically elaborate crack propagation in brittle materials. *International Journal for Numerical Methods in Engineering*, 88(10):1042–1065, 2011.
- [119] M. Ridha, V. B. C. Tan, and T. E. Tay. Traction-separation laws for progressive failure of bonded scarf repair of composite panel. *Composite Structures*, 93(4):1239–1245, 2011.
- [120] J. G. Rots, P. Nauta, G. M. A. Kuster, and J. Blaauwendraad. Smeared crack approach and fracture localization in concrete. *Heron*, 30, 1985.

- [121] E. F. Rybicki and M. F. Kanninen. A finite element calculation of stress intensity factors by a modified crack closure integral. *Engineering Fracture Mechanics*, 9(9):931–938, 1977.
- [122] Bernhard A. Schrefler, Stefano Secchi, and Luciano Simoni. On adaptive refinement techniques in multi-field problems including cohesive fracture. *Computer Methods in Applied Mechanics and Engineering*, 195(4):444–461, 2006.
- [123] A. R. Shahani and M. R. Amini Fasakhodi. Finite element analysis of dynamic crack propagation using remeshing technique. *Materials & Design*, 30(4):1032 – 1041, 2009.
- [124] Mark S. Shephard, Nabil A. B. Yehia, Gary S. Burd, and Theodore J. Weidner. Automatic crack propagation tracking. *Computers & Structures*, 20(1-3):211 – 223, 1985.
- [125] P. D. Soden, A. S. Kaddour, and M. J. Hinton. Recommendations for designers and researchers resulting from the world-wide failure exercise. *Composites Science and Technology*, 64(3-4):589–604, 2004.
- [126] Jeong-Hoon Song and Ted Belytschko. Dynamic fracture of shells subjected to impulsive loads. *Journal of Applied Mechanics-Transactions of the ASME*, 76(5), 2009.
- [127] Jeong-Hoon Song, Pedro M. A. Areias, and Ted Belytschko. A method for dynamic crack and shear band propagation with phantom nodes. *International Journal for Numerical Methods in Engineering*, 67(6):868–893, 2006.
- [128] Kyongchan Song, Yingyong Li, and Cheryl A Rose. Continuum damage mechanics models for the analysis of progressive failure in open-hole tension laminates. *AIAA*, 1861:1–18, 2011.
- [129] N. Sukumar, Z. Y. Huang, J. H. Prevost, and Z. Suo. Partition of unity enrichment for bimaterial interface cracks. *International Journal for Numerical Methods in Engineering*, 59(8):1075–1102, 2004.

- [130] X. S. Sun, V. B. C. Tan, G. Liu, and T. E. Tay. An enriched element-failure method (REFM) for delamination analysis of composite structures. *International Journal for Numerical Methods in Engineering*, 79(6):639–666, 2009.
- [131] L. S. Sutherland, R. A. Shenoi, and S. M. Lewis. Size and scale effects in composites: I. literature review. *Composites Science and Technology*, 59(2):209–220, 1999.
- [132] L. S. Sutherland, R. A. Shenoi, and S. M. Lewis. Size and scale effects in composites: II. unidirectional laminates. *Composites Science and Technology*, 59(2):221–233, 1999.
- [133] L. S. Sutherland, R. A. Shenoi, and S. M. Lewis. Size and scale effects in composites: III. woven-roving laminates. *Composites Science and Technology*, 59(2):235–251, 1999.
- [134] D. V. Swenson and A. R. Ingraffea. Modeling mixed-mode dynamic crack propagation using finite elements: theory and applications. *Computational Mechanics*, 3(6):381–397, 1988.
- [135] M. J. Swindeman, E. V. Iarve, R. A. Brockman, D. H. Mollenhauer, and S. R. Hallett. Strength prediction in open hole composite laminates by using discrete damage modeling. *AIAA Journal*, 51:936–945, 2013.
- [136] H. Tada, P. C. Paris, and G. R. Irwin. *The stress analysis of cracks handbook*. Second edition, Paris Productions Incorporated (and Del Research Corporation), 226 Woodbourne Dr. St. Louis, Missouri. 1985.
- [137] Nobuo Takeda and Shinji Ogiwara. Initiation and growth of delamination from the tips of transverse cracks in CFRP cross-ply laminates. *Composites science and technology*, 52(3):309–318, 1994.
- [138] J. X. Tao and C. T. Sun. Influence of ply orientation on delamination in composite laminates. *Journal of Composite Materials*, 32(21):1933–1947, 1998.

- [139] T. E. Tay. Characterization and analysis of delamination fracture in composites: An overview of developments from 1990 to 2001. *Applied Mechanics Reviews*, 56(1):1–32, 2003.
- [140] Stephen W. Tsai and Edward M. Wu. A general theory of strength for anisotropic materials. *Journal of Composite Materials*, 5(1):58–80, 1971.
- [141] A. Turon, C. G. Dávila, P. P. Camanho, and J. Costa. An engineering solution for mesh size effects in the simulation of delamination using cohesive zone models. *Engineering Fracture Mechanics*, 74(10):1665–1682, 2007.
- [142] A. Turon, P. P. Camanho, J. Costa, and J. Renart. Accurate simulation of delamination growth under mixed-mode loading using cohesive elements: Definition of interlaminar strengths and elastic stiffness. *Composite Structures*, 92(8):1857–1864, 2010.
- [143] F. P. van der Meer and L. J. Sluys. A phantom node formulation with mixed mode cohesive law for splitting in laminates. *International Journal of Fracture*, 158(2):107–124, 2009.
- [144] F. P. van der Meer and L. J. Sluys. Continuum models for the analysis of progressive failure in composite laminates. *Journal of Composite Materials*, 43(20):2131–2156, 2009.
- [145] F. P. van der Meer and L. J. Sluys. Mesh-independent modeling of both distributed and discrete matrix cracking in interaction with delamination in composites. *Engineering Fracture Mechanics*, 77(4):719–735, 2010.
- [146] F. P. van der Meer, L. J. Sluys, S. R. Hallett, and M. R. Wisnom. Computational modeling of complex failure mechanisms in laminates. *Journal of Composite Materials*, 46(5):603–623, 2012.
- [147] Frans P. van der Meer and Carlos G. Dávila. Cohesive modeling of transverse cracking in laminates under in-plane loading with a single layer of elements per ply. *International Journal of Solids and Structures*, 50(20-21):3308 – 3318, 2013.

- [148] G. M. Vyas, S. T. Pinho, and P. Robinson. Constitutive modelling of fibre-reinforced composites with unidirectional plies using a plasticity-based approach. *Composites Science and Technology*, 71(8):1068–1074, 2011.
- [149] A. S. D. Wang. Fracture mechanics of sublaminar cracks in composite materials. *Composites Technology Review*, 6(2):45–62, 1984.
- [150] W. Weibull. A statistical distribution function of wide applicability. *Journal of Applied Mechanics-Transactions of the ASME*, 18:293–297, 1951.
- [151] S. Weihe, B. Kroplin, and R. de Borst. Classification of smeared crack models based on material and structural properties. *International Journal of Solids and Structures*, 35(12):1289–1308, 1998.
- [152] M. R. Wisnom. Size effects in the testing of fibre-composite materials. *Composites Science and Technology*, 59(13):1937–1957, 1999.
- [153] M. R. Wisnom. Modelling discrete failures in composites with interface elements. *Composites Part A-Applied Science and Manufacturing*, 41(7):795–805, 2010.
- [154] M. R. Wisnom and S. R. Hallett. The role of delamination in strength, failure mechanism and hole size effect in open hole tensile tests on quasi-isotropic laminates. *Composites Part A-Applied Science and Manufacturing*, 40(4):335–342, 2009.
- [155] M. R. Wisnom, B. Khan, and S. R. Hallett. Size effects in unnotched tensile strength of unidirectional and quasi-isotropic carbon/epoxy composites. *Composite Structures*, 84(1):21–28, 2008.
- [156] M. R. Wisnom, S. R. Hallett, and C. Soutis. Scaling effects in notched composites. *Journal of Composite Materials*, 44(2):195–210, 2010.
- [157] Q. D. Yang and B. Cox. Cohesive models for damage evolution in laminated composites. *International Journal of Fracture*, 133(2):107–137, 2005.

- [158] Q. D. Yang, X. J. Fang, J. X. Shi, and J. Lua. An improved cohesive element for shell delamination analyses. *International Journal for Numerical Methods in Engineering*, 83(5):611–641, 2010.
- [159] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals, 6th Edition*. Elsevier Science, 2005.

Appendices

Appendix A

Phantom Node Method Evaluation and Extension

A.1 Literature Review

The Phantom Node Method (PNM) was firstly proposed by Hansbo and Hansbo [57] and later promoted and implemented by Belytschko and co-workers [5, 6, 127]. It has since been further extended and implemented in [33, 46, 49, 50, 83, 89–91, 113, 118, 126, 143, 146, 147]. In the PNM, a material domain with a potential internal discontinuity can be modelled by one element with two pairs of nodes, namely the real nodes and the phantom nodes. When the stress state of the element exceeds the strength limit of the material within the element, a discontinuity is modelled by forming two superposing elements with the help of phantom nodes (Figure A.1); each of the two elements contains only part of the material domain. Since all nodes (both real and phantom) of the elements are defined before analysis, the location of the discontinuity does not need to be known a priori. When modelling a strong discontinuity within an element, the PNM has been proven to be equivalent to the eXtended Finite Element Method (XFEM) with only the Heaviside enrichment [5, 127]. Both methods essentially use extra Degrees of Freedom (DoF) to interpolate the new crack surfaces. The difference between the two is that the PNM keeps the nodal DoF as displacement DoF and stores the extra DoF needed as the displacement DoF of the phantom nodes, while XFEM keeps the number of nodes constant and

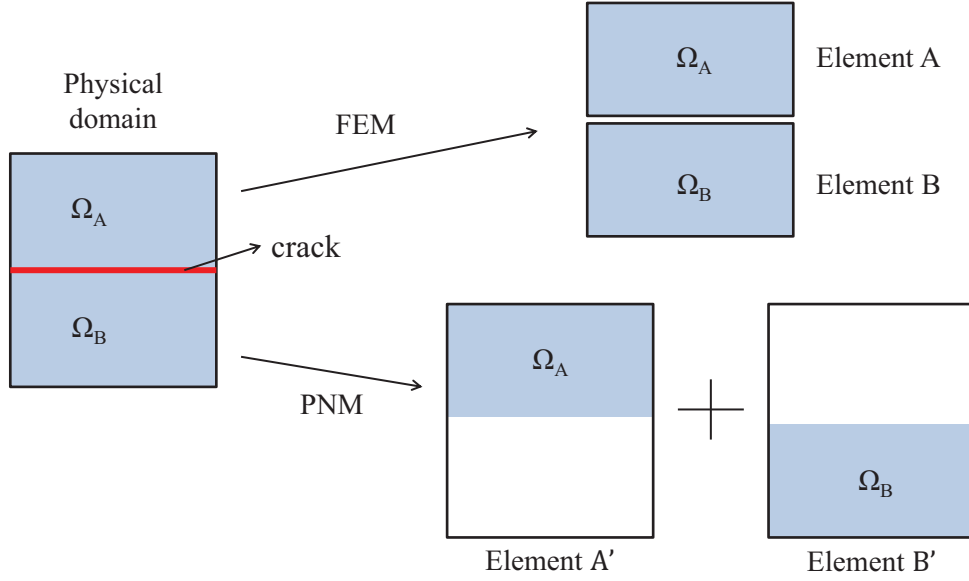


Figure A.1: Modelling of a crack by the FEM and the PNM

stores the extra DoF needed as enriched DoF at each node. The advantage of the former is that it is easier to be implemented in existing FEM programmes as each node has only the standard displacement DoF and only the standard FEM shape functions are used for interpolation.

Currently, the PNM has only been used to model one crack within an element. Although some concept of modelling multiple cracks has been presented in [83], the detailed formulation and implementation remain absent in the literature. The PNM is investigated and compared with the standard FEM in Section A.2 and some extension is proposed to the PNM to more efficiently model multiple cohesive cracks within one element in Section A.3, followed by applications in Section A.4 and discussions in Section A.5. Finally, the conclusion is given in Section A.6.

A.2 Comparison of FEM and PNM

In this section, a comparison between the solution procedures of the FEM and the PNM is presented. Taking the elements in Figure A.2 as an example, the displacement solutions over Ω by both the FEM and the PNM elements will be presented.

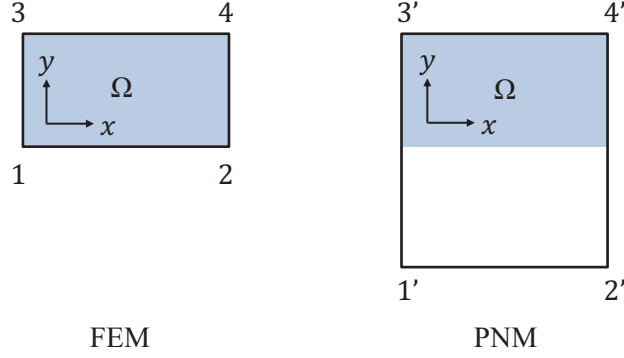


Figure A.2: A FEM element vs. a PNM element over the same material domain

The static equilibrium of a body with volume Ω under body forces with density $\mathbf{f}$ (acting on Ω) and traction $\mathbf{t}$ acting on the boundary Γ_Ω can be expressed in the weak form as:

$$\int_{\Omega} \boldsymbol{\epsilon}^T(\mathbf{v}) \boldsymbol{\sigma}(\mathbf{u}) \, d\Omega = \int_{\Omega} \mathbf{v}^T \mathbf{f} \, d\Omega + \int_{\Gamma_\Omega} \mathbf{v}^T \mathbf{t} \, d\Gamma_\Omega \quad (\text{A.1})$$

where $\mathbf{u}$ is the displacement vector; $\mathbf{v}$ is the test function; $\boldsymbol{\epsilon}$ is the strain tensor, related to $\mathbf{u}$ through the differential operator relative to Cartesian coordinates $\mathcal{L}_{\mathbf{x}}$ as $\boldsymbol{\epsilon} = \mathcal{L}_{\mathbf{x}}(\mathbf{u})$; and $\boldsymbol{\sigma}$ is the stress tensor, related to the strains through Hook's law as $\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon}$, with $\mathbf{D}$ being the constitutive tensor.

The displacement approximation by the FEM in Ω is:

$$\mathbf{u}(\mathbf{x}) = \mathbf{N}(\mathbf{x}) \mathbf{q}, \quad (\text{A.2})$$

where $\mathbf{q}$ and $\mathbf{N}$ are respectively the nodal displacement vector and the shape function matrix (in physical coordinates) of the FEM element. They are typically of the following form:

$$\mathbf{q} = \{q_x^1, q_y^1, \dots, q_x^4, q_y^4\}^T \quad (\text{A.3})$$

and

$$\mathbf{N}(\mathbf{x}) = \begin{bmatrix} N_1(\mathbf{x}) & 0 & \dots & N_4(\mathbf{x}) & 0 \\ 0 & N_1(\mathbf{x}) & \dots & 0 & N_4(\mathbf{x}) \end{bmatrix} \quad (\text{A.4})$$

where q_i^j is the i th degree of freedom of node j and N_j is the shape function associated with node j , $j = 1, 2, 3, 4$.

Under the assumption of infinitesimal deformation, the strain of the standard element can be calculated as:

$$\boldsymbol{\epsilon}(\mathbf{x}) = \mathcal{L}_{\mathbf{x}}(\mathbf{u}(\mathbf{x})) = \mathcal{L}_{\mathbf{x}}(\mathbf{N}(\mathbf{x})) \mathbf{q} = \mathbf{B}(\mathbf{x}) \mathbf{q}, \quad (\text{A.5})$$

where

$$\mathbf{B}(\mathbf{x}) = \mathcal{L}_{\mathbf{x}}(\mathbf{N}(\mathbf{x})), \quad (\text{A.6})$$

leading to the stiffness matrix:

$$\mathbf{K} = \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega, \quad (\text{A.7})$$

and to the vector of nodal forces:

$$\mathbf{Q} = \int_{\Omega} \mathbf{N}^T \mathbf{f} d\Omega + \int_{\Gamma_{\Omega}} \mathbf{N}^T \mathbf{t} d\Gamma_{\Omega}, \quad (\text{A.8})$$

and finally to the FEM equilibrium equation:

$$\mathbf{K} \mathbf{q} = \mathbf{Q}. \quad (\text{A.9})$$

The solution procedures of the PNM are the same as those of the FEM, except that another set of nodal displacement vector and shape functions are used. The PNM displacement approximation over the domain Ω , $\mathbf{u}'$, can be expressed as:

$$\mathbf{u}'(\mathbf{x}) = \mathbf{N}'(\mathbf{x})|_{\mathbf{x} \in \Omega} \mathbf{q}', \quad (\text{A.10})$$

where $\mathbf{q}'$ and $\mathbf{N}'$ are respectively the nodal displacement vector and the shape function matrix (in physical coordinates) of the PNM element.

