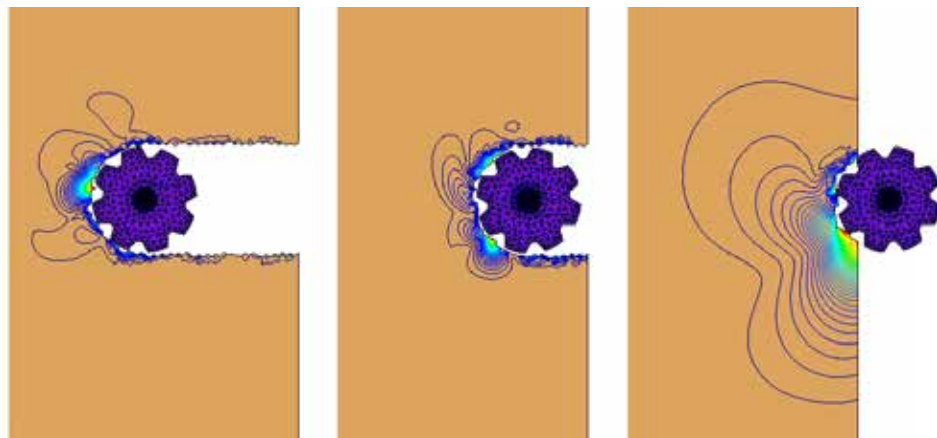


# Modeling of Ground Excavation with the Particle Finite Element Method

J.M. Carbonell  
E. Oñate  
B. Suárez



# **Modeling of Ground Excavation with the Particle Finite Element Method**

J.M. Carbonell  
E. Oñate  
B. Suárez

Monografía CIMNE N<sup>o</sup>-116, February 2010

INTERNATIONAL CENTER FOR NUMERICAL METHODS IN ENGINEERING  
Edificio C1, Campus Norte UPC  
Gran Capitán s/n  
08034 Barcelona, Spain  
[www.cimne.com](http://www.cimne.com)

First edition: February 2010

**MODELING OF GROUND EXCAVATION WITH THE PARTICLE FINITE ELEMENT METHOD**

Monograph CIMNE M116

© The author

ISBN: 978-84-96736-85-6

Depósito legal: B-112410-2010

## AGRAÏMENTS

Voldria donar les gràcies a tota la gent que m'estima, m'ha donat suport i m'ha animat durant l'etapa de la meua vida en la qual he estat elaborant aquest costós treball. No ha estat senzill, m'ha ocasionat molts mals de cap, molts desànim i hi he tingut de dedicar moltíssimes hores. Ho deixo per escrit aquí perquè més endavant no me n'oblidi.

Sóc jo el que ho he viscut, pero hi ha hagut moltes persones al meu costat que ho han compartit. Ells són la meua família, els meus pares Pere Carbonell i Dolors Puigbó i la meua germana M<sup>a</sup> Carme Carbonell i el seu marit. També hi tinc d'afegir una persona molt especial, la meua nòvia Diana-Angela Lazar, amb la qual he començat a construir el meu projecte de vida no precisament de la manera més senzilla. A tots ells els agraeixo de tot cor el seu suport incondicional.

Finalment voldria fer menció d'un home que no vaig conèixer, el meu besavi, que segons m'han explicat, es veu que era una ment inconformista com jo. Suposo que algun tipus d'herència genètica del seu caràcter és el que m'ha portat a ser com sóc i a lluitar sempre per aconseguir coses millors.

Per altra banda, vull agrair també els recursos materials i el suport moral que m'han donat els meus tutors Eugenio Oñate i Benjamín Suárez, els quals han mostrat sempre plena confiança en mi com a persona i com a investigador.



# Abstract

The present work introduces a new application of the Particle Finite Element Method (PFEM) for the modeling of excavation problems.

Lagrangian descriptions of motion of the continuum medium are the natural way of describing motion in solid mechanics. Particle finite element methods are based in these solid mechanics settings that can treat large material deformations and rapidly changing boundaries.