It is easy to see that the shape functions of the FEM element in Fig-

ure A.2 are:

$$N_1(x, y) = \frac{(x_2 - x)(y_3 - y)}{(x_2 - x_1)(y_3 - y_1)}, \quad (\text{A.11})$$

$$N_2(x, y) = \frac{(x_1 - x)(y_4 - y)}{(x_1 - x_2)(y_4 - y_2)}, \quad (\text{A.12})$$

$$N_3(x, y) = \frac{(x_4 - x)(y_1 - y)}{(x_4 - x_3)(y_1 - y_3)}, \quad (\text{A.13})$$

$$N_4(x, y) = \frac{(x_3 - x)(y_2 - y)}{(x_3 - x_4)(y_2 - y_4)}, \quad (\text{A.14})$$

and those of the PNM element in Figure A.2 are

$$N'_1(x, y) = \frac{(x_{2'} - x)(y_{3'} - y)}{(x_{2'} - x_{1'})(y_{3'} - y_{1'})}, \quad (\text{A.15})$$

$$N'_2(x, y) = \frac{(x_{1'} - x)(y_{4'} - y)}{(x_{1'} - x_{2'})(y_{4'} - y_{2'})}, \quad (\text{A.16})$$

$$N'_3(x, y) = \frac{(x_{4'} - x)(y_{1'} - y)}{(x_{4'} - x_{3'})(y_{1'} - y_{3'})}, \quad (\text{A.17})$$

$$N'_4(x, y) = \frac{(x_{3'} - x)(y_{2'} - y)}{(x_{3'} - x_{4'})(y_{2'} - y_{4'})}. \quad (\text{A.18})$$

It can be seen that the above FEM and the PNM shape functions are polynomials of the same order over Ω and they both form a basis. Therefore, each PNM shape function (restricted over Ω) can be expressed as a unique linear combination of the FEM shape functions, and vice versa:

$$N'_l(\mathbf{x})|_{\mathbf{x} \in \Omega} = \sum_{j=1}^4 a_j^l N_j(\mathbf{x}), \quad l = 1, 2, 3, 4, \quad (\text{A.19})$$

where

$$a_j^l = N'_l(\mathbf{x}_j). \quad (\text{A.20})$$

In other words, there exist a constant invertible map $\mathbf{T}$ such that:

$$\mathbf{N}'(\mathbf{x})|_{\mathbf{x} \in \Omega} = \mathbf{N}(\mathbf{x}) \mathbf{T}, \quad (\text{A.21})$$

where

$$\mathbf{T} = \begin{bmatrix} \mathbf{A}_1^1 & \mathbf{A}_1^2 & \dots & \mathbf{A}_1^4 \\ \mathbf{A}_2^1 & \mathbf{A}_2^2 & \dots & \mathbf{A}_2^4 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_4^1 & \mathbf{A}_4^2 & \dots & \mathbf{A}_4^4 \end{bmatrix}, \quad (\text{A.22})$$

and

$$\mathbf{A}_i^j = \begin{bmatrix} a_i^j & 0 \\ 0 & a_i^j \end{bmatrix}. \quad (\text{A.23})$$

Substituting A.21 into A.10, we can get:

$$\mathbf{u}'(\mathbf{x}) = \mathbf{N}(\mathbf{x}) \mathbf{T} \mathbf{q}', \quad (\text{A.24})$$

Proceeding to strain approximation, we can get:

$$\boldsymbol{\epsilon}(\mathbf{x}) = \mathcal{L}_{\mathbf{x}}(\mathbf{u}'(\mathbf{x})) = \mathcal{L}_{\mathbf{x}}(\mathbf{N}'(\mathbf{x})) \mathbf{q}' = \mathbf{B}'(\mathbf{x}) \mathbf{q}', \quad (\text{A.25})$$

where

$$\mathbf{B}'(\mathbf{x}) = \mathcal{L}_{\mathbf{x}}(\mathbf{N}'(\mathbf{x})) = \mathcal{L}_{\mathbf{x}}(\mathbf{N}(\mathbf{x})) \mathbf{T} = \mathbf{B}(\mathbf{x}) \mathbf{T}, \quad (\text{A.26})$$

leading to the PNM stiffness matrix:

$$\mathbf{K}' = \int_{\Omega} \mathbf{B}'^T \mathbf{D} \mathbf{B}' d\Omega = \mathbf{T}^T \mathbf{K} \mathbf{T}, \quad (\text{A.27})$$

and to the PNM nodal force vector:

$$\mathbf{Q}' = \int_{\Omega} \mathbf{N}'^T \mathbf{f} d\Omega + \int_{\Gamma_{\Omega}} \mathbf{N}'^T \mathbf{t} d\Gamma_{\Omega} = \mathbf{T}^T \mathbf{Q}, \quad (\text{A.28})$$

and finally to the PNM equilibrium equation:

$$\mathbf{K}' \mathbf{q}' = \mathbf{Q}', \quad (\text{A.29})$$

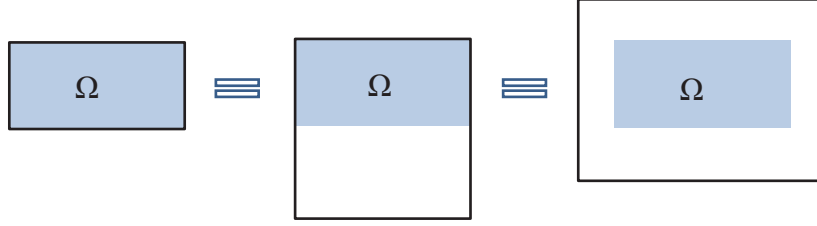


Figure A.3: Complete distinction between the numerical domain and the material domain

or

$$\mathbf{T}^T \mathbf{K} \mathbf{T} \mathbf{q}' = \mathbf{T}^T \mathbf{Q}, \quad (\text{A.30})$$

Comparing (A.30) with A.9 and making use of the fact that $\mathbf{T}$ is invertible and $\mathbf{K}$ is symmetric positive definite, we can obtain the following relationship for nodal displacement vectors:

$$\mathbf{T} \mathbf{q}' = \mathbf{q}, \quad (\text{A.31})$$

and for displacement solutions:

$$\mathbf{u}'(\mathbf{x}) = \mathbf{N}'(\mathbf{x})|_{\mathbf{x} \in \Omega} \mathbf{q}' = \mathbf{N}(\mathbf{x}) \mathbf{T} \mathbf{q}' = \mathbf{N}(\mathbf{x}) \mathbf{q} = \mathbf{u}(\mathbf{x}). \quad (\text{A.32})$$

Therefore, the displacement approximations over Ω by the FEM element and the FNM element in Figure A.2 are identical.

A.3 Extension

A.3.1 Separation of numerical domain and material domain

Looking back at the derivations in Section A.2, it can be observed that as long as the mapping in (A.21) holds, the same conclusions (A.31) and A.32 can be reached. Section A.2 has shown that the mapping in (A.21) exists when the shape functions of both the FEM element and the PNM element are well-defined in physical coordinates and they both form a basis for same-ordered polynomials over the material domain. This is the case

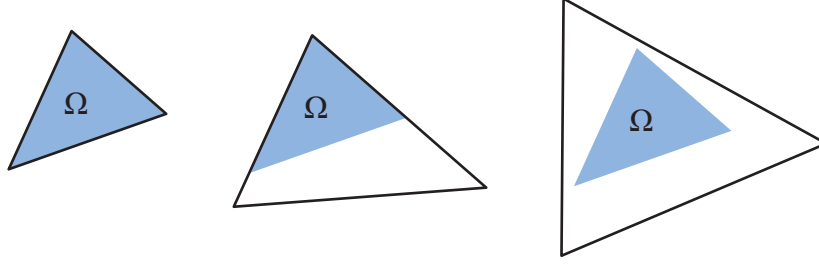


Figure A.4: Linear triangular FNM elements are equivalent to a FEM triangular element.

when the material domain (hence the FEM element domain) is rectangular and the PNM element is also rectangular with edges parallel to those of the material domain (Figure A.2), as long as the numerical domain of the PNM element contains the material domain and regardless of the positions of the phantom nodes. Therefore, the PNM rectangular element in Figure A.2 can have all its nodes as phantom nodes (completely detached from the material domain that it interpolates on) and it still gives the same solution (Figure A.3).

It can be proven that the mapping in (A.21) also exists between a linear triangular FEM element and a linear triangular PNM element on the same material domain, regardless of the orientation of the edges and the positions of the nodes, as long as the numerical domain of the PNM element contains the material domain (Figure A.4). This is because the shape functions of linear triangular elements are well-defined in physical coordinates and they form basis for first-order polynomial interpolations over their domains. Over the same material domain, there exists a constant invertible map between two sets of basis functions, thus the relationship in (A.21) holds between any of the two elements in Figure A.4.

The mapping described by (A.21) also holds between a linear FEM cohesive element (Figure A.5 (a)) and linear rectangular PNM cohesive elements with a boundary parallel to the cohesive surface (Figure A.5 (b) and (c)), as the restrictions of the PNM element shape functions along the cohesive surface are linear and they are equivalent to the FEM cohesive element shape functions.

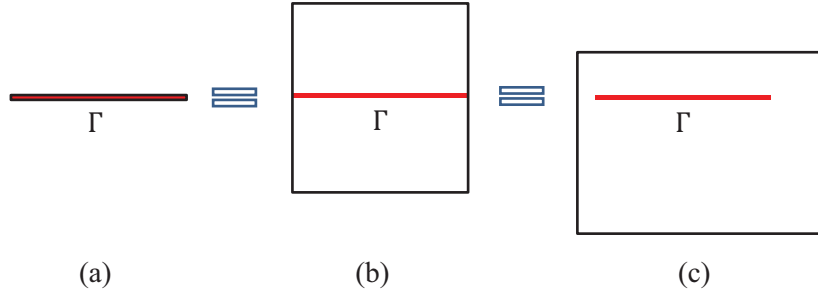


Figure A.5: A FEM cohesive element (a) is equivalent to linear rectangular PNM cohesive elements (b, c), when a boundary of the PNM cohesive element is parallel to the cohesive interface. All the nodes of the PNM cohesive elements are phantom nodes.

A.3.2 Modelling cohesive cracks by PNM cohesive elements

The above insights lead to the idea of an extension to the current PNM, where cohesive cracks can be modelled more efficiently within a PNM element, where all its nodes are phantom nodes. When modelling one cohesive crack within a rectangular PNM element with an extra pair of phantom nodes and the cohesive surface is parallel to an edge of the element (Figure A.6 (a)), the element can be broken into three parts with the cohesive crack modelled by a PNM cohesive element (Figure A.6 (b)). The cohesive forces are represented as internal forces of the PNM cohesive element, which contribute directly to the internal energy of the PNM element, rather than to the surface tractions on the boundaries of Ω_A and Ω_B (as done in previous works on the PNM [83, 127, 143]). In this way, the need of modifying the equilibrium equations of the two material domains Ω_A and Ω_B , and solving the coupled system of equations for the two domains, is eliminated; the system equation of the PNM element after introducing the cohesive crack can be obtained by the standard FEM assembly procedure for a mesh of three elements. Similarly, multiple cohesive cracks can be modelled within a PNM element, using a suitable number of phantom nodes to form multiple PNM cohesive elements (Figure A.6).

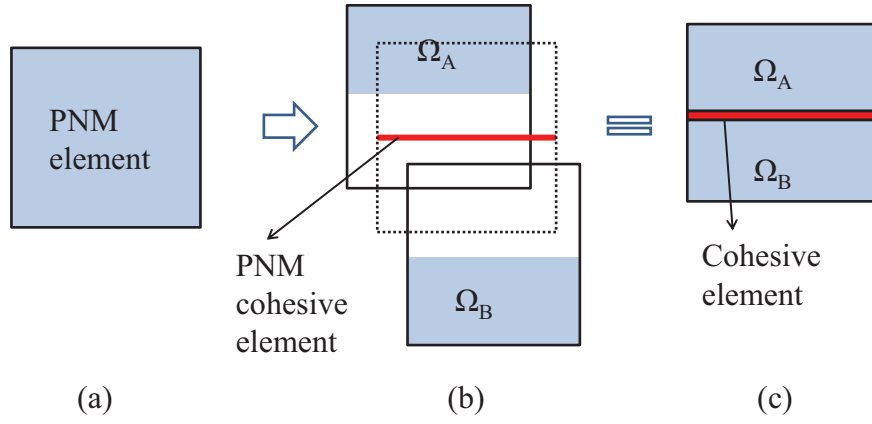


Figure A.6: Modelling of a cohesive crack in a PNM element; the phantom nodes for the PNM cohesive element are actually not needed as they can be fully interpolated by the nodes of the other two elements. The PNM element in (a) is equivalent to a FEM mesh of two rectangular elements and one cohesive element in (c).

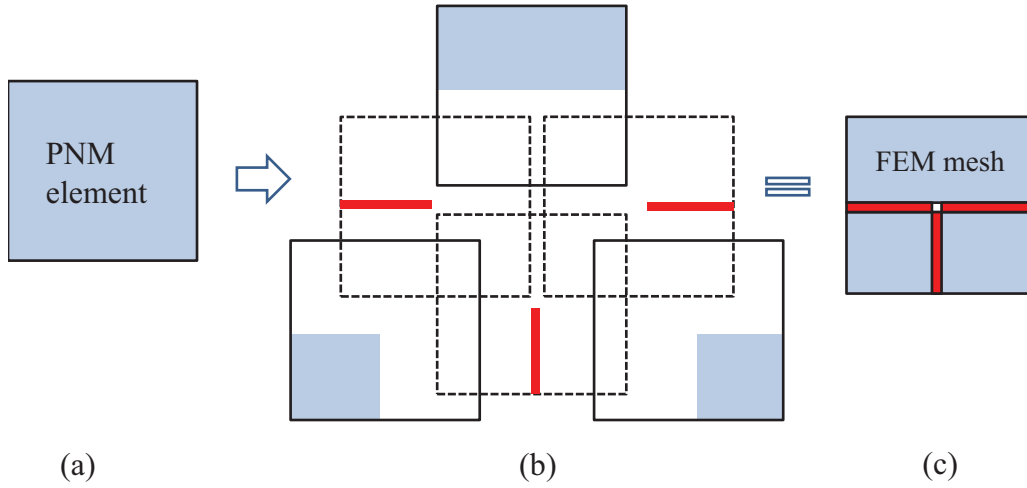


Figure A.7: Modelling of multiple cohesive cracks in a PNM element; the phantom nodes for the PNM cohesive elements can be fully interpolated by the nodes of the rectangular elements. The PNM element in (a) is equivalent to a FEM mesh of three cohesive elements and three rectangular elements in (c).

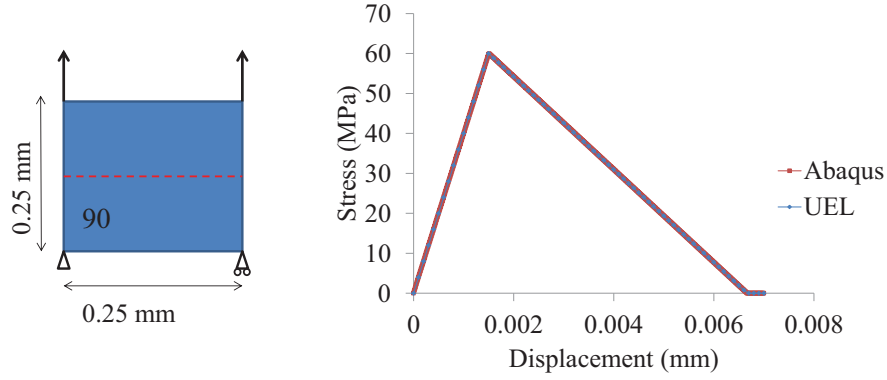
Table A.1: Mechanical properties of a glass/epoxy composite (assumed)

Property	Value
Longitudinal Young's modulus E_1 (GPa)	50.0
Transverse Young's modulus E_2 (GPa)	10.0
Poisson's ratio ν	0.3
Shear modulus G_{12} (GPa)	5.0
Transverse tensile strength Y_t (MPa)	60
Mode I critical energy release rate G_{Ic} (kJ/m ²)	0.2
Shear strength S (MPa)	90
Mode II critical energy release rate G_{IIc} (kJ/m ²)	1.0

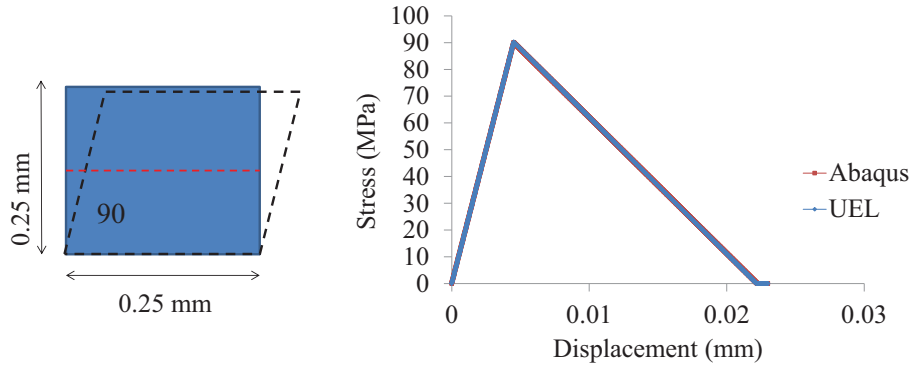
A.4 Applications

A.4.1 Modelling of single cohesive crack

The extended PNM is firstly employed in the modelling of a single cohesive crack within an element. A standard FEM mesh as shown in Figure A.6 (c) with Ω_A and Ω_B modelled by two quadrilateral elements and the cohesive crack modelled by a cohesive element is built in Abaqus. A PNM element as shown in Figure A.6 (a) and (b) is written in an Abaqus User Element (UEL) subroutine. Based on the conclusion in Section A.3, the PNM element should behave exactly like the standard FEM mesh. A simple tension loading and a simple shear loading are performed on both the Abaqus FEM mesh and the UEL PNM element, assuming that the material is a 90 degree lamina with a cohesive crack appearing in the middle of the element upon failure initiation. The material is assumed to be a typical glass-epoxy composite and the properties used are summarised in Table A.1. The stress-displacement curves by the two methods are compared in Figure A.8a and Figure A.8b. It can be seen that the two methods give identical response



(a) Simple tension loading



(b) Simple shear loading

Figure A.8: Single cohesive crack modelling by the PNM

which supports the conclusion in Section A.3, and that the area under the two curves correspond to the fracture toughness of the respective fracture modes which indicate that correct energy dissipations have been predicted on both cases.

A.4.2 Modelling of multiple crack interactions in a composite laminate

A simple uniform tensile loading problem of a section of a cross-ply laminate is studied in this section (Figure A.9). The material is assumed to be the same glass-epoxy composite as that of the previous section whose properties are shown in Table A.1. It has been shown in [49] that for such a problem,

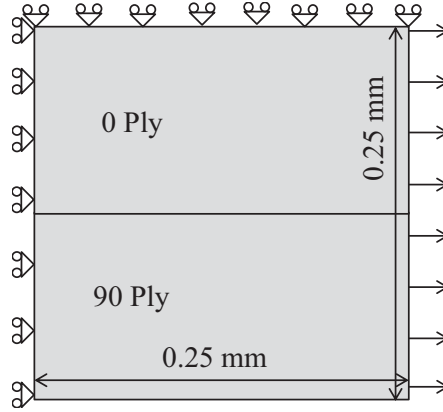


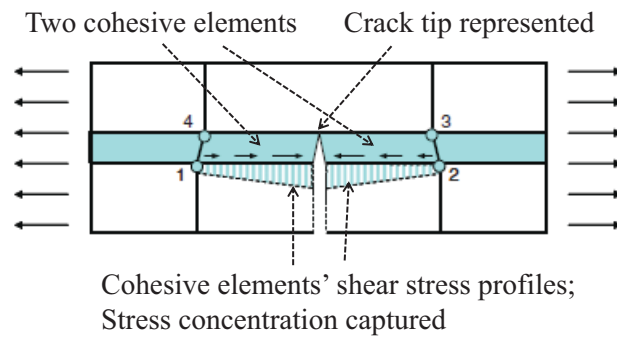
Figure A.9: Uniform tensile loading of a small section of a cross-ply laminate

it is important for the numerical model to represent the 90-ply matrix crack tip at the interface to be able to capture the correct stress concentration at that location, as illustrated in Figure A.10.

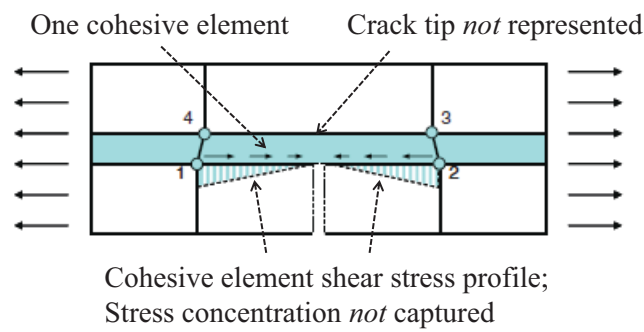
The extended PNM is employed to develop a complex element with multiple pairs of phantom nodes to model multiple discontinuities of a cross-ply $[90/0]_S$ composite laminate, as shown in Figure A.11. The entire laminate across the thickness is modelled within such an element. The PNM behaves in such a way that:

- initially, four phantom nodes of the PNM element are already active and the PNM element is composed of two PNM rectangular elements (for the two plies) and one PNM cohesive element (for the interface); they are equivalent to a FEM mesh of two quadrilateral elements and one cohesive element (Figure A.11a).
- After the stress state within the 90-ply PNM element fulfils the failure criterion, more phantom nodes are activated and the 90-ply PNM element is broken into two rectangular PNM elements and one PNM cohesive element; the PNM cohesive element at the interface also breaks into two new PNM cohesive elements with respect to the position of the matrix crack tip (Figure A.11b).

The PNM element is written in an Abaqus UEL subroutine. Post-processing is performed by a Matlab program.

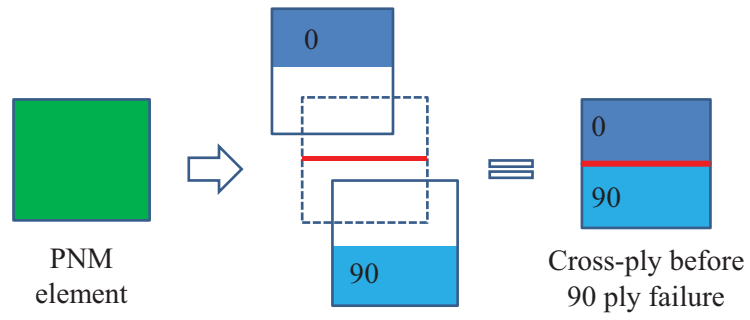


(a) Laminate interface modelled by two cohesive elements with matrix crack tip embedded

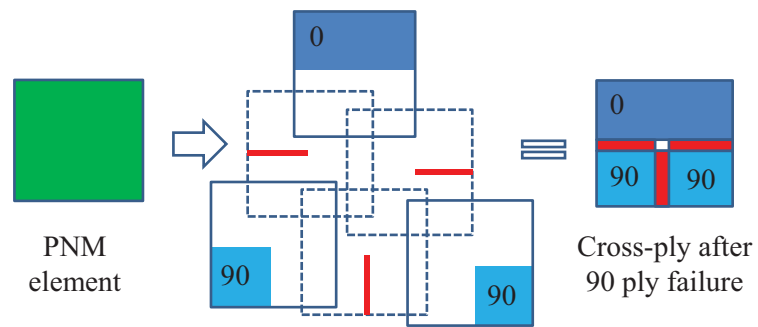


(b) Laminate interface modelled by one cohesive element without embedding matrix crack tip

Figure A.10: Importance of matrix crack tip representation on the laminate interface [49]



(a) Before 90-ply matrix failure



(b) After 90-ply matrix failure

Figure A.11: A PNM element representing a cross-ply laminate

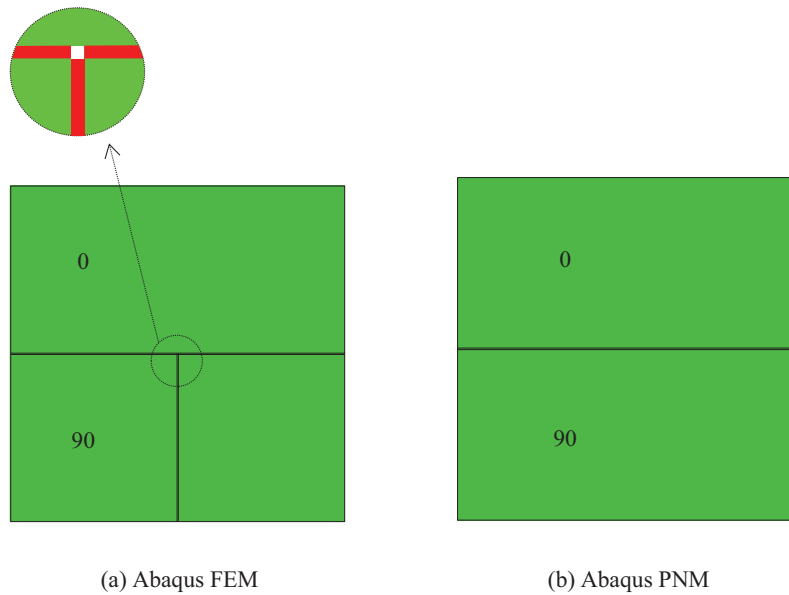


Figure A.12: Two other models for the modelling of a cross-ply section

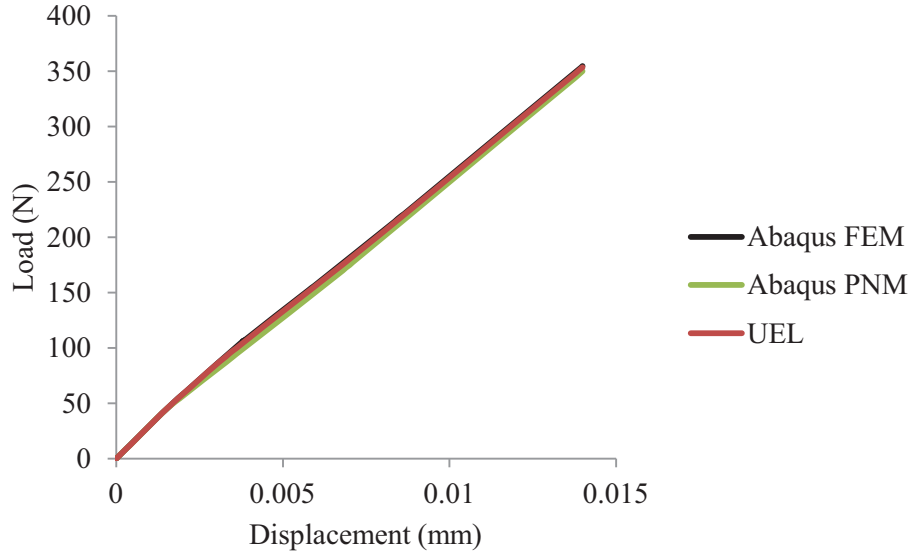
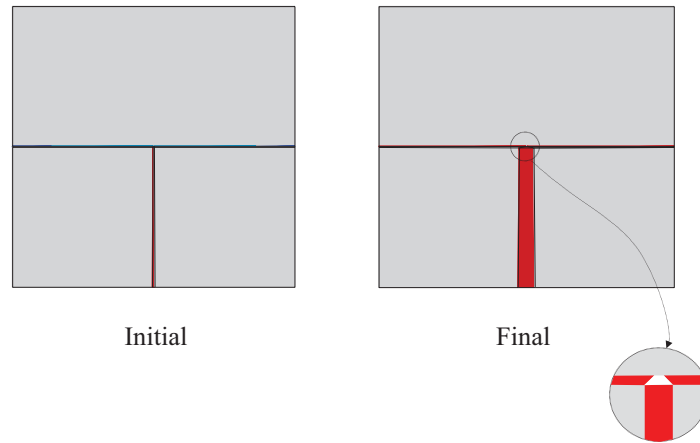


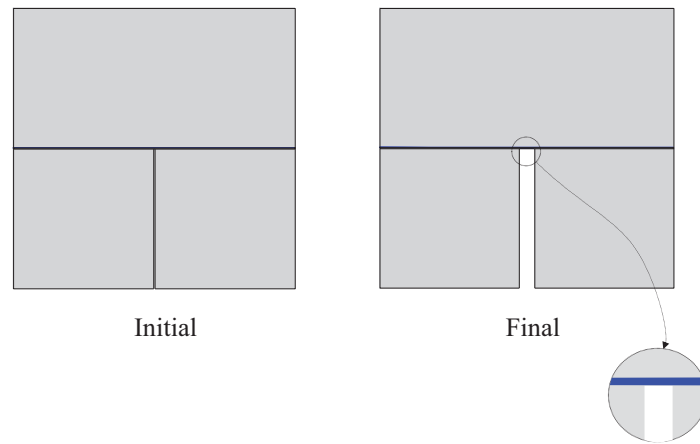
Figure A.13: Comparison of load-displacement curves of Abaqus FEM result, Abaqus PNM result and the extended PNM element result (UEL)

The PNM element in Figure A.11 is applied to model the problem in Figure A.9 with one such element. In order to compare the performance of the proposed PNM element against existing methods, two other models as shown in Figure A.12 are built and applied on this problem. Figure A.12 (a) is the Abaqus FEM model which comprises of a standard FEM mesh of three linear quadrilateral elements and three linear cohesive elements; the delamination, matrix crack and the matrix crack tip at the interface are all explicitly represented. Figure A.12 (b) is the Abaqus PNM model composed of a mesh of three elements; the 0-ply is a linear quadrilateral element, the interface is a linear cohesive element and the 90-ply element is a PNM element (Abaqus XFEM element). The matrix crack tip at the interface cannot be considered by the Abaqus PNM model as the topology of cohesive element at the interface cannot be modified during analysis.

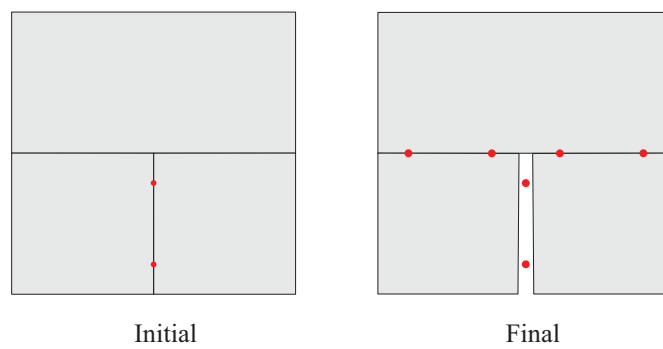
The predictions of the laminate load-displacement curve by the three models do not show any significant difference and they are summarised in Figure A.13. The predictions of matrix crack and delamination by the three methods are compared in Figure A.14. For the PNM element, failure infor-



(a) Abaqus FEM model failure patterns



(b) Abaqus PNM model failure patterns



(c) Extended PNM (UEL) failure patterns

Figure A.14: Comparison of failure patterns by three different models; blue color indicates intact and red indicates failure for (a) and (b); red dots in (c) indicate failure at the Gauss integration points.

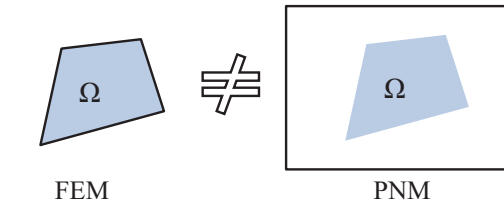
mation is available only at the integration points of the cohesive elements; a red dot appears when there is failure. Both the deformation and the failure sequence predictions by the FEM mesh and the extended PNM are in good agreement with each other; they both predict an eventual formation of a ‘T’ shaped crack network. However, the Abaqus PNM approach cannot predict the initiation of delamination throughout the analysis, which is the expected result from the analysis in Figure A.10.

A.5 Discussions

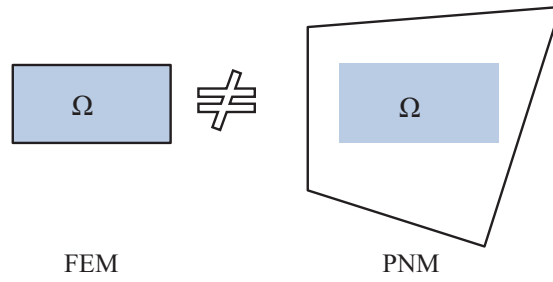
The above results are limited to the case of PNM elements’ shape being rectangular and the cohesive crack surface being straight and parallel to an edge of the PNM element. Under generic conditions, the PNM elements’ shape and/or the material domain shape may be distorted, resulting in general quadrilateral elements whose shape functions may not be defined in physical coordinates (Figure A.15 (a) and (b)). The cohesive crack may not be parallel to the edges of the PNM element, resulting in higher-order polynomial interpolation along the crack surface (Figure A.15 (c)). In such situations, the mapping in (A.21) does not exist between the PNM element and the FEM element defined over the material domain or the cohesive crack surface.

In generic situations, the idea of using PNM cohesive elements to represent cohesive cracks within a PNM element can still be employed; but the solution by the PNM element are generally different from that by a FEM mesh. Figure A.16 illustrates the solution procedures of both a FEM mesh and a PNM element on the modelling of a straight cohesive crack in a quadrilateral material domain. It can be seen that when mapping the element from physical coordinates to natural coordinates for numerical integration, the PNM would transform a straight boundary into a curved one, which would require more complex procedures for an accurate integration (Figure A.16 (b)).

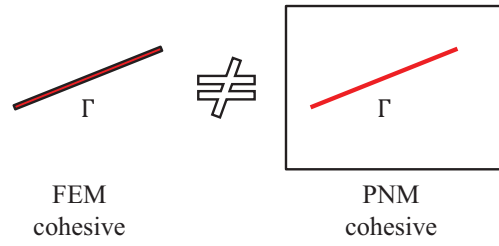
The work in this chapter has evolved into the Floating Node Method (FNM) which is presented in Chapter 4. The FNM removes the complexity of integration in the PNM, while retains the flexibility of modelling multiple cohesive cracks during analysis. The FNM solutions are the same as those



(a) Arbitrary quadrilateral material domain

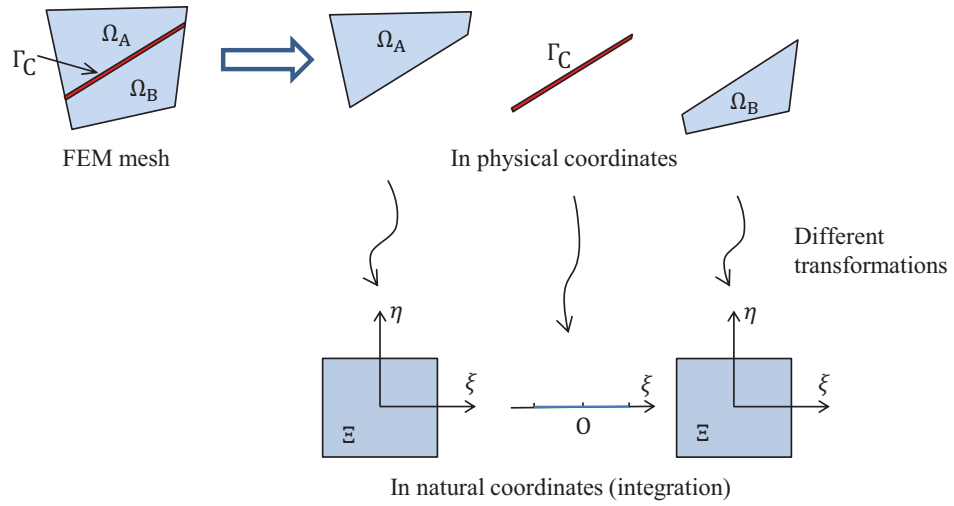


(b) Arbitrary quadrilateral numerical domain

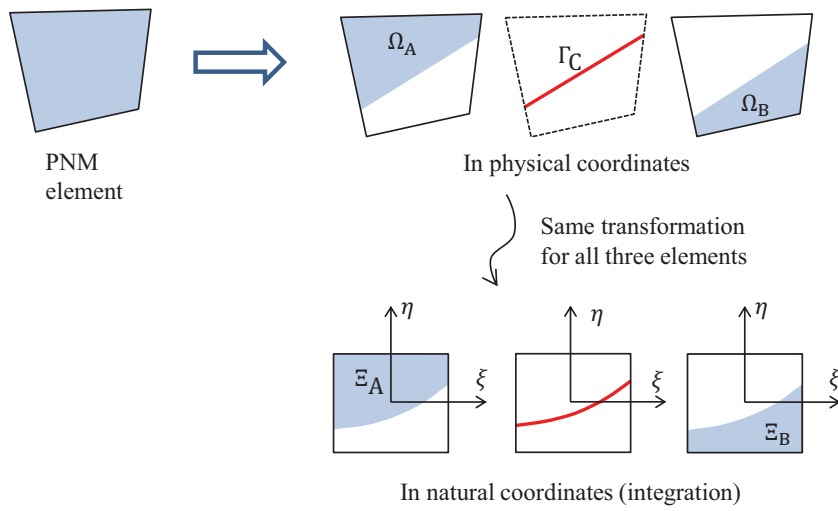


(c) Cohesive crack not parallel to edges of the PNM element

Figure A.15: Situations where the mapping between the FEM element and the PNM element does not exist



(a) FEM solution procedure



(b) PNM solution procedure

Figure A.16: Comparison of FEM and PNM solutions in a generic case

of a FEM mesh under generic situations and the integration and assembly employ standard FEM procedures.