These capabilities are very suitable for modeling fluid motions and moving free surfaces. That is the reason that the most of the research and applications of PFEM can be found in the context of computational fluid dynamics (CFD) instead of solid mechanics. The satisfying results in modeling fluids have been the motivation to use this method in dynamic solid mechanics.

The simulation of an excavation process is a non-linear dynamic problem. It contains geometrical, material and contact non-linearities. Modeling the contact process can be classified as the main difficulty. The simulation face the problems of: the detection of a changing geometry, the detection of contact between several solid domains, the estimation of correct interacting forces, the computation of the wear related to this contact forces and the removing of the material that has been excavated from the model.

The PFEM have its fundamentals in the classical non linear finite element analysis. The formulation provides a foundation in the updated lagrangian formulation for solids. The dynamic problem is integrated using an implicit scheme. Remeshing strategies are employed in order to identify the boundary surfaces. A remeshing of the domain is introduced for the detection of rapidly changing boundaries. The *Delaunay* tessellation and the *alpha-shape* concept together, are used as a methodology to define the boundary from a cloud of points. At the same time an interface mesh is created for contact recognition. A particular constitutive contact law has been developed for capturing contact normal and frictional forces using this mesh.

By means of an *Archard-type* law, the excavation and damage caused in the ground is quantified. The erosion and wear parameters of the grounds under study determine the excavability. The behavior of the geomaterials is described using a *Damage Model* which model fracture in the continuum. The update and storage of the variables in particles have especial schemes for its correct treatment. The method and the meshing process is adapted for the simulation of an excavation problem.

PFEM is presented as a very suitable tool for the treatment of excavation problem. The method gives solution for the analysis of all processes that derive from it. The method has a high versatility and a reasonable computational cost. The obtained results are really promising.



# Resumen

En este trabajo se presenta una nueva aplicación del Particle Finite Element Method (PFEM) para la simulación de procesos de excavación y de desgaste.

La forma habitual de describir el movimiento en mecánica de sólidos es mediante la descripción lagrangeana del movimiento del medio continuo. Los métodos de partículas tienen estas características basadas en la mecánica de sólidos que les permite tratar grandes deformaciones de los materiales y rápidos cambios en los contornos.

Estas capacidades son muy adecuadas para modelar movimientos de fluidos y superficies libres. Por esta razón la mayor investigación y aplicaciones de PFEM se encuentran en el contexto de la dinámica fluidos computacional (CFD) en lugar de la mecánica de sólidos. Los resultados satisfactorios en la modelación de fluidos han motivado el uso del método en la dinámica de la mecánica de sólidos.

Los procesos de excavación son problemas complejos. Su descripción física da lugar a problemas dinámicos no-lineales con múltiples factores que influyen en la no linealidad. Entre ellos, los grandes desplazamientos que se producen, la no-linealidad de los geomateriales y el contacto entre sólidos. Al problema se le añade la particularidad que el material es excavado y tiene que ser extraído del modelo.

El PFEM se fundamenta en la teoría de elementos finitos no lineal. Utiliza una formulación lagrangeana actualizada integrada dentro de un esquema implícito de solución. Los cuerpos del dominio se identifican con una estrategia de remallado. Mediante la teselación de *Delaunay* y el concepto de *Alpha-shapes* se definen los contornos de los cuerpos a partir de una nube de partículas. Al mismo tiempo se usa el remallado para crear una malla de interfase que identifica el contacto y permite modelarlo. Se ha desarrollado una ley constitutiva particular para capturar las fuerzas normales y tangenciales producidas.

La excavación y degradación del material se modela mediante una ley de tipo *Archard*. Ello nos permite cuantificar la erosión y el desgaste del suelo según su excavabilidad. El comportamiento de los geomateriales se describe mediante *Modelos de Daño* que modelan la fractura en el continuo. Los procesos de actualización y almacenamiento de las variables en las partículas tienen esquemas especiales para su correcto tratamiento. El proceso de remallado es adaptado para poder simular la pérdida de material producido por la excavación.