A.6 Conclusion

The PNM is reviewed and compared against the FEM. It was found that under certain conditions, the PNM solution is the same as that of the FEM. A complete separation of numerical domain and material domain is proposed, where a PNM element can have all its nodes completely detached from the material domain, enabling the representation of cohesive cracks by PNM cohesive elements where all its nodes are phantom. This extension facilitates the modelling of multiple cohesive cracks within a PNM element, and enables the capturing of interaction mechanisms between intersecting cracks in a composite laminate. This extension has now evolved into the FNM in Chapter 4 which is simpler, more accurate and more generic.

A.7 Publications

The content of this chapter has produced the following publications:

- B. Y. Chen, P. M. Baiz, S. T. Pinho, and T. E. Tay. An extended phantom node method for crack interactions in composites. In *Proceedings of the 20th UK Conference of the Association for Computational Mechanics in Engineering, the University of Manchester, Manchester, UK*, March 2012
- B. Y. Chen, S. T. Pinho, P. M. Baiz, and T. E. Tay. An extended phantom node method for crack interactions in composites. In *Proceedings of the 10th World Congress on Computational Mechanics, São Paulo, Brazil*, number 19582, July 2012

Appendix B

Error in the mapping of a straight crack in the PNM/XFEM

In the actual XFEM/PNM calculation, a curved crack is often approximated as a straight crack so as to form polygonal sub-domains for numerical integration [51]. This approximation would introduce an error in the areas of integration for the two sub-domains. Consider, as an example, the linear quadrilateral element with domain in physical coordinates shown in Figure B.1 (a); a straight crack passes through the origin, and the slope of the line is k . Using standard shape functions for linear quadrilateral elements, we can express the physical coordinates of a point (x, y) in terms of the natural coordinates (ξ, η) as:

$$\begin{aligned} x(\xi, \eta) &= \sum_{i=1}^4 N_i(\xi, \eta) x_i = \frac{\delta}{4} + \left(1 + \frac{\delta}{4}\right) \xi + \frac{\delta}{4} \eta + \frac{\delta}{4} \xi \eta, \\ y(\xi, \eta) &= \sum_{i=1}^4 N_i(\xi, \eta) y_i = \eta, \end{aligned} \tag{B.1}$$

where N_i are the shape functions of a linear quadrilateral element and (x_i, y_i) are the physical coordinates of node i whose values are shown in Figure B.1 (a).

The equation of the line of crack in the physical coordinates shown in

Figure B.1 (a) is:

$$y = kx. \quad (\text{B.2})$$

In XFEM/PNM, this line is mapped to the natural coordinates through the shape functions of the original element, i.e., shape functions of the quadrilateral. Using Equation B.1, we can write Equation B.2 in natural coordinates:

$$\eta(\xi) = \frac{k\delta + k(4 + \delta)\xi}{(4 - k\delta) - k\delta\xi}, \quad (\text{B.3})$$

which shows that the straight line given by Equation B.2 in the physical coordinate system corresponds to a curved line given by Equation B.3 in the natural coordinate system. Denoting:

$$A = k\delta, \quad B = k(4 + \delta), \quad C = 4 - k\delta. \quad (\text{B.4})$$

Equation B.3 can be expressed as:

$$\eta(\xi) = \frac{A + B\xi}{C - A\xi} = -\frac{B}{A} - \frac{1 + \frac{BC}{A^2}}{\xi - \frac{C}{A}}. \quad (\text{B.5})$$

The actual area of Ξ_B in Figure B.1 (b) can be obtained as:

$$\text{Area}_{\Xi_B} = \int_{-1}^1 (\eta(\xi) + 1) d\xi = 2 - \frac{2B}{A} - \left(1 + \frac{BC}{A^2}\right) \ln \left| \frac{A - C}{A + C} \right|. \quad (\text{B.6})$$

If the equation of crack line in the natural coordinates is approximated by connecting a straight line between the two ends ξ_r and ξ_s in Figure B.1 (b), whose coordinates can be easily obtained as:

$$\begin{aligned} \xi_r &= -1, & \eta_r &= -k \\ \xi_s &= -1, & \eta_s &= \frac{k(2 + \delta)}{2 - k\delta}, \end{aligned} \quad (\text{B.7})$$

then the approximated area of is Ξ_B :

$$\text{Area}_{\Xi_B}^* = \eta_r + \eta_s + 2 = 2 - k - \frac{k(2 + \delta)}{2 - k\delta}. \quad (\text{B.8})$$

If $k = 0.5$ and $\delta = 0.5$, Equation B.6 would give an exact area of 2.160 while Equation B.8 would give an approximate area of 2.214, resulting in a

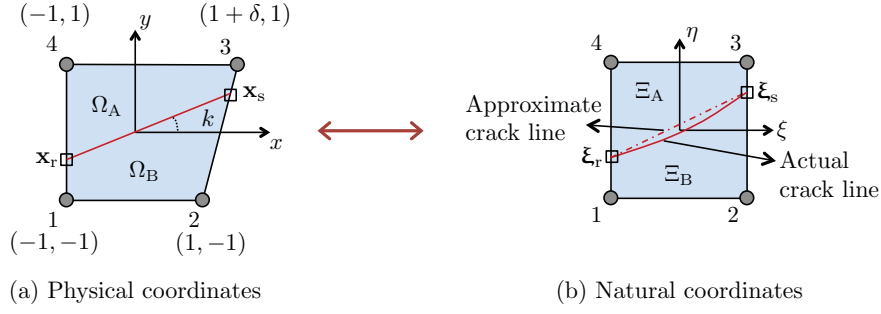


Figure B.1: Geometry of the cracked domain

2.5% over-approximation of the area of Ξ_B ; the exact and approximate area of Ξ_A are then 1.840 and 1.786 respectively, the latter giving a 2.9% under-approximation of the area of Ξ_A . This error is not associated with the FNM (or FEM/remeshing), as the crack line coincides with edges of the elements along which the interpolation is always linear (Figure 4.4). Therefore, a straight crack remains straight after transformation onto the natural space, leading to an exact area of integration. The eventual differences in the calculated stiffness matrix between PNM/XFEM and FNM are however subject to the individual numerical integration schemes used.

Appendix C

Sample codes for the implementation of the floating node method

A sample Abaqus UEL subroutine written in Fortran 77 is attached for reference in Section C.1. The UEL only models a single cohesive crack within an element in a two-dimensional context under the assumption of infinitesimal deformation. However, it suffices to demonstrate the working of the method. The codes are written by the author. Some subroutines for basic matrix operations are modified or re-written from existing codes in the Applied Mechanics Lab, Department of Mechanical Engineering, National University of Singapore. The cohesive element subroutine ‘kcoh2d4.elm’ and related subroutines are modified and re-written from the openly-available UEL code ‘CZM_for_instability.for’ by Gao and Bower in 2004 [52]. The UEL subroutine only considers linear quadrilateral sub-elements and linear cohesive sub-elements for the purpose of method demonstration. More element subroutines can be included and more element types can be considered for the parent and sub-element calculations. Extensions to model more complex cracking cases can be achieved by allowing more floating nodes in the input and enriching the failure criterion subroutine ‘kpilyfail’ to generate more types of failure status and their respective partitions.

The raw input file ‘Job-1.inp’ in Section C.2.1 is created using Abaqus

CAE. It is pre-processed by a Matlab pre-processing programme to obtain the input file ‘uel2d-Job-1.inp’ in Section C.2.2 suitable for the UEL subroutine. The pre-processing programme ‘uelpreproc2d.m’ is attached in Section C.3.

To run the calculation, it suffices to type in the Abaqus command ‘abaqus job=uel2d-Job-1 user=UEL2d.f’. The analysis generates a status file ‘uel2d-Job-1.sta’ and a data file ‘uel2d-Job-1.dat’. The status file tells the number of increments and the data file has all the output variables from the UEL calculation. These two files are passed in a Matlab post-processing programme for the visualization of deformation and failure. The post-processing programme is attached in Section C.4.

C.1 Sample Abaqus UEL subroutine

```

c   Abaqus user-element subroutine to implement the floating node method for the modelling
c   of one cohesive crack within a quadrilateral element; for demonstration purpose, the
c   crack is assumed to appear horizontally in the middle of the parent element.
c   -- codes written by Bo-Yang Chen
c   Department of Mechanical Engineering, National University of Singapore and
c   Department of Aeronautics, Imperial College London
c   -- Last modified on 20 February, 2014, by Bo-Yang Chen
c
c*****
c***** subroutine uel for floating node method *****
c*****
  subroutine uel(rhs,amatrx,svars,energy,ndofel,nrhs,nsvars,
&      props,nprops,coords,mcrd,nnode,u,du,v,a,jtype,time,dtime,
&      kstep,kinc,jelem,params,ndload,jdltyp,adlmag,predef,npredf,
&      lflags,mlvarx,ddlmag,mdload,pnewdt,jprops,njprop,period)
c
  include 'aba_param.inc'
c
  dimension rhs(mlvarx,*),amatrx(ndofel,ndofel),props(*)
  dimension svars(*),energy(8),coords(mcrd,nnode),u(ndofel)
  dimension du(mlvarx,*),v(ndofel),a(ndofel),time(2),params(*)
  dimension jdltyp(mdload,*),adlmag(mdload,*),ddlmag(mdload,*)
  dimension predef(2,npredf,nnode),lflags(*),jprops(*)
c
c   variables to be updated:
c   rhs          residual vector
c   amatrx       element k (stiffness) matrix or jacobian
c   svars        solution dependent variables (sdvs)
c
c   variables passed in:
c   coords       nodal coordinate matrix: coord(j,n) is jth coord of nth node
c   u            element displacement vector, (u_i^1, u_i^2,.. u_i^mcrd), i=1,...,nnode
c   du           element incremental displacement vector
c   nnode        no. of nodes in the element
c   ndofel       no. of dof in the element
c   nprops       no. of real-valued user-defined properties
c   mcrd         max value of coordinates parameter
c
c
c=====
c---- define internal variables -----
c=====
c   2d condition (plain stress/strain)
  integer j2d
c   no. of real nodes and no. of floating nodes
  integer nndr,nndfl,ndofr,ndoffl
c   no. of strains and integration points in parent element
  integer nst,nig0
c   no. of cohesive cracks; no. of sub-elements; max no. of nodes in an elm
  integer ncoh,nsub,mxnd
c   parent element type
  character(len=10):: parent
c
c=====
c --- check element type from jtype: -----
c=====
c   jtype: 4-digit number

```

```

c digit1: number of real nodes
c digit2: number of floating nodes
c digit3: max number of cohesive cracks designed to be modelled by this elm
c digit4: max number of sub-elements designed to be allowed in this elm
c
c      if(jtype .eq. 4413) then
c
c----- example: 2d solid floating node element -----
c nnode=8      8-node element, 4 real and 4 floating
c
c assuming crack appears horizontally in the middle:
c      4----3          4 ---- 3
c      |    | ----> (5)----(6) + (8)----(7)
c      1----2          1 ---- 2
c      partition 1      partition 2: after failure
c ndofel=16
c mcrd=2      (x,y)
c-----
c
c      extract element information from jtype
c      j2d=0 ! 0:plane stress; 1:plane strain
c      nndr=4 ! no. of real nodes = 4 (quadrilateral parent element)
c      parent='quad4'
c      nndfl=4 ! no. of floating nodes = 4 (sufficient for one coh. crack)
c      ndofr=8 ! no. of real dof
c      ndoffl=8 ! no. of fl. dof
c      ncoh=1 ! max no. of coh. crack = 1
c      nsub=3 ! max no. of sub-elms (2 quad4 and 1 coh2d4 in this example)
c      nst=3 ! 3 strains in 2d bulk element
c      mxnd=4 ! max 4 nodes in 2d element
c      nig0=4 ! 4 integration points in quad4 element
c
c      check user element inputs
c      if (nnode .ne. 8) then
c        write(7,*) 'no. of nodes inconsistent!'
c        call xit
c      end if
c      if (mcrd .ne. 2) then
c        write(7,*) 'no. of coordinates inconsistent!'
c        call xit
c      end if
c
c      call kfnm2d(rhs,amatrix,svars,energy,ndofel,nrhs,nsvars,
&      props,nprops,coords,mcrd,nnode,u,du,v,a,jtype,time,dtime,
&      kstep,kinc,jelem,params,ndload,jdltyp,adlmag,predef,npredf,
&      lflags,mlvarx,ddlmag,mdload,pnewdt,jprops,njprop,period,
&      j2d,nndr,nndfl,ndofr,ndoffl,ncoh,nsub,parent,nst,nig0,mxnd)
c      else
c        write(7,*) 'element type undefined!'
c        call xit
c      end if
c
c      return
c      end subroutine uel
c
c
c

```

```

c
c
c
c*****
c***** subroutine kfnm2d for 2d floating node method *****
c*****
      subroutine kfnm2d(rhs,amatrx,svars,energy,ndofel,nrhs,nsvars,
&      props,nprops,coords,mcrd,nnode,u,du,v,a,jtype,time,dtim,
&      kstep,kinc,jelem,params,ndload,jdltyp,adlmag,predef,npredf,
&      lflags,mlvarx,ddlmag,mdload,pnewdt,jprops,njprop,period,
&      j2d,nndr,nndfl,ndofr,ndoffl,ncoh,nsup,parent,nst,nig0,mxnd)

c
      include 'aba_param.inc'

c
      dimension rhs(mlvarx,*),amatrx(ndofel,ndofel),props(*)
      dimension svars(*),energy(8),coords(mcrd,nnode),u(ndofel)
      dimension du(mlvarx,*),v(ndofel),a(ndofel),time(2),params(*)
      dimension jdltyp(mdload,*),adlmag(mdload,*),ddlmag(mdload,*)
      dimension predef(2,npredf,nnode),lflags(*),jprops(*)

c      parent element type
c      character(len=10):: parent
c
c=====
c---- define internal variables -----
c=====
c -- parent element arrays
      dimension xr(mcrd,nndr) !-coordinates of real nodes
      dimension ar(ndofr,ndofr),fr(ndofr),ur(ndofr) !-a*u=f
      dimension sigr(nst,nig0),epsr(nst,nig0)!-stresses&strains at gauss points
      integer cncr(ndofr) !-dof connectivity matrix

c -- floating nodes arrays
      dimension xfl(mcrd,nndfl),ufl(ndoffl) !-coords&dof of fl. nodes; initially all zero
      integer cncfl(ndoffl)

c -- coords of all nodes after failure (fl. nodes coords are crack tip coords)
      dimension coordsf(mcrd,nnode)

c -- sub-element connectivities
      integer cncsub(mxnd+1,nsup) !-max nsup sub-elements with max. mxnd nodes per elm

c -- 2d standard sub-elm relevant arrays
c      4 node quadrilateral sub-element: quad4
      dimension xquad4(2,4) !-coordinates of 4 nodes
      dimension aquad4(8,8),fqquad4(8),uquad4(8) !-a*u=f
      dimension sigquad4(3,4),epsquad4(3,4)!-3 stresses&strains at 4 gauss points
      integer cncquad4(8) !-dof connectivity matrix (quad elm)

c      4 node cohesive sub-element: coh2d4
      dimension xcoh4(2,4) !-coordinates of 4 nodes
      dimension xigcoh4(2,2) !-coordinates of 2 integration points
      dimension acoh4(8,8),fcoh4(8),ucoh4(8) !-a*u=f
      dimension sigcoh4(2,2) !-2 stresses at 2 integration points
      integer cnccoh4(8) !-dof connectivity matrix (coh elm)

c      damage & failure of all coh elms (stored in svars)
      dimension fstcoh4(2,ncoh) !-failure status variables(at 2 intg pnts of all coh elms)
      dimension dcoh4(2,ncoh),decoh4(2,ncoh) !-damage variables
      dimension u0coh4(2,ncoh),ufcoh4(2,ncoh) !-initial&final failure displacements

c      increment counter, failure step and failure increment
      integer iinc,istep0,iinc0
      integer i,ic,j,l,m,kntr !counters; ic: index of coh crack

```

```

integer ksdvpr,ksdvfl,ksdvsl,ksdvcoh !no. sdvs prnt elm, fl nds, sub-elm, coh-elms
integer cntsdv,cntsdv0,cntsdv1,cntsdv2 ! counts of sdvs used
integer ksub,isb ! no. of sub-elms

c
c -- character variable: sub-element type
character(len=10):: element

c
c -- define parameters
parameter (zero=0.d0,one=1.d0,two=2.d0,three=3.d0,four=4.d0)

c
c
c=====
c----- initialize input and stored properties -----
c=====
c
c
c      initialize local matrices and vectors of the parent element
call kinitial(xr,mcrd,nndr)
call kinitial(ar,ndofr,ndofr)
call kinitial(fr,ndofr,1)
call kinitial(ur,ndofr,1)
call kinitial(sigr,nst,nig0)
call kinitial(epsr,nst,nig0)
call kinitial(xfl,mcrd,nndfl)
call kinitial(ufl,ndoffl,1)
call kinitial(coordsf,mcrd,nnode)
call kinitial(fstcoh4,2,ncoh)
call kinitial(dcoh4,2,ncoh)
call kinitial(decoh4,2,ncoh)
call kinitial(u0coh4,2,ncoh)
call kinitial(ufcOH4,2,ncoh)
c
c      initialize integer arrays of parent element
call kinitial_n(cncr,ndofr,1)
call kinitial_n(cncfl,ndoffl,1)
call kinitial_n(cncsub,mxnd+1,nsub)

c
c
c      - read material properties from uel property line in input file
if (nprops .lt. 15) then
  write(7,*) 'no. of properties insufficient!'
  call xit
end if
e1  = props(1) !-young's modulus
e2  = props(2)
e3  = props(3)
g12 = props(4) !-shear modulus
g13 = props(5)
g23 = props(6)
v12 = props(7) !-poisson's ratio
v13 = props(8)
v23 = props(9)
theta = props(10) !- material orientation w.r.t global x direction
yt  = props(11) !-transverse tensile strength
s   = props(12) !-shear strength
gnc = props(13) !-mode i fracture toughness
gsc = props(14) !-mode ii fracture toughness
bketa = props(15) !- eta for bk formula of mixed-mode gc
c
c      - cohesive element penalty stiffness

```

```

cohk=100.d0*max(e2,e3)
c   - initialize cnc and u of real nodes
do i = 1,ndofr
    cncr(i)=i !-connectivity matrix
    ur(i)=u(cncr(i))
end do
c   - initialize cnc and u of floating nodes
do i = 1,ndoffl
    cncfl(i)=ndofr+i !-connectivity matrix
    ufl(i)=u(cncfl(i))
end do
c   - initialize svars array
if(kstep.eq. 1 .and. kinc.eq. 0) then
    do i=1,nsvars
        svars(i)=zero
    end do
end if
c   - read stored variables from svars
c   - about iinc & kinc, iinc0 & istep0:
c   - kinc: the increment number (of a step) of the current iteration given by abaqus
c   - iinc: the increment number (of a step) of the previous iteration stored in svars
c   - if kinc = iinc: the increment hasn't converged (still running iterations)
c   - else: the previous increment has converged and new increment has started
fstat=svars(1) !-failure status of the parent element
iinc=svars(2) !- increment number of last iteration
istep0=svars(3) !- failure step number
iinc0=svars(4) !- failure increment number
ksdvpr=4
c   (4 svars are used to store failure variables of parent element)
c   - update floating nodes coordinates
kntr=0
do i=1,nndfl
    do j=1,mcrd
        kntr=kntr+1
        xfl(j,i)=svars(ksdvpr+kntr)
    end do
end do
ksdvfl=kntr !-ksdvfl=mcrd*nndfl
c   - update sub-element connectivities
kntr=0
do i=1,nsup
    do j=1,mxnd+1
        kntr=kntr+1
        cncsub(j,i)=int(svars(ksdvpr+ksdvfl+kntr))
    end do
end do
ksdvsub=kntr !-ksdvsub = nsup*mxnd
c   - read damage variables of all cohesive cracks
cntsdv1=ksdvpr+ksdvfl+ksdvsub !-counter to count the number of svars used
kntr=0
do ic=1,ncoh
c   cohesive element ic stored failure information
do j=1,2 ! 2 intg pnts for coh2d4 element
    fstcoh4(j,ic)=svars(cntsdv1+kntr+1) !-failure status of coh elm gauss point j
    dcoh4(j,ic)=svars(cntsdv1+kntr+2) !- damage of coh elm gauss point j
    decoh4(j,ic)=svars(cntsdv1+kntr+3) !- damage of prev, equilibrium at gauss point j

```

```

        u0coh4(j,ic)=svars(cntsdv1+kntr+4) !- effect. disp. at fail. init. of gauss point j
        ufcoh4(j,ic)=svars(cntsdv1+kntr+5) !- effect. disp. at total fail of gauss point j
        kntr=kntr+5
    end do
end do
ksdvcoh=kntr !-ksdvcoh=ncoh*10
cntsdv2=cntsdv1+ksdvcoh !-counter to count the number of svars used
(10 svars per cohesive element are used to store variables wrt failure)
c
c=====
c----- postprocessing with nodal coords output in .dat file -----
c=====
c      svars are output in .dat file as sdvs
c      abaqus .dat file format: 9 svars per line;
c      1 coh. crack may create n sub-elms, with max 4 nodes and 8 coords per elm,
c      one line in .dat file can output the nodal coords of one bulk sub-elm;
c      one line can output the coords & damage of intg pnts in one coh sub-elm.
c      thus, the proposed rules for svars output and postprocessing:
c      - initially, the lines output the nodal coords of all the bulk sub-elms;
c      - the first svars in the line is the no. of nodes in the bulk sub-elm;
c      - afterwards, the lines output coords&damage of intg pnts in all coh sub-elms;
c      - if the first svars = -1, it indicates the line is for a coh sub-elm.
c
c      initialize svars counter for nodal coords output in .dat file;
c      if(mod(cntsdv2,9).ne.0) then
c          cntsdv0=(cntsdv2/9+1)*9 !- svars output for nodal coords starts from next line
c      else
c          cntsdv0=cntsdv2
c      end if
c
c      in this example, 1 coh crack partitions the parent elm into 2 bulk sub-elms
c      and 1 coh sub-elm (nsub=3); no. of svars needed is 3*9. in a general case,
c      the crack may partition the parent elm into 1 tri and 1 hex sub-domains;
c      more sub-elms are needed, depending on the partition of the hex sub-domain.
c      without loss of generality, only the case of nsub=3 is demonstrated here.
c      check if the number of svars (nsvars) is sufficient
c      if (nsvars .lt. (cntsdv0+nsub*9)) then
c          write(7,*) 'no. of svars insufficient!'
c          call xit
c      end if
c
c
c=====
c----- when the element is intact : quad4 element -----
c=====
c
c      if (fstat .eq. zero) then !-parent element intact
c          - initialize arrays
c          call kinitial(amatrix,ndofel,ndofel)
c          call kinitial(rhs,ndofel,1)
c      - set initial counter value for svars
c      cntsdv=cntsdv0
c
c***** parent element calculation *****
c      - initialize coords matrix of parent elm

```

```

do i=1,mcrd
  do j=1,nndr
    xr(i,j)=coords(i,j)
  end do
end do

c
c - get a matrix, f vector, stress and strain
if (parent .eq. 'quad4') then
  call kquad4_elm(e1,e2,e3,g12,g13,g23,v12,v13,v23,theta,
1 nndr,mcrd,j2d,nst,nig0,ndofr,xr,ar,fr,ur,epsr,sigr)
else
  write(7,*) 'parent element type not supported!'
  call xit
end if
c*****
c
c***** check for failure initiation in parent element *****
c -failure criterion to calculate failure status and crack tip coords
  call kplyfail(nst,nig0,sigr,yt,s,fstat,parent,
& mcrd,xr,xfl,nndfl,nndr,cncsub,mxnd)
c
  if (fstat .eq. zero) then !- no failure, export amatrix & rhs
c - assemble to global amatrix and rhs
    do i = 1,ndofr
      do j = 1,ndofr
        amatrix(cncr(i),cncr(j))=amatrix(cncr(i),cncr(j))+ar(i,j)
      end do
      rhs(cncr(i),1)=rhs(cncr(i),1)-fr(i)
    end do
c - update nodal positions to svars for postprocessing
    cntsdv=cntsdv+1 ! 1st output of a new line in .dat file
    svars(cntsdv)=nndr+100 !100(indic. parent elm)+no. nodes
    kntr=0
    do i=1,nndr
      do j=1,mcrd !-2d
        cntsdv=cntsdv+1
        kntr=kntr+1
        svars(cntsdv)=xr(j,i)+ur(kntr)
      end do
    end do
    cntsdv=cntsdv+(8-kntr) !-fill up the line
  else !- failure initiated, fstat>1
    istep0=kstep !- failure step number
    iinc0=kinc !- failure increment number
c - update failure info in svars
    svars(1)=fstat
    svars(3)=istep0
    svars(4)=iinc0
c - update fl. nodes coords (crack tip coords) in svars
    kntr=0
    do i=1,nndfl
      do j=1,mcrd
        kntr=kntr+1
        svars(ksdvpr+kntr)=xfl(j,i)
      end do
    end do

```

```

c      - update sub-element connectivities
      kntr=0
      do i=1,nsub
        do j=1,mxnd+1
          kntr=kntr+1
          svars(ksdvpr+ksdvfl+kntr)=cnsub(j,i)
        end do
      end do
    end if

c
c      end if !-fstat=0
c
c
c
c=====
c----- when the element is failed (partitions): -----
c-----
c=====
c
c      if (fstat .gt. zero) then !- parent element failed
c      - initialize arrays
      call kinitial(amatrix,ndofel,ndofel)
      call kinitial(rhs,ndofel,1)
c      - initialize and update coordsf matrix
      do i = 1,nnode
        do j = 1,mcrd
          coordsf(j,i)=coords(j,i)
        end do
      end do
      do i = 1,nndfl
        do j = 1,mcrd
          coordsf(j,i+nndr)=xfl(j,i) !-coords of crack tips
        end do
      end do
c      - set initial counter value for svars
      cntsdv=cntsdv0
c
      ksub=int(fstat/10) !- no. of sub-elements
      ic=0 !-cohesive sub-elm index
c-----
c***** sub-elm calculation *****
c-----
      do isb=1,ksub ! isb: index of sub-element
        if(cnsub(1,isb).gt.0) then !-bulk sub-elm
          nnds=cnsub(1,isb) ! no. of nodes in sub-elm
          if(nnds .eq. 4) then
            element='quad4'
            nig=4 !-no. of intg pnts
            ndofs=8 !-no. of dof in sub-elm
c          - initialize arrays for quad4 sub-element
            call kinitial(xquad4,mcrd,nnds)
            call kinitial(aquad4,ndofs,ndofs)
            call kinitial(fquad4,ndofs,1)
            call kinitial(uquad4,ndofs,1)
            call kinitial(sigquad4,nst,nig)

```

```

      call kinitial(epsquad4,nst,nig)
      call kinitial_n(cncquad4,ndofs,1)
c      - obtain element coords, connec, u vector
      l=0 !-counter
      do i=1,nnds
        do j=1,mcrd
          xquad4(j,i)=coordsf(j,cncsub(1+i,isb))
          l=l+1
          cncquad4(l)=(cncsub(1+i,isb)-1)*mcrd+j
          uquad4(l)=u(cncquad4(l))
        end do
      end do
c      - calculate a matrix and f vector
      call kquad4_elm(e1,e2,e3,g12,g13,g23,v12,v13,v23,theta,
1      nnds,mcrd,j2d,nst,nig,ndofs,xquad4,aquad4,fquad4,uquad4,
1      epsquad4,sigquad4)
c      - assemble to global amatrix and rhs
      do i = 1,ndofs
        do j = 1,ndofs
          amatrix(cncquad4(i),cncquad4(j))=
1          amatrix(cncquad4(i),cncquad4(j))+aquad4(i,j)
        end do
        rhs(cncquad4(i),1)=rhs(cncquad4(i),1)-fquad4(i)
      end do
c      - update nodal positions to svars for postprocessing
      cntsdv=cntsdv+1 ! 1st output of a new line in .dat file
      svars(cntsdv)=nnds ! no. of nodes in this element
      kntr=0
      do i=1,nnds
        do j=1,mcrd !-2d
          cntsdv=cntsdv+1
          kntr=kntr+1
          svars(cntsdv)=xquad4(j,i)+uquad4(kntr)
        end do
      end do
      cntsdv=cntsdv+(8-kntr) !-fill up the line
c
      else ! no. nodes not 4
        write(7,*) 'no. of nodes in bulk sub-elm is not supported!'
        call xit
      end if
c
      else if (cncsub(1,isb).lt.0) then !-coh sub-elm
        ic=ic+1 !-update cohesive sub-elm index
        nnds=-cncsub(1,isb) ! no. of nodes in sub-elm
        if(nnds .eq. 4) then
          element='coh2d4'
          nig=2 !-no. of intg pnts
          ndofs=8 !-no. of dof in sub-elm
c        - initialize arrays for quad4 sub-element
          call kinitial(xcoh4,mcrd,nnds)
          call kinitial(xigcoh4,mcrd,nig)
          call kinitial(acoh4,ndofs,ndofs)
          call kinitial(fcoh4,ndofs,1)
          call kinitial(ucoh4,ndofs,1)
          call kinitial(sigcoh4,mcrd,nig)

```

```

c      call kinitial_n(cnccoh4,ndofs,1)
c      - obtain element coords, connec, u vector
      l=0
      do i=1,nnds
        do j=1,mcrd
          xcoh4(j,i)=coordsf(j,cncsub(1+i,isb))
          l=l+1
          cnccoh4(l)=(cncsub(1+i,isb)-1)*mcrd+j
          ucoh4(l)=u(cnccoh4(l))
        end do
      end do
c      calculate a matrix and f vector
      call kcoh2d4_elm(cohk, yt, s, gnc, gsc, bketa, ucoh4, fcoh4, acoh4,
& xcoh4, xigcoh4, sigcoh4, fstcoh4, dcoh4, decoh4, u0coh4, ufcogh4,
& istep0, iinc0, kstep, kinc, iinc, dtime, ndofs, nnds, nig, mcrd, ic)
c      assemble to global amatrix and rhs
      do i = 1, ndofs
        do j = 1, ndofs
          amatrix(cnccoh4(i),cnccoh4(j))=
1          amatrix(cnccoh4(j),cnccoh4(i))+acoh4(i,j)
        end do
        rhs(cnccoh4(i),1)=rhs(cnccoh4(i),1)-fcogh4(i)
      end do
c      - update damage variables to svars for postprocessing
      cntsdv=cntsdv+1
      svars(cntsdv)=-nig ! neg. value for coh subelm; nig: no. of intg pnts
      kntr=0
      do i=1,nig
        do j=1,mcrd !-2d
          cntsdv=cntsdv+1
          kntr=kntr+1
          svars(cntsdv)=xigcoh4(j,i)
        end do
      end do
      do i=1,nig
        cntsdv=cntsdv+1
        kntr=kntr+1
        svars(cntsdv)=decoh4(i,ic)
      end do
      cntsdv=cntsdv+(8-kntr) !-fill up the line
c
      else
        write(7,*) 'no. of nodes in coh sub-elm is not supported!'
        call xit
      end if
c
      else
        write(7,*) 'no. of nodes in sub-elm is zero!',isb
        call xit
      end if
c
end do ! looping over all sub-elms
c
c      - update damage variables of all cohesive cracks
c      kntr=0
      do ic=1,ncoh !-looping over all cohesive cracks

```

```

c      cohesive element ic failure information
      do j=1,2 ! 2 integ pnts for coh2d4 element
        svars(cntsdv1+kntr+1)=fstcoh4(j,ic) !-failure status of coh elm gauss point j
        svars(cntsdv1+kntr+2)=dcoh4(j,ic) !- damage of coh elm gauss point j
        svars(cntsdv1+kntr+3)=decoh4(j,ic) !- damage of prev, equilibrium at gauss point j
        svars(cntsdv1+kntr+4)=u0coh4(j,ic) !- effect. disp. at fail init. of gauss point j
        svars(cntsdv1+kntr+5)=ufcoh4(j,ic) !- effect. disp. at total fail of gauss point j
        kntr=kntr+5
      end do
    end do

c
    end if !-fstat=1

c
    svars(2)=kinc !- update increment number of last iteration

c
    return
  end subroutine kfnm2d
c*****
c*****
c
c
c
c
c
c*****
c***** subroutine kplyfail *****
c***** quadratic stress failure criteria and element partition criterion *****
c*****
      subroutine kplyfail(nst,nig,sig,yt,s,fstat,element,
& ndim,xelm,xfl,nndfl,nndr,cncsub,mxnd)

c
      include 'aba_param.inc'

c
      dimension sig(nst,*),xelm(ndim,*)
      dimension xfl(ndim,*)
      integer cncsub(mxnd+1,*)

c
      integer ig
      parameter (zero=0.d0,one=1.d0,two=2.d0,three=3.d0,half=one/two)

c
      fstat=zero

c
      if((yt .eq. zero) .or. (s .eq. zero)) then
        write(7,*) 'zero strength value!'
        call xit
      else
        do ig=1,nig
          if(nst .eq. 3) then
            ff=(max(zero,sig(2,ig))/yt)**2+(sig(3,ig)/s)**2
          else
            write(7,*) 'unsupported no. of strains!'
            call xit
          end if
        end do

c
        if(ff .ge. one) then

```

```

        fstat=one
    else
        fstat=zero
    end if
end do
end if

c
c ----- determine crack surface -----
c ideally, the crack tip locations are calculated based on the stress
c or the damage variables in the element integration points;
c this example focuses on the fnm rather than the failure theory;
c thus, for simplicity, crack is assumed to appear horizontally in the middle:
c
c      4----3              4 ---- 3
c      |    |  ----> (5)----(6) + (8)----(7)
c      1----2              1 ---- 2
c      partition 1      partition 2: after failure
if(fstat .eq. one) then
    if(element .eq. 'quad4') then
        fstat=31.d0 ! 3 sub-elms
        xct1=half*xelm(1,1)+half*xelm(1,4)
        yct1=half*xelm(2,1)+half*xelm(2,4)
        xct2=half*xelm(1,2)+half*xelm(1,3)
        yct2=half*xelm(2,2)+half*xelm(2,3)
        xfl(1,1)=xct1 !-fl. node 5 coord
        xfl(2,1)=yct1
        xfl(1,2)=xct2 !-fl. node 6 coord
        xfl(2,2)=yct2
        xfl(1,3)=xfl(1,2) !-fl. node 7 coord
        xfl(2,3)=xfl(2,2)
        xfl(1,4)=xfl(1,1) !-fl. node 8 coord
        xfl(2,4)=xfl(2,1)
        cnclsub(1,1)=4 ! no. of nodes in bulk sub-elm 1
        cnclsub(1,2)=4 ! no. of nodes in bulk sub-elm 2
        cnclsub(1,3)=-4 ! no. of nodes in coh sub-elm 3
        cnclsub(1+1,1)=1 !-sub elm 1 connec
        cnclsub(1+2,1)=2
        cnclsub(1+3,1)=6
        cnclsub(1+4,1)=5
        cnclsub(1+1,2)=8 !-sub elm 2 connec
        cnclsub(1+2,2)=7
        cnclsub(1+3,2)=3
        cnclsub(1+4,2)=4
        cnclsub(1+1,3)=5 !-sub elm 3 connec
        cnclsub(1+2,3)=6
        cnclsub(1+3,3)=7
        cnclsub(1+4,3)=8
    else
        write(7,*) 'unsupported element for failure!'
        call xit
    end if
end if

c

return
end subroutine kplyfail
c*****
c*****