El PFEM se presenta como una buena herramienta para tratar problemas de excavación en suelos y rocas. Así mismo es muy adecuado para simular los procesos que derivan de esos problemas. Tiene una versatilidad muy alta y un coste computacional moderado. Los resultados obtenidos son realmente prometedores.





# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Continuum solid mechanics within the PFEM</b>	<b>5</b>
2.1	Introduction . . . . .	5
2.2	Continuum mechanics . . . . .	5
2.2.1	Motion and deformation . . . . .	5
2.2.2	Strain measures . . . . .	8
2.2.3	Stress measures . . . . .	11
2.3	Lagrangian formulation . . . . .	12
2.3.1	Balance laws . . . . .	12
2.3.2	Weak form of balance of momentum . . . . .	14
2.3.3	Minimum of total potential energy . . . . .	14
2.4	Constitutive models . . . . .	15
2.4.1	Nonlinear Elasticity . . . . .	16
2.4.2	Multiaxial nonlinear elasticity: Hyperelastic response . . . . .	17
2.4.3	Incremental constitutive tensor . . . . .	21
2.4.4	Isotropic damage models . . . . .	22
2.4.5	Damage surfaces . . . . .	28
2.5	Linearizations . . . . .	33
2.5.1	Linearization of the kinematic equations . . . . .	34
2.5.2	Linearization of the constitutive models . . . . .	35
2.5.3	Linearization of the weak form . . . . .	36
2.6	Discretization of the continuum . . . . .	37
2.6.1	Isoparametric concept . . . . .	37
2.6.2	Shape Functions . . . . .	40
2.6.3	Integration in the parameterized space . . . . .	42
2.6.4	Discretization of the weak form . . . . .	43
<b>3</b>	<b>Formulation of the Particle Finite Element Method</b>	<b>49</b>
3.1	Introduction . . . . .	49
3.2	Particle concept . . . . .	49
3.3	Continuum formulation for the PFEM . . . . .	50
3.4	Incremental equations of motion based on particles . . . . .	52
3.5	Updating and transferring information . . . . .	54
3.5.1	Smoothing minimization and variable recovery . . . . .	54
3.5.2	Material allocation to particles . . . . .	61
3.6	Solution algorithms . . . . .	67
3.6.1	Explicit time integration scheme . . . . .	67
3.6.2	Implicit time integration scheme . . . . .	69
3.6.3	Energy conservation methods . . . . .	72
3.6.4	Flowchart for implicit time integration in the PFEM . . . . .	75

3.7	Constitutive laws: Numerical implementation . . . . .	77
3.7.1	Damage model in the PFEM . . . . .	77
3.7.2	Perturbation method to obtain the tangent constitutive tensor . . . . .	78
3.7.3	Updating and transferring constitutive internal variables . . . . .	80
3.7.4	Damage, comparison between the FEM and the PFEM . . . . .	81
<b>4</b>	<b>Meshing process in the PFEM</b>	<b>91</b>
4.1	Introduction . . . . .	91
4.2	Remeshing techniques . . . . .	91
4.2.1	Some combinatorial topology . . . . .	91
4.2.2	Delaunay tessellation and convex hull . . . . .	93
4.2.3	Voronoi diagrams and Delaunay tessellations . . . . .	95
4.3	Boundary recognition . . . . .	95
4.3.1	Alpha shape concept . . . . .	96
4.4	Remeshing strategy for the PFEM . . . . .	99
4.4.1	Assigning non-uniform $\alpha$ -shapes . . . . .	99
4.4.2	Refining $\alpha$ -shapes . . . . .	101
4.4.3	Data structures and algorithms . . . . .	102
4.5	Geometric contact detection . . . . .	107
4.5.1	Settling the boundaries . . . . .	108
4.5.2	Contracting boundaries . . . . .	108
<b>5</b>	<b>Contact mechanics</b>	<b>113</b>
5.1	Introduction . . . . .	113
5.2	Contact kinematics . . . . .	113
5.2.1	Normal Contact . . . . .	114
5.2.2	Tangential Contact . . . . .	116
5.2.3	Normal gap for the PFEM . . . . .	118
5.3	Contact boundary value problem . . . . .	118
5.3.1	Constraint formulation . . . . .	118
5.3.2	Frictionless contact in finite deformations . . . . .	120
5.3.3	Frictional contact . . . . .	121
5.4	Treatment of the contact constraints . . . . .	125
5.4.1	Lagrange multiplier method . . . . .	126
5.4.2	Penalty method . . . . .	127
5.4.3	Continuum constraint method . . . . .	127
5.5	Contact search . . . . .	129
5.5.1	Self-contact in the PFEM . . . . .	130
5.5.2	Contact detection . . . . .	132
5.5.3	Enforcement of the contact constraints . . . . .	133
5.6	Contact mechanics in the PFEM . . . . .	134
5.6.1	Discretization of the contact . . . . .	135
5.6.2	Computation of normal contact forces . . . . .	136

5.6.3	Computation of tangential contact forces . . . . .	139
5.6.4	Linearization of the contact constraint . . . . .	142
5.6.5	Approximation for the tangent . . . . .	148
5.7	Implicit contact scheme . . . . .	150
5.8	Impact . . . . .	154
5.8.1	Longitudinal collinear impact of rods . . . . .	154
5.8.2	Impact of one rod on a rigid wall . . . . .	155
5.8.3	Impact of two rods . . . . .	158
5.8.4	Adaptive time step . . . . .	161
5.9	Numerical examples . . . . .	163
5.9.1	Contact patch test . . . . .	163
5.9.2	Hertzian contact problem . . . . .	165
5.9.3	Contact between an elastic ring and a foundation . . . . .	176
5.9.4	Friction problem . . . . .	180
<b>6</b>	<b>Modeling excavation processes</b>	<b>187</b>
6.1	Introduction . . . . .	187
6.2	Friction . . . . .	188
6.3	Wear . . . . .	189
6.3.1	Abrasive wear . . . . .	190
6.3.2	Impact wear . . . . .	193
6.3.3	Wear calibration test . . . . .	194
6.4	Excavation . . . . .	197
6.4.1	Boring machines . . . . .	198
6.4.2	TBMs . . . . .	198
6.4.3	Roadheaders . . . . .	205
6.5	Modeling excavation with the PFEM . . . . .	207
6.5.1	Particle release . . . . .	208
6.5.2	Surface shaping . . . . .	209
6.5.3	Justification for the conservation of mass . . . . .	212
6.5.4	Constitutive models and excavation . . . . .	212
6.5.5	Algorithm for excavation with the PFEM . . . . .	214
6.6	Excavation scheme . . . . .	215
6.7	Examples . . . . .	216
6.7.1	2D Excavation with a roadheader . . . . .	216
6.7.2	Disc linear cutting test . . . . .	221
6.7.3	3D Excavation with a roadheader . . . . .	228
6.7.4	TBM excavation . . . . .	235
<b>7</b>	<b>Conclusions</b>	<b>243</b>
7.1	Final Remarks . . . . .	243
7.2	Further lines of research . . . . .	244

<b>A</b>	<b>Circumradius</b>	<b>247</b>
A.0.1	Circumcircle . . . . .	247
A.0.2	Circumsphere . . . . .	248
<b>B</b>	<b>Developed Code</b>	<b>251</b>
B.1	Description of the code . . . . .	251
B.2	General scheme of the program . . . . .	252
B.3	Description of the main classes . . . . .	254
B.4	Computational cost and time consuming . . . . .	255