```

```

C
C
C
C*****
C***** subroutine kquad4_elm *****
C*** subroutine to calculate k-matrix and f-vector of a linear quadrilateral element *****
C*****
      subroutine kquad4_elm(e1,e2,e3,g12,g13,g23,v12,v13,v23,theta,
1  nnode,ndim,j2d,nst,nig,ndofel,coords,amatrix,au,u,strain,stress)
C
      include 'aba_param.inc'
C
C  passed-in variables
      dimension amatrix(ndofel,ndofel),au(ndofel),u(ndofel) ! elm system matrices
      dimension coords(ndim,nnode) ! coordinates of the element nodes
      dimension stress(nst,nig),strain(nst,nig) !stress&strain vectors at intg pnts
C
C  local variables
      dimension w(nig),pq(nig,ndim) !integration pnts weight & location
      dimension fn(nnode),dn(nnode,ndim) ! shape functions & their deriv. physical space
      dimension ajac(ndim,ndim),gn(nnode,ndim) ! jacobian & shape func. deriv. natural space
      dimension bee(nst,ndofel),beet(ndofel,nst) ! b matrix and its transpose
      dimension dee(nst,nst),deeg(nst,nst) ! d matrix in local & global coord. syst.
      dimension btd(ndofel,nst),btbd(ndofel,ndofel) ! b'*d & b'*d*b
      dimension stress_l(nst),strain_g(nst) ! local stress and global strain
      integer i,j,ig
C
      parameter (zero=0.d0,one=1.d0,two=2.d0,three=3.d0,four=4.d0,
+   five=5.d0,six=6.d0)
C
C  initialize local matrices and vectors
      call kinitia(w,nig,1)
      call kinitia(pq,nig,ndim)
      call kinitia(fn,nnode,1)
      call kinitia(dn,nnode,ndim)
      call kinitia(ajac,ndim,ndim)
      call kinitia(gn,nnode,ndim)
      call kinitia(bee,nst,ndofel)
      call kinitia(beet,ndofel,nst)
      call kinitia(dee,nst,nst)
      call kinitia(deeg,nst,nst)
      call kinitia(btd,ndofel,nst)
      call kinitia(btbd,ndofel,ndofel)
      call kinitia(stress_l,nst,1)
      call kinitia(strain_g,nst,1)
C
C
C----- calculate material local stiffness matrix dee -----
      call kdeemat(e1,e2,e3,g12,g13,g23,v12,v13,v23,ndim,j2d,nst,dee)
C
C-----calculate material stiffness matrix in global coordinates deeg-----
      call kdeemat_global(nst,dee,deeg,theta)
C
C----- set up integration points and weights -----
      call kquad4_intg(pq,w,nig,ndim)

```

```

c
c----- calculate amatrix -----
do ig=1,nig !-calculate at each int point
c
    call kquad4_shape(pq,fn,dn,nig,ig,nnode,ndim) !-to get n and dn
    call kmatrix_mul(coords,dn,ajac,ndim,nnode,ndim) !-ajac=coords*dn
    call kdeterminant(ajac,detj,ndim) !-get det(ajac)
    call kjac_inv(ajac,detj,ndim) ! invert ajac onto itself
    call kmatrix_mul(dn,ajac,gn,nnode,ndim,ndim) !-gn=dn*inv(ajac)
    call kbeemat(bee,gn,nnode,nst,ndofel) !-calculate b matrix nst*ndofel
    call kmatrix_transpose(bee,beet,nst,ndofel) !-beet = transpose(bee)
    call kmatrix_mul(beet,deeg,btd,ndofel,nst,nst) !-btd = beet * deeg
    call kmatrix_mul(btd,bee,btdb,ndofel,nst,ndofel) !-btdb=beet*deeg*bee

c
    do i=1,ndofel
        do j=1,ndofel
            amatrix(i,j) = amatrix(i,j)+btdb(i,j)*detj*w(ig) !-gauss integration
        end do
    end do

c
c----- calculate strain in global coords: epsilon = b * u -----
do i=1,nst
    do j=1,ndofel
        strain_g(i) = strain_g(i)+bee(i,j)*u(j)
    end do
end do

c----- transform strain into local coords -----
call ktransfer_strain(nst,strain_g,theta) !-strain_g now in local coords

c
c----- calculate stress in local coords -----
call kmatrix_mul(dee,strain_g,stress_l,nst,nst,1)

c
c----- store in stress and strain for output -----
do i=1,nst
    strain(i,ig)=strain_g(i)
    stress(i,ig)=stress_l(i)
end do

c
end do !-looped over all int points. ig=nig

c
c----- calculate a*u -----
call kmatrix_mul(amatrix,u,au,ndofel,ndofel,1) !- a*u = amatrix*u = -rhs

c
return
end subroutine kquad4_elm
c*****
c*****
c
c
c
c*****
c***** shape functions and derivatives for quad4 element *****
c*****
c*****
subroutine kquad4_shape(pq,f,df,nig,ig,nnode,ndim)
include 'aba_param.inc'
c

```

```

dimension f(nnode),df(nnode,ndim),pq(nig,ndim)
c
parameter (zero=0.d0,one=1.d0,quarter=1.d0/4.d0)
c
c
if (ndim .eq. 2) then
  xi=pq(ig,1)
  eta=pq(ig,2)
  if (nnode .eq. 4) then
    f(1)=quarter*(one-xi)*(one-eta)
    f(2)=quarter*(one+xi)*(one-eta)
    f(3)=quarter*(one+xi)*(one+eta)
    f(4)=quarter*(one-xi)*(one+eta)
    df(1,1) = -quarter*(one-eta)
    df(2,1) =  quarter*(one-eta)
    df(3,1) =  quarter*(one+eta)
    df(4,1) = -quarter*(one+eta)
    df(1,2) = -quarter*(one-xi)
    df(2,2) = -quarter*(one+xi)
    df(3,2) =  quarter*(one+xi)
    df(4,2) =  quarter*(one-xi)
  else
    write(7,*) 'no. of nodes incorrect for kquad4_shape!'
    call xit
  end if
else
  write(7,*) 'dimension incorrect for kquad4_shape!'
  call xit
end if
c
c
return
end subroutine kquad4_shape
c
c
c*****
c***** integration points for quad4 element *****
c*****
subroutine kquad4_intg(pq,wt,nig,ndim)
c  this subroutine updates pq(*,*), coords of intg pnts, and wt(*), weights
  include 'aba_param.inc'
c
dimension pq(nig,ndim),wt(nig)
c
parameter (zero=0.d0,one=1.d0,two=2.d0,three=3.d0,four=4.d0,
+ five=5.d0,six=6.d0,half=1.d0/2.d0,one_third=1.d0/3.d0,
+ two_third=2.d0/3.d0,quarter=1.d0/4.d0,one_sixth=1.d0/6.d0)
c
root3= one/sqrt(three)
c
if (ndim .eq. 2) then
  if (nig .eq. 4) then
    pq(1,1)= -root3
    pq(1,2)= -root3
    pq(2,1)=  root3

```

```

        pq(2,2)= -root3
        pq(3,1)= -root3
        pq(3,2)=  root3
        pq(4,1)=  root3
        pq(4,2)=  root3
        do i=1,4
            wt(i) = one
        end do
    else
        write(7,*) 'no. of intg pnts incorrect for kquad4_intg!'
        call xit
    end if
else
    write(7,*) 'dimension incorrect for kquad4_intg!'
    call xit
end if

C
return
end subroutine kquad4_intg
C*****
C*****
C
C
C
C*****
C***** subroutine kdeemat *****
C***** returns the elastic constitutive matrix of the material *****
C*****
        subroutine kdeemat(e1,e2,e3,g12,g13,g23,v12,v13,v23,
& ndim,j2d,nst,dee)
C
        include 'aba_param.inc'
C
        dimension dee(nst,nst)
C
        parameter (zero=0.d0,one=1.d0,two=2.d0,half=one/two)
C
        v21= e2*v12/e1
        v31= e3*v13/e1
        v32= e3*v23/e2
C
        del= one-v12*v21-v13*v31-v23*v32-two*v21*v32*v13
        if (nst .eq. 3) then
            if (j2d .eq. 0) then
                dee(1,1)= e1/(one-v12*v21)
                dee(1,2)= e1*v21/(one-v12*v21)
                dee(2,2)= e2/(one-v12*v21)
                dee(2,1)= dee(1,2)
                dee(3,3)= g12
            else if (j2d .eq. 1) then
                dee(1,1)= e1*(one-v23*v32)/del
                dee(1,2)= e1*(v21+v23*v31)/del
                dee(2,2)= e2*(one-v13*v31)/del
                dee(2,1)= dee(1,2)
                dee(3,3)= g12
            else

```

```

        write(7,*) 'j2d value not supported for kdeemat!'
        call xit
    end if
else
    write(7,*) 'no. of strains not supported for kdeemat!'
    call xit
end if

C
C
    return
end subroutine kdeemat

C
C
C
C*****
C***** subroutine deemat_global *****
C**** to transform d matrix from local to global coordinates *****
C*****
    subroutine kdeemat_global(nst,dee,deeg,theta)

C
    include 'aba_param.inc'

C
    dimension dee(nst,nst), deeg(nst,nst)

C
    parameter (zero=0.d0,one=1.d0,two=2.d0,four=4.d0,
& halfcirc=180.d0,pi=3.14159265359d0)

C
    c=cos(pi*theta/halfcirc)
    s=sin(pi*theta/halfcirc)

C
    if (nst .eq. 3) then
        dee(1,1) = c*c*c*dee(1,1) + two*c*c*s*s*(dee(1,2)
& + two*dee(3,3)) + s*s*s*s*dee(2,2)
        dee(1,2) = s*s*c*c*(dee(1,1) + dee(2,2) - four*dee(3,3))
& + (s*s*s*s+c*c*c*c)*dee(1,2)
        dee(2,1) = dee(1,2)
        dee(2,2) = s*s*s*s*dee(1,1) + two*c*c*s*s*(dee(1,2)
& + two*dee(3,3)) + c*c*c*c*dee(2,2)
        dee(1,3) = s*c*(c*c*(dee(1,1) - dee(1,2) - two*dee(3,3))
& + s*s*(dee(1,2) - dee(2,2) + two*dee(3,3)))
        dee(3,1) = dee(1,3)
        dee(2,3) = s*c*(s*s*(dee(1,1) - dee(1,2) - two*dee(3,3))
& + c*c*(dee(1,2) - dee(2,2) + two*dee(3,3)))
        dee(3,2) = dee(2,3)
        dee(3,3) = c*c*s*s*(dee(1,1)+dee(2,2)-2*dee(1,2))
& + (c*c-s*s)**2*dee(3,3)
    else
        write(7,*) 'no. of strains not supported for kdeemat_global!'
        call xit
    end if

C
    return
end subroutine kdeemat_global
C*****
C*****
C

```

```

C
C
C*****
C***** subroutine kmatrix_mul *****
C***** matrix multiplication *****
C*****
      subroutine kmatrix_mul(a,b,c,l,m,n)
C
      include 'aba_param.inc'
C
      dimension a(l,m), b(m,n), c(l,n)
C
      integer i,j,k
      parameter (zero=0.d0)
C
      do i=1,l
        do j=1,n
          c(i,j) = zero !-initialize
          do k=1,m
            c(i,j) = c(i,j) + a(i,k)*b(k,j)
          end do
        end do
      end do
C
      return
      end subroutine kmatrix_mul
C*****
C*****
C
C
C*****
C***** subroutine kdeterminant *****
C***** returns the determinant of a jacobian matrix *****
C*****
      subroutine kdeterminant(ajacob,detj,ndim)
C
      include 'aba_param.inc'
C
      dimension ajacob(ndim,ndim)
C
      if (ndim .eq. 2) then
        detj= ajacob(1,1)*ajacob(2,2) - ajacob(1,2)*ajacob(2,1)
      else
        write(7,*) 'dimension not supported for kdeterminant!'
        call xit
      end if
C
      return
      end subroutine kdeterminant
C*****
C*****
C
C
C

```

```

c*****
c***** subroutine kbeemat *****
c***** strain-displacement matrix for infinitesimal deformation *****
c*****
c
  subroutine kbeemat(bee,gn,nnode,nst,ndofel)
c
  include 'aba_param.inc'
c
  dimension gn(nnode,*),bee(nst,ndofel)
c
  integer i,j,k,l,m
  parameter (zero=0.d0)
c
  if (nst .eq. 3) then
    do m=1,nnode
      k= 2*m
      l=k-1
      x=gn(m,1)
      y=gn(m,2)
      bee(1,l)= x
      bee(3,k)= x
      bee(2,k)= y
      bee(3,l)= y
    end do
  else
    write(7,*) 'no. of strains not supported for kbeemat!'
    call xit
  end if
c
  return
end subroutine kbeemat
c*****
c*****
c
c*****
c***** subroutine kjac_inv *****
c***** inverts a jacobian matrix onto itself *****
c*****
  subroutine kjac_inv(ajacob,detj,ndim)
c
  include 'aba_param.inc'
c
  dimension ajacob(ndim,ndim)
c
  dimension ajac(ndim,ndim)
  integer i,j
  parameter (zero=0.d0,one=1.d0,two=2.d0)
c
  initialize local matrices and vectors
  call kinitial(ajac,ndim,ndim)
c
  do i=1,ndim
    do j=1,ndim

```

```

        ajac(i,j)=ajacob(i,j)
    end do
end do
C
if(ndim .eq. 2) then
    ajacob(1,1)=ajac(2,2)
    ajacob(2,1)=-ajac(2,1)
    ajacob(1,2)=-ajac(1,2)
    ajacob(2,2)=ajac(1,1)
else
    write(7,*) 'dimension not supported for kjac_inv!'
    call xit
end if
C
if(detj .gt. zero) then
    do i=1,ndim
        do j=1,ndim
            ajacob(i,j)=ajacob(i,j)/detj
        end do
    end do
else
    write(7,*) 'zero or negative detj in kjac_inv!'
    call xit
end if
C
return
end subroutine kjac_inv
C*****
C*****
C
C
C
C*****
C***** subroutine kmatrix_transpose *****
C***** calculate matrix transpose *****
C*****
    subroutine kmatrix_transpose(a,at,m,n)
C
    include 'aba_param.inc'
    dimension a(m,n), at(n,m)
C
    do i=1,m
        do j=1,n
            at(j,i) = a(i,j)
        end do
    end do
C
    return
end subroutine kmatrix_transpose
C*****
C*****
C
C
C
C*****
C***** subroutine ktransfer_strain *****

```

```

c***** transfer strains form global to material coordinate systems *****
c*****
      subroutine ktransfer_strain(nst, strain, theta)
c
      include 'aba_param.inc'
c
      dimension strain(nst)
c
      dimension strn(nst), t(nst, nst)
      integer i, j
      parameter (halfcirc=180.d0, pi=3.14159265359d0,
& zero=0.d0, one=1.d0, two=2.d0)
c
      initialize local matrices and vectors
      call kinitial(strn, nst, 1)
      call kinitial(t, nst, nst)
c
      c=cos(pi*theta/halfcirc)
      s=sin(pi*theta/halfcirc)
c
      if (nst .eq. 3) then
         t(1,1)=c*c
         t(1,2)=s*s
         t(1,3)=c*s
         t(2,1)=s*s
         t(2,2)=c*c
         t(2,3)=-c*s
         t(3,1)=-two*c*s
         t(3,2)=two*c*s
         t(3,3)=c*c-s*s
      else
         write(7,*) 'no. of strains not supported for ktransfer_strain!'
         call xit
      end if
      call kmatrix_mul(t, strain, strn, nst, nst, 1)
c
      do i=1, nst
         strain(i) = strn(i)
      end do
c
      return
      end subroutine ktransfer_strain
c*****
c*****
c
c
c
c*****
c***** subroutine interface *****
c***** subroutine for the calculation of a 2d four node linear cohesive element *****
c*****
      subroutine kcoh2d4_elm(dcoh, yt, s, gnc, gsc, bketa, u, au, amatrix,
& coords, xigp, tract, cfstat, dm, dmeq, u0_eff, uf_eff, istep0, iinc0,
& kstep, kinc, iinc, dtime, ndof, nnode, nig, ndim, icoh)
c
      include 'aba_param.inc'

```

```

c
dimension au(ndof),amatrx(ndof,ndof),u(ndof)
dimension coords(ndim,nnode),xigp(ndim,nig),tract(ndim,nig)
dimension cfstat(nig,*),dm(nig,*),dmeq(nig,*)
dimension u0_eff(nig,*),uf_eff(nig,*)
c      dcoh      penalty stiffness
c      nig      no. integration points
c      icoh      cohesive crack index
c      pq(nig)   local coordinates of integration points
c      w(nig)    weights of integration points
c      f(nnode)  shape function
c      df(nnode) derivative of shape function w.r.t local coordinate
c
dimension w(nig),pq(nig,ndim-1)
dimension f(nnode),df(nnode,ndim-1),ccoords(ndim,nnode)
dimension rnm(ndim),tangent(ndim)
dimension qmatrx(ndim,ndim)
dimension ujump(ndim),fmatrx(ndim,ndof)
dimension delta(ndim),dmatrx(ndim,ndim)
dimension qf(ndim,ndof),dqf(ndim,ndof)
dimension ftqt(ndof,ndim),ftqtdqf(ndof,ndof)
dimension ftqttau(ndof),tract2(ndim)
integer isnewton, kntr,i,j,n
c
parameter (zero=0.d0,one=1.d0,two=2.d0, half=one/two,dfail=one)
c      initialize local matrices and vectors
call kinitia(w,nig,1)
call kinitia(pq,nig,ndim-1)
call kinitia(f,nnode,1)
call kinitia(df,nnode,ndim-1)
call kinitia(ccoords,ndim,nnode)
call kinitia(rnm,ndim,1)
call kinitia(tangent,ndim,1)
call kinitia(qmatrx,ndim,ndim)
call kinitia(ujump,ndim,1)
call kinitia(fmatrx,ndim,ndof)
call kinitia(delta,ndim,1)
call kinitia(dmatrx,ndim,ndim)
call kinitia(qf,ndim,ndof)
call kinitia(dqf,ndim,ndof)
call kinitia(ftqt,ndof,ndim)
call kinitia(ftqtdqf,ndof,ndof)
call kinitia(ftqttau,ndof,1)
call kinitia(tract2,ndim,1)
c      check convergence of last iteration
isnewton=0
if(iinc .eq. kinc) then
    isnewton=1 !-current increment still doing newton iterations
else !-previous increment has converged
    do i=1,nig !-update dmeq
        dmeq(i,icoh)=dm(i,icoh) !-equilibrium dm
    end do
end if
c      calculate current coordinates of nodes
kntr=0
do i=1,nnode