# List of Figures

2.1	Reference configuration and current configuration [131]. . . . .	6
2.2	Transformation between area and volume elements [131] . . . . .	8
2.3	(a) The real damaged solid and (b) the equivalent non-damaged solid [98] .	23
2.4	Exponential hardening and softening functions. . . . .	28
2.5	Linearization of $f$ at $\bar{x}$ [131]. . . . .	33
2.6	Isoparametric description of a deformed element. . . . .	38
2.7	Linear Triangle, 3 nodes element. . . . .	40
2.8	Four-noded tetrahedron. . . . .	41
3.1	Discrete body definition from a cloud of particles . . . . .	50
3.2	The variables are segregated into elements and particles:(a) Transfer of information from particles to nodes, (b) Transfer from nodes to element and (c) Transfer from element to nodes. The way back is also performed when the information is updated and recovered. . . . .	53
3.3	Interpolation of two triangular elements. Four element-node-element stress transfers. . . . .	55
3.4	Clamped shallow circular arch . . . . .	58
3.5	Six curves representing the solutions with the FEM for different meshes and the solution with the PFEM, for the same finer mesh and using the transfer scheme of Box 3.2 . . . . .	59
3.6	Clamped shallow circular arch, three curves representing the theoretical solution, the solution with the FEM and with the PFEM, for the same fine mesh of 18959 elements. . . . .	60
3.7	Elastic beam clamped in one edge and with a load in the other edge. . . . .	60
3.8	Displacements in an elastic beam after loading and downloading cycle for a tensile load, computed with FEM. Curves show deformations for $F = 0; 45 \cdot 10^8; 90 \cdot 10^8; 45 \cdot 10^8; 0$ (N). . . . .	61
3.9	Displacements in an elastic beam after loading and downloading cycle for a tensile load, computed with PFEM. Curves show deformations for $F = 0; 45 \cdot 10^8; 90 \cdot 10^8; 45 \cdot 10^8; 0$ (N). . . . .	61
3.10	Elastic beam after loading and downloading cycle for a tensile load, computed with FEM and PFEM. . . . .	62
3.11	Material assignment from elements to particles in a single body domain. $x$ is determined by the volumes associated to the elements. . . . .	63
3.12	A domain defined by four different materials,(a) Assignment of materials from particles taking in account a transition in the interface (b) No transition between materials, conserving the discontinuity of material properties .	65
3.13	Solution scheme for the PFEM . . . . .	76
3.14	Fracture test model of a notched unreinforced beam. . . . .	82
3.15	Finite element idealization for the notched unreinforced beam. . . . .	82

3.16	Load vs CMSD response for fracture problem using <i>Simo-Ju</i> model with <i>exponential softening</i> . . . . .	83
3.17	Crack pattern and stresses at-peak using <i>Simo-Ju</i> model with <i>exponential softening</i> . Analysis with the FEM on the right and with the PFEM on the left. . . . .	84
3.18	Load vs CMSD response for fracture problem using <i>Simo-Ju</i> model with <i>linear softening</i> . . . . .	85
3.19	Load vs CMSD response for fracture problem using <i>Mohr-Coulomb</i> model with <i>linear softening</i> . . . . .	86
3.20	Crack pattern and stresses at-peak using <i>Simo-Ju</i> model with <i>linear softening</i> . Analysis with FEM on the right and with PFEM on the left. . . . .	87
3.21	Crack pattern and stresses at-peak using <i>Mohr-Coulomb</i> model with <i>linear softening</i> . Analysis with FEM on the right and with PFEM on the left. . . . .	87
3.22	Fracture test model of a 3D notched unreinforced beam. . . . .	88
3.23	Finite element idealization for a 3D notched unreinforced beam. . . . .	88
3.24	Load vs deflection response for fracture problem using <i>Simo-Ju</i> model with <i>exponential softening</i> in a 3D model. . . . .	89
3.25	Crack pattern and strains at-peak using <i>Simo-Ju</i> model with <i>exponential softening</i> . Analysis of a 3D model with FEM on the right and with PFEM on the left. Comparison at the same time step. . . . .	90
3.26	Stresses in the notch area, at-peak using <i>Simo-Ju</i> model with <i>exponential softening</i> . Analysis of a 3D model with FEM on the right and with PFEM on the left. Comparison at the same time step. . . . .	90
4.1	(a) A two-dimensional simplicial complex $K$ consists of 2 triangles (which are shaded), 7 edges, and 5 vertices (b) The underlying space (c) A sub-complex, which is a 1-dimensional simplicial complex, consists of 4 edges, and 4 vertices. (TetGen Manual, [4]) . . . . .	92
4.2	The Delaunay criterion and Delaunay triangulation in two dimensions (a) Both the 2-simplex $abc$ and the 1-simplex $de$ are Delaunay (b) The corresponding Delaunay triangulation of the point set shown in (a)(TetGen Manual, [4]) . . . . .	93
4.3	The relation between Delaunay triangulation in $\mathbb{E}^d$ and convex hull in $\mathbb{E}^{d+1}$ (here $d = 2$ ) (a) Some 2D points and their corresponding 3D lift points (b) The Delaunay triangulation of a set of 2D points and the lower convex hull of its 3D lifted points. (TetGen Manual, [4]) . . . . .	94
4.4	Two flip operations in three dimensions. A 2-to-3 flip replaces a non-locally Delaunay face (the red face on the left figure) into three locally Delaunay faces (the blue and white faces on the right figure); while a 3-to-2 flip does the inverse. (TetGen Manual, [4]) . . . . .	95