```

```

        do j=1,ndim
            kntr=kntr+1
            ccoords(j,i)=coords(j,i)+u(kntr)
        end do
    end do
c      set up integration points and weights
    call kcoh2d4_intg(pq,w,nig,ndim)
c
c      do ig=1,nig !-looping over all integration points
c
c          shape functions and derivatives
        call kcoh2d4_shape(pq,ig,f,df,nnode,ndim,nig)
c
c          current coordinates of the integration point
        xigp(1,ig)=half*(f(1)*cccoords(1,1)+f(2)*cccoords(1,2)+
&          f(3)*cccoords(1,3)+f(4)*cccoords(1,4))
        xigp(2,ig)=half*(f(1)*cccoords(2,1)+f(2)*cccoords(2,2)+
&          f(3)*cccoords(2,3)+f(4)*cccoords(2,4))
c
c          compute tangent of the interface: node 2 coords - node 1 coords
        tangent(1)=coords(1,2)-coords(1,1)
        tangent(2)=coords(2,2)-coords(2,1)
c          normalize tangent vector
        call kunitv(tangent,det,ndim) !-tangent vector normalized
        det=half*det !-ref. line starts from -1 to 1, length=2
c
c          compute normal of the interface
        rnm(1)=-tangent(2)
        rnm(2)=tangent(1)
c
c          compute q matrix: rotate global coords to local coords
        q = transpose[rnm,tangent]
        do j=1,2
            qmatrx(1,j)=rnm(j)
            qmatrx(2,j)=tangent(j)
        end do
c
c          ujump:displacement jump of the two crack surface, in global coords
        fmatrx: ujump (at each int pnt) = fmatrx*u
        do n = 1,ndim
            do m = 1,nnode/2
                fmatrx(n,n+(m-1)*ndim)= -f(m)
                fmatrx(n,n+(nnode-m)*ndim)=f(m)
            end do
        end do
c          calculate ujump
        call kmatrix_mul(fmatrx,u,ujump,ndim,ndof,1)
c          calculate separation delta in local coords: delta=qmatrx*ujump
        call kmatrix_mul(qmatrx,ujump,delta,ndim,ndim,1)
c
c      -- cohesive law at integration pnt ig -----
        if(cfstat(ig,icoh) .eq. zero) then !-intact
            dnn=dcoh
            dss=dcoh
            do k=1,ndim
                tract(k,ig)=dcoh*delta(k)
            end do
        end if
    end do
end do

```

```

end do
c -- check for failure
fchck=(max(zero,tract(1,ig))/yt)**2+(tract(2,ig)/s)**2
if(fchck .ge. one) then
c cfstat(ig,icoh)=one !-failure initiation
ratio=one/sqrt(fchck) !-fchck may >> 1
c -- effective displacement at failure
u_eff=sqrt(max(zero,delta(1))**2+delta(2)**2)*ratio
c -- effective stress at failure
t_eff=dcch*u_eff
c -- work of normal and shear stress at failure
gn=half*max(zero,tract(1,ig))*max(zero,delta(1))
gs=half*tract(2,ig)*delta(2)
c -- mixed-mode ratio (b-k) at failure
b=gs/(gn+gs)
c -- mixed-mode fracture energy defined at failure
gmc=gnc+(gsc-gnc)*b**bketa
u0_eff(ig,icoh)=u_eff
uf_eff(ig,icoh)=two*gmc/(t_eff)
end if
end if

c
c if(cfstat(ig,icoh) .eq. one) then !-damaged
c
c if(dmeq(ig,icoh) .lt. dfail) then
c -- degradation: linear
dm2=zero
c if(uf_eff(ig,icoh) .le. u0_eff(ig,icoh)) then
dm2=dfail !-brittle failure
else
c -- effective displacement
u_eff=sqrt(max(zero,delta(1))**2+delta(2)**2)
c if(u_eff .ne. zero) then
- linear softening law
c dm2=uf_eff(ig,icoh)*(u_eff-u0_eff(ig,icoh))/
1 (u_eff*(uf_eff(ig,icoh)-u0_eff(ig,icoh)))
dm2=min(dfail,dm2)
dm2=max(zero,dm2)
end if
end if

c
c if(dm2 .ge. dmeq(ig,icoh)) then !-loading
dm(ig,icoh)=dm2
c update stresses
if(delta(1) .gt. zero) then
dnn = dcch*(one-dm(ig,icoh))
else
dnn = dcch !-damage doesn't show in compression
end if
dss = dcch*(one-dm(ig,icoh))
tract(1,ig)=dnn*delta(1)
tract(2,ig)=dss*delta(2)
else !-unloading
dm(ig,icoh)=dmeq(ig,icoh) !-cannot < existing damage level
if(delta(1) .gt. zero) then
dnn = dcch*(one-dm(ig,icoh))

```

```

        else
            dnn = dcoh !-damage doesn't show in compression
        end if
        dss = dcoh*(one-dm(ig,icoh))
        tract(1,ig)=dnn*delta(1)
        tract(2,ig)=dss*delta(2)
    end if !-loading/unloading
C
    else ! dmeq=dfail, element already failed
        dm(ig,icoh)=dfail
        if(delta(1) .gt. zero) then
            dnn = dcoh*(one-dfail)
        else
            dnn = dcoh !-damage doesn't show in compression
        end if
        dss = dcoh*(one-dfail)
        tract(1,ig)=dnn*delta(1)
        tract(2,ig)=dss*delta(2)
    end if
C
end if !-damaged or not
C
C-----
C
C -- calculate a matrix and rhs contribution of integration pnt ig -----
C a(ig) = (fmatrx'*q'*d*q*fmatrx)*w(ig), f(ig)=a*u=(fmatrx'*q'*tract)*w(ig)
C   dmatrx(1,1)=dnn
C   dmatrx(2,2)=dss
C
C   call kmatrix_mul(qmatrx,fmatrx,qf,ndim,ndim,ndof)
C   call kmatrix_transpose(qf,ftqt,ndim,ndof)
C   call kmatrix_mul(dmatrx,qf,dqf,ndim,ndim,ndof)
C   call kmatrix_mul(ftqt,dqf,ftqtdqf,ndof,ndim,ndof)
C
C   do j=1,ndof
C   do k=1,ndof
C       amatrix(j,k) = amatrix(j,k)+ftqtdqf(j,k)*det*w(ig)
C   end do
C   end do
C
C   do j=1,ndim
C       tract2(j)=tract(j,ig)
C   end do
C   call kmatrix_mul(ftqt,tract2,ftqttau,ndof,ndim,1)
C
C   do j=1,ndof
C       au(j)=au(j)+ftqttau(j)*w(ig)*det
C   end do
C
C
C   end do !-looped over all integration pnts
C
C
C   return
C   end subroutine kcoh2d4_elm
C*****

```

```

C*****
C
C
C
C***** shape functions and derivatives for coh2d4 element *****
C*****
C
  subroutine kcoh2d4_shape(pq,ig,f,df,nnode,ndim,nig)
  include 'aba_param.inc'

C
  dimension f(nnode),df(nnode,ndim-1),pq(nig,ndim-1)

C
  integer k,j
  parameter (zero=0.d0,one=1.d0,half=1.d0/2.d0)

C
    if(ndim .eq. 2) then
      xi=pq(ig,1)
      if(nnode .eq. 4) then
        f(1)=half*(one-xi)
        f(2)=half*(one+xi)
        f(3)=f(2)
        f(4)=f(1)
        df(1,1)=-half
        df(2,1)=half
        df(3,1)=df(2,1)
        df(4,1)=df(1,1)
      else
        write(7,*) 'wrong number of nodes for kcoh2d4_shape!'
        call xit
      end if
    else
      write(7,*) 'wrong dimension for kcoh2d4_shape!'
      call xit
    end if

C
C
  return
end subroutine kcoh2d4_shape

C
C
C
C*****
C***** integration points for coh2d4 element *****
C*****
C
  subroutine kcoh2d4_intg(pq,wt,nig,ndim)
C this subroutine outputs pq(*,*), coords of all intg pnts, and wt(*), weight
  include 'aba_param.inc'

C
  dimension pq(nig,ndim-1),wt(nig)

C
  integer i,j
  parameter (zero=0.d0,one=1.d0)

C
    if(ndim .eq. 2) then

```

```

        if(nig .eq. 2) then
c          cn=0.5773502691896260d0 !-gauss
          cn=one !-newton cotes
          pq(1,1)=-cn
          pq(2,1)=cn
          wt(1)=one
          wt(2)=one
        else
          write(7,*) 'wrong no. of intg pnts for kcoh2d4_intg!'
          call xit
        end if
      else
        write(7,*) 'wrong dimension for kcoh2d4_intg!'
        call xit
      end if
    end if
  end if
c
c
  return
end subroutine kcoh2d4_intg
c*****
c*****
c
c
c
c*****
c***** subroutine kunitv *****
c***** normalize vector and return its magnitude *****
c*****
  subroutine kunitv(a,amag,ndim)
c
  include 'aba_param.inc'
c
  dimension a(ndim)
  integer i
  parameter (zero=0.d0)
c
  amag=zero
  do i=1,ndim
    amag=amag+a(i)*a(i)
  end do
  amag=dsqrt(amag)
  if(amag .ne. zero) then
    do i=1,ndim
      a(i)=a(i)/amag
    end do
  else
    write(7,*) 'zero vector for kunitv!'
c    do nothing
  end if
c
c
  return
end subroutine kunitv
c*****
c*****
c
c

```

```

C
C*****
C***** subroutine kinitial *****
C***** initialize a real*8 matrix *****
C*****
      subroutine kinitial(amtrx,nrow,ncol)
C
      include 'aba_param.inc'
C
      dimension amtrx(nrow,ncol)
      integer i,j
      parameter (zero=0.d0)
C
      do i=1,nrow
        do j=1,ncol
          amtrx(i,j)=zero
        end do
      end do
C
      return
      end subroutine kinitial
C*****
C*****
C
C
C
C*****
C***** subroutine kinitial_n *****
C***** initialize an integer matrix *****
C*****
      subroutine kinitial_n(amtrx,nrow,ncol)
C
      include 'aba_param.inc'
C
      integer amtrx(nrow,ncol)
      integer i,j,izero
      parameter (izero=0)
C
      do i=1,nrow
        do j=1,ncol
          amtrx(i,j)=izero
        end do
      end do
C
      return
      end subroutine kinitial_n
C*****
C*****

```

C.2 Input file

C.2.1 Raw input file from Abaqus

```

*Heading
** Job name: Job-1 Model name: Model-1
** Generated by: Abaqus/CAE 6.11-1
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
** PARTS
**
*Part, name=Part-1
*Node
    1,      -0.125,      -0.125
    2,       0.125,      -0.125
    3,      -0.125,       0.125
    4,       0.125,       0.125
*Element, type=CPS4R
1, 1, 2, 4, 3
*End Part
**
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=Part-1-1, part=Part-1
*End Instance
**
*Nset, nset=bottom, instance=Part-1-1
    1, 2
*Nset, nset=top, instance=Part-1-1
    3, 4
*Nset, nset=left, instance=Part-1-1
    1, 3
*Nset, nset=right, instance=Part-1-1
    2, 4
*End Assembly
**
** BOUNDARY CONDITIONS
**
** Name: BC-1 Type: Displacement/Rotation
*Boundary
bottom, 1, 1
bottom, 2, 2
** Name: BC-2 Type: Displacement/Rotation
*Boundary
top, 1, 1
** -----
**
** STEP: load
**
*Step, name=load
*Static
0.0001, 0.007, 7e-08, 0.0001
**
** BOUNDARY CONDITIONS
**
** Name: load Type: Velocity/Angular velocity
*Boundary, type=VELOCITY

```

```
top, 2, 2, 1.  
**  
** OUTPUT REQUESTS  
**  
*Restart, write, frequency=0  
**  
** FIELD OUTPUT: F-Output-1  
**  
*Output, field, variable=PRESELECT  
**  
** HISTORY OUTPUT: H-Output-1  
**  
*Output, history, variable=PRESELECT  
*End Step
```

C.2.2 Pre-processed input file for UEL

```

*Heading
** Job name: Job-1 Model name: Model-1
** Generated by: Abaqus/CAE 6.11-1
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
** PARTS
**
*Part, name=Part-1
*Node
1, -0.125000, -0.125000
2, 0.125000, -0.125000
3, -0.125000, 0.125000
4, 0.125000, 0.125000
5, 0.000000, 0.000000
6, 0.000000, 0.000000
7, 0.000000, 0.000000
8, 0.000000, 0.000000
*USER ELEMENT, TYPE=U4413, NODES=8, COORDINATES=2, PROPERTIES=15, VARIABLES=72
1,2
*Element, type=U4413, elset=whole
1, 1, 2, 4, 3, 5, 6, 8, 7
*Element, type=CPE4, elset=floating
2, 1, 2, 4, 3
**Section: Section-1
*Solid Section, elset=floating, material=Material-1

*UEL PROPERTY, ELSET=whole
100000.000000, 10000.000000, 10000.000000, 5000.000000, 5000.000000, 5000.000000, 0.300000,
0.300000,
0.300000, 0.000000, 60.000000, 90.000000, 0.200000, 1.000000, 2.000000,
*End Part
**
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=Part-1-1, part=Part-1
*End Instance
**
*Nset, nset=bottom, instance=Part-1-1
1, 2
*Nset, nset=top, instance=Part-1-1
3, 4
*Nset, nset=left, instance=Part-1-1
1, 3
*Nset, nset=right, instance=Part-1-1
2, 4
*End Assembly
*Material, name=Material-1
*Elastic
1.000000e-09, 0
**
** BOUNDARY CONDITIONS
**
** Name: BC-1 Type: Displacement/Rotation

```

```

*Boundary
bottom, 1, 1
bottom, 2, 2
** Name: BC-2 Type: Displacement/Rotation
*Boundary
top, 1, 1
** -----
**
** STEP: load
**
*Step, name=load
*Static
0.0001, 0.007, 7e-08, 0.0001
*CONTROLS, PARAMETERS=TIME INCREMENTATION
160, 200, , 300, 240, 50, , 15, , ,
**
** BOUNDARY CONDITIONS
**
** Name: load Type: Velocity/Angular velocity
*Boundary, type=VELOCITY
top, 2, 2, 1.
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=0
**
*EL PRINT,elset=Part-1-1.whole, FREQ=1
SDV,
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, variable=PRESELECT
**
** HISTORY OUTPUT: H-Output-1
**
*Output, history, variable=PRESELECT
*End Step

```

C.3 Pre-processing Matlab programme for Job-1

```

function [] = uelpreproc2d

clc
close('All')

%*****
%***** Define useful variables *****
%*****

filename='Job-1'; %filename

% must be consistent with uel.f
ndim=2; % dimension
elmname=4413;
nnode=8; % total number of nodes in elm;
fltshr=1; % no. of pair of shared floating nodes in the elm
fltint=0; % no. of internal floating nodes in the elm
mxnd=4; % max no. of nodes in a sub-element
nprops=15;
nsvars=72;

meshnnd=10; % max no. of nodes in the mesh
meshnelm=2; % max no. of elms in the mesh

outfreq=1; % output frequency for SDVs in .dat file

% props array
props=zeros(nprops);
props(1)=100000.0; %E1
props(2)=10000.0; %E2
props(3)=10000.0; %E3
props(4)=5000.0; % G12
props(5)=5000.0; % G23
props(6)=5000.0; % G13
props(7)=0.3; % nu12
props(8)=0.3; % nu23
props(9)=0.3; % nu13
props(10)=0.0; % theta, orientation of the ply
props(11)=60.0; % Yt
props(12)=90.0; % S
props(13)=0.2; % Gnc
props(14)=1.0; % Gsc
props(15)=2.0; % Eta (BK law)

%----- Defined based on above info -----

```

```

% .inp file from Abaqus for reading
file1=strcat(filename, '.inp');
file2=strcat('uel2d-', file1);
fid = fopen(file1, 'r'); %r+ for reading and writing
fid2 = fopen(file2, 'w');

% Initialize nodal coords real matrix
ndcoords=zeros(meshnnd,ndim);
% Initialize elm nodes integer matrix; first col. is the no. of nodes in
% this elm
elmnodes=zeros(meshnelm,mxnd+1);

% Read over non-useful lines
tline=fgets(fid);
fprintf(fid2, '%s', tline);
for i=1:8,
    tline=fgets(fid);
    fprintf(fid2, '%s', tline);
end

%*****
%***** Read nodes *****
%*****
nnodes=0; % no. of nodes
tline=fgets(fid);
while ~strcmp('*', tline(1)), % before the *Element
    nnodes=nnodes+1; % update no. of nodes
    % Store the real values of the nodal coords in ndcoords()
    % Re-read line as integers & real numbers.
    if (ndim==1)
        tline=sscanf(tline, '%u%c%f');
    elseif (ndim==2)
        tline=sscanf(tline, '%u%c%f%c%f');
    else
        tline=sscanf(tline, '%u%c%f%c%f%c%f');
    end
    for i=1:ndim,
        ndcoords(nnodes,i)=tline(2*i+1);
    end
    % ndcoords(nnodes,:)
    tline=fgets(fid);
end

%*****
%***** Read elements *****
%*****

```

```

nelm=0; % no. of elements
tline=fgets(fid);
while ~strcmp('*',tline(1)),% before the *
    nelm=nelm+1; % update no. of elms
    % Store the integer values of the nodes of the elm in elmnodes()
    tline=sscanf(tline,'%u%c'); % Re-read line as integers and space
char.
    % sscanf can read repetative formats until end of line.
    elmnodes(nelm,1)=length(tline)-1; % no. of nodes in this elm
    for i=1:length(tline)-1, % first no. in tline is the elm no.
        elmnodes(nelm,i+1)=tline(i+1);
    end
    %   elmnodes(nelm,:)
    tline=fgets(fid);
end

%*****
%***** Write nodes *****
%*****

% Write original nodes:
for i=1:nnodes,
    fprintf(fid2,' %u',i);
    for j=1:ndim-1,
        fprintf(fid2,' %f',ndcoords(i,j));
    end
    fprintf(fid2,' %f\n',ndcoords(i,ndim));
end
% Write shared floating nodes:
if(fltshr > 0)
    for j=1:fltshr,
        for i=1:nnodes,
            fprintf(fid2,' %u',i+nnodes*j);
            for k=1:ndim-1,
                fprintf(fid2,' %f',0); % floating nodes' location doesn't
matter
            end
            fprintf(fid2,' %f\n',0);
        end
    end
end
end
% Write internal floating nodes:
% nodes listed according to elm no.
% eg. first 4 int. fl. nodes are for elm 1
cnt=(1+fltshr)*nnodes;
if(fltint > 0)

```

```

    for i=1:nelm,
    for k=1:fltint,
        cnt=cnt+1;
        fprintf(fid2,' %u',cnt);
        for m=1:ndim-1,
            fprintf(fid2,' %f',0);
        end
        fprintf(fid2,' %f\n',0);
    end
end
end

%*****
%***** Write elements *****
%*****

% ----- Write User Element information -----
fprintf(fid2,...
    '*USER ELEMENT, TYPE=U%u, NODES=%u, COORDINATES=%u, PROPERTIES=%u, ✓
    VARIABLES=%u\n',...
    elmname, nnode, ndim, nprops, nsvars);
for i=1:ndim-1,
    fprintf(fid2,'%u',i);
end
fprintf(fid2,'%u\n',ndim);
% ----- Write *Element -----
fprintf(fid2,'*Element, type=U%u, elset=whole\n',elmname);
cntint=(fltshr+1)*nnodes; % counter for internal nodes
for i=1:nelm,
    fprintf(fid2,' %u',i); % print elm no.
    for j=1:elmnodes(i,1), % print real nodes
        fprintf(fid2,', %u',elmnodes(i,j+1));
    end
    if(fltshr > 0)
        for k=1:fltshr, % print shared floating nodes
            for j=1:elmnodes(i,1), % no. of real nodes in elm i
                fprintf(fid2,', %u',k*nnodes+elmnodes(i,j+1));
            end
        end
    end
    if(fltint > 0)
        for k=1:fltint, % print internal floating nodes
            cntint=cntint+1;
            fprintf(fid2,', %u',cntint);
        end
    end
end

```

```

        fprintf(fid2, '\n');
    end

% ----- Write Overlaying Element information -----
fprintf(fid2, '*Element, type=CPE4, elset=floating\n');
for i=1:nelm,
    fprintf(fid2, ' %u', i+nelm); % print elm no.
    for j=1:elmnodes(i,1), % print real nodes
        fprintf(fid2, ' %u', elmnodes(i,j+1));
    end
    fprintf(fid2, '\n');
end

% ----- Write Overlaying Section information -----
fprintf(fid2, '**Section: Section-1\n');
fprintf(fid2, '*Solid Section, elset=floating, material=Material-1\n');
fprintf(fid2, '\n');

% ----- Write UEL properties -----
fprintf(fid2, '*UEL PROPERTY, ELSET=whole\n');
cntprops=0;
for i=1:nprops,
    cntprops=cntprops+1;
    if (cntprops < 8)
        fprintf(fid2, ' %f', props(i));
    else
        fprintf(fid2, ' %f\n', props(i));
        cntprops=0;
    end
end
if(cntprops~=0)
    fprintf(fid2, '\n');
end

fprintf(fid2, '%s', tline); %write out the last line read
% ----- Copy the rest of *Part-----
while ((length(tline)<9) || (~strcmp('*End Part', tline(1:9)))) ,% before the ✓
*End Part
    tline=fgets(fid);
    fprintf(fid2, '%s', tline);
end

% ----- Copy *ASSEMBLY-----
while ((length(tline)<13) || (~strcmp('*End Assembly', tline(1:13)))) ,% ✓
before the *End Assembly
    tline=fgets(fid);

```

```

        fprintf(fid2,'%s',tline);
end

% ----- Write Material for overlaying elements ----
fprintf(fid2,'*Material, name=Material-1\n');
fprintf(fid2,'*Elastic\n');
fprintf(fid2,'%d, %d\n', 0.000000001, 0.0);

% ----- Copy *BDC -----
while ((length(tline)<4)||(~strcmp('* -',tline(1:4)))),% before the ✓
*STEP
    tline=fgets(fid);
    fprintf(fid2,'%s',tline);
end

%----- Copy *STEP -----
while ((length(tline)<5)||(~strcmp('*Step',tline(1:5)))) ,
    tline=fgets(fid);
    fprintf(fid2,'%s',tline);
end
for i=1:2,
    tline=fgets(fid);
    fprintf(fid2,'%s',tline);
end

%----- Write *Control parameters -----
fprintf(fid2,'*CONTROLS, PARAMETERS=TIME INCREMENTATION\n');
fprintf(fid2,'160, 200, , 300, 240, 50, , 15, , , \n');

%----- Copy *BOUNDARY CONDITIONS -----
while ((length(tline)<9)||(~strcmp('*Restart,',tline(1:9)))) ,
    tline=fgets(fid);
    fprintf(fid2,'%s',tline);
end

%----- Write SDV output to .dat file -----
fprintf(fid2,'**\n');
fprintf(fid2,'*EL PRINT,elset=Part-1-1.whole, FREQ=%u\n',outfreq);
fprintf(fid2,'SDV,\n');

% ----- Copy the rest -----
while ~feof(fid),
    tline=fgets(fid);
    fprintf(fid2,'%s',tline);
end

```

```
fclose(fid);  
fclose(fid2);  
end
```

C.4 Post-processing Matlab programme for Job-1

```

function [] = uelplot2dfigvid

clc
close('All')
%clear mex

% --- turn on profile to see computational time distribution; -----
% --- need to uncomment the last line 'view profile' as well -----
profile on;

%
%*****
%***** Define useful stuffs *****
%*****
%
% ***** Things need to change: *****
fname='uel2d-job-1'; %filename
tnelm=1; % no. of elements in the mesh
xmin=-0.25; % axis limits
xmax=0.25;
ymin=-0.25;
ymax=0.25;

ndim=2; % dimension
mxnd=4; % max no. of nodes in 2d element
nndr=4; % no. of material(real) nodes in the parent element
ncrack=1; % no. of cracks modelled in the parent element
nig=2; % no. of integration points on cohesive element
nsub=3; % no. of sub-elms after break
nsdv0=45; % no. of sdv to read over without storing
nsdv=72; % total no. of sdv
freq=1; % frequency of sdv output

%----- Defined based on above info -----
ndofr=nndr*ndim; % d.o.f per parent element

file1=strcat(fname, '.dat');
file2=strcat(fname, '.sta');
videoname=strcat(fname, '.avi'); % name of the video for output

% Open files
fid = fopen(file1, 'r');
fid2 = fopen(file2, 'r');

% Define empty newline for newline identification later on

```

```

newline = sprintf('\r\n');
newline2 = sprintf('\n');

%*****
%***** Read .sta file for no. of increments *****
%*****
% Read over non-useful stuff
for i=1:5,
    tline=fgets(fid2);
end
% Read through for no. of increments
ninc=0;
tline=fgets(fid2);
while ~strcmp('T',tline(2)),% before the last line
    if ~strcmp('U',tline(15)),% be careful of multiple attempts
        ninc=ninc+1;
    end
    tline=fgets(fid2);
end
ninc=ninc-1;

% Row number: number of increments recorded
nrow=ceil(ninc/freq);

%-----
%----- Initialize arrays -----
%-----
% Initialize sdv
sdv=zeros(nrow,nsdv-nsdv0,tne1m);
% Initialize kinc
kinc=zeros(nrow,1);
% Initialize step time and total time array (currently the same for 1↙
step
% analysis)
tstep=zeros(nrow,1);
ttotal=zeros(nrow,1);
% Initialize x and y coordinate matrices
xcor=zeros(nrow,(nsub-ncrack)*(mxnd+1),tne1m);
ycor=zeros(nrow,(nsub-ncrack)*(mxnd+1),tne1m);
xtemp=zeros(mxnd+1,1);
ytemp=zeros(mxnd+1,1);
% Initialize array for failure status in the elm at each loading step
fstat=zeros(nrow,tne1m);
% Initialize array for the no. of sub-elms at each step in the element

```

```

ksub=zeros(nrow,tneim);
% Initialize integration point coordinate matrices
xgpt=zeros(nrow,nig*ncrack,tneim);
ygpt=zeros(nrow,nig*ncrack,tneim);
% Initialize damage index at gauss points
dm=zeros(nrow,nig*ncrack,tneim);

%*****
%***** Read SDVs from .dat file *****
%*****

% Read over non-useful lines
tline=fgets(fid);
while (length(tline)<9||~strcmp(' (6)',tline(7:9))),
    tline=fgets(fid);
end
for i=1:2,
    tline=fgets(fid);
end

%row number for sdv, irow th increment output
irow=1;

while irow<=nrow,

%***** Read information for one increment *****

%----- Read for increment number -----
% read a line
tline=fgets(fid);
% convert this line into a continuous string with no space in between
inc=sscanf(tline,'%s');
% find the part of the string that contains the number of increment
kincs=inc(10:length(inc)-7);
% convert ninc into an integer
kinc(irow)=sscanf(kincs,'%u');

%----- Read for step time and total time completed -----
% read over empty lines
for i=1:2,
    tline=fgets(fid);
end
% read over time increment info (use it if needed)
tline=fgets(fid);
% read step time & total time

```

```

tline=fgets(fid);
tline=sscanf(tline,'%s');
% find the position of the comma
i=1;
while ~strcmp(',',tline(i)),
    i=i+1;
end
% find the step time completed
tsteps=tline(18:i-1);
tstep(irow)=sscanf(tsteps,'%E');
% find the total time completed
ttotals=tline(i+19:length(tline));
ttotal(irow)=sscanf(ttotals,'%E');
% Read over non-useful lines
for i=1:8,
    tline=fgets(fid);
end

%----- Read for SDVs -----
isdv=0;
% Read all the full rows
while ((isdv+9)<=nsdv),
    % Read over non-useful lines
    for i=1:7,
        tline=fgets(fid);
    end
    % Read first elm's SDVs
    tline=fgets(fid);
    if(~strcmp(newline,tline)&&~strcmp(newline2,tline)), % not empty line
        if(isdv<nsdv0), %SDVs to be ignored
            % Read over other elms
            tline=fgets(fid);
            while(~strcmp(newline,tline)&&~strcmp(newline2,tline)), % not
empty line
                tline=fgets(fid);
            end % when it reaches an empty line
            % Read over non-useful lines
            for i=1:7,
                tline=fgets(fid);
            end
        else % SDVs to be recorded
            % Record first elm's SDVs
            tline=sscanf(tline,'%E');
            for i=1:9,
                sdv(irow,isdv-nsdv0+i,1)=tline(2+i);
            end
        end
    end
    isdv=isdv+9;
end

```

```

        % Record following elms' SDVs
        if(tnelm >= 2),
            ielm=2;
            tline=fgets(fid);
            while(~strcmp(newline,tline)&&~strcmp(newline2,tline)), % ↙
not empty line
                tline=sscanf(tline,'%E');
                for j=1:9,
                    sdv(irow,isdv-nsdv0+j,ielm)=tline(2+j);
                end
                tline=fgets(fid);
                ielm=ielm+1;
            end
        else
            tline=fgets(fid); % read over empty line
        end
        % Read over non-useful lines
        for i=1:7,
            tline=fgets(fid);
        end
    end
else % empty line
    % read over non-useful rows
    for i=1:3,
        tline=fgets(fid);
    end
end
isdv=isdv+9;
end
% Read the remaining non-full rows if any
if(isdv<nsdv),
    % Read over non-useful lines
    for i=1:7,
        tline=fgets(fid);
    end
    % Read SDVs
    tline=fgets(fid);
    if(~strcmp(newline,tline)&&~strcmp(newline2,tline)), % not empty line
        tline=sscanf(tline,'%E');
        for i=1:(nsdv-isdv),
            sdv(irow,isdv-nsdv0+i,1)=tline(2+i);
        end
        % Record following elms' SDVs
        for ielm=2:tnelm,
            tline=fgets(fid);
            tline=sscanf(tline,'%E');

```

```

        for j=1:(nsdv-isdv),
            sdv(irow,isdv-nsdv0+j,ielm)=tline(2+j);
        end
    end
    % Read over non-useful lines
    for i=1:8,
        tline=fgets(fid);
    end
else % empty line
    % read over non-useful rows
    for i=1:3,
        tline=fgets(fid);
    end
end
isdv=nsdv;
end

% Update row number
irow=irow+1;

end

%*****
%***** Plot deformation and failure *****
%*****
writerObj=VideoWriter(videoname);
writerObj.FrameRate = 10; % must be placed before open
writerObj.Quality = 100;
open(writerObj);

scrsz = get(0,'ScreenSize');
figure('Position',[0 0 scrsz(3) scrsz(4)]);
set(gca, 'nextplot','replacechildren');
%***** Figure properties✓
%*****
axis image
axis off % remove all axis labels
% set axis limits so that axis unit scale will not change with each✓
frame
% it's also important for writeVideo to work properly
xlim([xmin,xmax])
ylim([ymin,ymax])
% The above property commands only work when placed after✓
figure/hold

```

```

% Obtain x and y coordinate matrices
for irow=1:nrow, % no. of increments recorded
    for ielm=1:tnelm,
        nndpr=sdv(irow,1,ielm); % 1st sdv for 1st elm
        if (nndpr > 100), % parent elm intact
            fstat(irow,ielm)=0;
            ksub(irow,ielm)=1;
            for isb=1:ksub(irow,ielm), % no. of sub-elements
                ksdrv=1; % count of no. of sdvs
                knds=(isb-1)*(mxnd+1); % count of no. of nodes
                for inds=1:mxnd,
                    xcor(irow,knds+inds,ielm)=sdv(irow,ksdrv+1,ielm);
                    ycor(irow,knds+inds,ielm)=sdv(irow,ksdrv+2,ielm);
                    ksdrv=ksdrv+2;
                end
                % final coords to form a closed plot
                xcor(irow,knds+(mxnd+1),ielm)=xcor(irow,knds+1,ielm);
                ycor(irow,knds+(mxnd+1),ielm)=ycor(irow,knds+1,ielm);
            end
        else % parent elm failed
            fstat(irow,ielm)=1;
            ksub(irow,ielm)=nsub;
            for isb=1:ksub(irow,ielm), % no. of sub-elements
                ksdrv=(isb-1)*9+1;
                knds=(isb-1)*(mxnd+1); % count of no. of nodes
                nnds=sdv(irow,ksdrv,ielm); % 1st sdv for sub elm isb
                if(nnds > 0), % bulk sub-elm
                    for inds=1:mxnd,
                        xcor(irow,knds+inds,ielm)=sdv(irow,ksdrv+1,ielm);
                        ycor(irow,knds+inds,ielm)=sdv(irow,ksdrv+2,ielm);
                        ksdrv=ksdrv+2;
                    end
                    % final coords to form a closed plot
                    xcor(irow,knds+(mxnd+1),ielm)=xcor(irow,knds+1,ielm);
                    ycor(irow,knds+(mxnd+1),ielm)=ycor(irow,knds+1,ielm);
                end
            else % coh sub-elm
                for m=1:ncrack,
                    for n=1:nig,
                        xgpt(irow,(m-1)*nig+n,ielm)=sdv(irow,ksdrv+1,ielm);
                        ygpt(irow,(m-1)*nig+n,ielm)=sdv(irow,ksdrv+2,ielm);
                        ksdrv=ksdrv+2;
                    end
                    for n=1:nig,
                        dm(irow,(m-1)*nig+n,ielm)=sdv(irow,ksdrv+1,ielm);
                        ksdrv=ksdrv+1;
                    end
                end
            end
        end
    end
end

```

```

        end
    end
end
end
end
end

drow=1;
for q=1:floor(nrow/drow),
    irow=q*drow;

% *** To keep the axis properties defined above:***
    set(gca, 'nextplot','replacechildren');

% ----- Plot deformation shape -----
    for ielm=1:tnelem,
        if(fstat(irow,ielm)==0),% parent elm intact
            ksb=ksub(irow,ielm);
        else
            ksb=ksub(irow,ielm)-ncrack;
        end
        for isb=1:ksb, % loop over all bulk sub-elm
            knds=(isb-1)*(mxnd+1);
            % store nodal coords in temporary x & y arrays
            xtemp(1:mxnd+1)=xcor(irow,knds+1:kn ds+mxnd+1,ielm);
            ytemp(1:mxnd+1)=ycor(irow,knds+1:kn ds+mxnd+1,ielm);
% *** comment on the following: ***
% plot gives only the outline
% area gives wierd overlapping of element boundaries
% Patch does not delete the previous graphs
% fill is the best
%         fcolor='b';
%         fcolor=[0.9 0.9 0.9]; % Grey/white color
%         fill(xtemp(1:mxnd+1),ytemp(1:mxnd+1),fcolor,'LineWidth',0.2)
% *** colorbar makes the program very very slow!***
% colorbar('location','EastOutside');
% caxis([0 1]);
% hold on
        end
    end
end

% ----- Plot failure -----
    for ielm=1:tnelem,
        for ig=1:nig*ncrack, % loop over all integration points
            if dm(irow,ig,ielm)==1.0, % failure
                size=25;
%                 size=100*dm(irow,i,ielm); % size of the ball grows with✓
            end
        end
    end
end

```

```

damage
    color='r';
%         color=dm(irow,i,ielm); % color changes w.r.t damage level
    scatter(xgpt(irow,ig,ielm),ygpt(irow,ig,ielm),size,↵
color,'filled')
    end
    end
    end
    hold off

    frame=getframe; %add (gcf) to capture the entire figure
    writeVideo(writerObj,frame);
    if((irow==1)||(rem(irow,10)==0)),
        ffname=int2str(irow);
        % Print figure
        print('-dpdf',ffname);
        print('-depvc',ffname);
    end
end

close(writerObj);
close(ffigure);
fclose(fid);
fclose(fid2);

% Notification of finish
cf = 2000;           % carrier frequency (Hz)
sf = 22050;          % sample frequency (Hz)
d = 0.5;             % duration (s)
n = sf * d;          % number of samples
s = (1:n) / sf;       % sound data preparation
s = sin(2 * pi * cf * s); % sinusoidal modulation
sound(s, sf);         % sound presentation
%pause(d + 0.5);      % waiting for sound end

profile viewer;

end

```