4.5	Properties of Delaunay triangulations. (a) The relation between Delaunay triangulation and Voronoi diagram. (b) Locally Delaunay property. Edge $ab$ is locally Delaunay. Here only $c$ and $d$ affect the property because there are triangles $abc$ and $abd$ sharing edge $ab$ . $e$ and $f$ are excluded from the definition of the property.(TetGen Manual,[4]) . . . . .	96
4.6	The alpha-shape parameter is compared with the radius of the circumcircle and the circumsphere formed by a set of points in two-dimensions or three-dimensions respectively, see Appendix A . . . . .	97
4.7	The complete family of alpha shapes of a set of points in two-dimensions.(Del Pin, [36]) . . . . .	98
4.8	Boundary recognition of a set of particles using the $\alpha$ -shape concept ([84]) .	98
4.9	Flowchart for meshing and boundary recognition in the PFEM. From a cloud of points a mesh is defined and filtered by means of the $\alpha$ -shape criterion. . . . .	100
4.10	Starting from an initial two-dimensional geometry (left figure), a cloud of points is defined and a new mesh is created. (a) Resultant mesh applying only the $\alpha$ -shape concept (b) Result after using the normals to the boundaries and refining the $\alpha$ -shape . . . . .	101
4.11	The boundary normals are defined in the initial three-dimensional geometry (on the left), a cloud of points determine the domain and a new mesh is made. First the $\alpha$ -shape concept is applied and complemented with the refining scheme. . . . .	103
4.12	Remeshing schemes. Arrows show the possible updating schemes after remeshing. Last scheme NO REMESH - NO UPDATE coincides with the classical FEM. . . . .	104
4.13	Remeshing scheme for the PFEM . . . . .	105
4.14	Updating scheme for the PFEM after the remesh of the domain . . . . .	106
4.15	Using the alpha-shapes the detection of contact between sub-domains is done automatically. This generates contact elements when the sub-domains are close enough. . . . .	107
4.16	Different sub-domains in contact. An interface mesh is created by the <i>Delaunay</i> tessellation using $\alpha$ -shapes. . . . .	108
4.17	Offset algorithm: (a) Original geometry, (b) Domain discretization, (c) Identification of the contour particles, (d) Inward normals defined on contour particles, (e) Contracted body mesh, (d) Contact interface mesh. . . .	109
4.18	Contact detection and offset algorithm: (a) Original state and geometry, (b) After first detection of active contact elements (black elements) the offset of the boundary is applied, (c) Contracted geometry, (d) Exact geometrical contact. . . . .	111
5.1	Finite deformation contact [132]) . . . . .	113
5.2	Deformed configuration $\mathcal{B}^\alpha$ and minimum distance between bodies [132] . .	114
5.3	Minimum distance problem and coordinate system [132] . . . . .	115



5.4	Path of the point $\mathbf{x}^2$ relative to the contact surface [132]	117
5.5	Stresses in the contact interface [132]	119
5.6	Contact force versus normal gap [132]	119
5.7	Unilateral contact of a solid [132]	120
5.8	<i>Coulomb</i> friction law (Zavarise et al., [134])	123
5.9	Velocity-dependent <i>Coulomb</i> friction law [134]	124
5.10	Regularization of <i>Coulomb</i> friction law [134]	125
5.11	Self-contact elements in 3D formed between two exterior surfaces of the domain: (a) <i>Node to face</i> element, (b) <i>Edge to edge</i> element.	131
5.12	Detection of self-contact in a two dimensional example	131
5.13	Detection of self-contact in a three dimensional example. The offset algorithm is applied to get the exact contact geometry	132
5.14	Activation and de-activation of contact elements and pressure distribution arising from a linear approximation.	134
5.15	Geometrical configuration of a single element, irregular inclined elements can produce tensile forces.	135
5.16	Mesh of contact elements between two domains discretized with triangles	136
5.17	Definition of the normal for each element face. The normal to the surface $\partial\mathcal{B}_\sigma^B$ changes from element to element in the contact domain $\mathcal{B}^c$ .	137
5.18	Contact forces in a two-dimensional contact element.	141
5.19	Tangent vectors to the contact surface [132].	144
5.20	Variation of the surface normals from the reference to the current configuration.	145
5.21	<i>Embedded Newton</i> iteration with contact search.	152
5.22	<i>Rotational Execution</i> of contact search and <i>Newton</i> iteration.	153
5.23	Longitudinal impact of a bar against a rigid wall	156
5.24	Mesh and boundary constrained lines for the numerical analysis of a bar colliding a rigid wall	157
5.25	Evolution of the contact force for a bar colliding a rigid wall	157
5.26	Evolution of the contact force for a bar colliding a rigid wall for bigger and smaller penalizations	158
5.27	Longitudinal impact of two bars	159
5.28	Mesh and boundary constrained lines for the numerical analysis of two colliding bars	159
5.29	Evolution of the contact force for two colliding bars	160
5.30	Evolution of the contact force for two colliding bars using an adaptative time stepping	162
5.31	Patch test between two blocks. Different discretizations are used for each block.	163
5.32	Displacement field on non-conforming meshes.	164
5.33	Mesh and vertical normal stress for a non-uniform displacement field.	164
5.34	Mesh and vertical normal stress for a non-uniform displacement field.	165

5.35	The problem is modelled with three different meshes that refine successively the contact zone. . . . .	166
5.36	Evolution of the contact pressure and number of contact elements in the contact zone for the 2D <i>Hertzian</i> cylinder. . . . .	167
5.37	Evolution of the real contact area for the 2D <i>Hertzian</i> cylinder when the normal force increases. The model is considered as a hardly polished surface.	167
5.38	Numerical results on the 2D finer analyzed mesh. Strains and contact forces and distribution of the normal contact forces. . . . .	168
5.39	Model of a half cylinder in 3D. . . . .	169
5.40	Evolution of the contact pressure on contact surfaces for the 2D and 3D <i>Hertzian</i> cylinders. . . . .	170
5.41	Numerical results 3D cylinder. Stress distribution in the vertical direction. .	170
5.42	Evolution of the real contact area for the 2D and 3D <i>Hertzian</i> cylinders when the normal force increases. . . . .	171
5.43	Numerical results for 3D cylinder. Strain distribution in the vertical direction.	171
5.44	Numerical results for 3D cylinder. Contact normal forces on contact surfaces.	172
5.45	Model of a half sphere in 3D. . . . .	173
5.46	Evolution of the contact pressure on contact surfaces. Comparison between the <i>Hertzian</i> sphere and the <i>Hertzian</i> cylinder. . . . .	173
5.47	Evolution of the real contact area comparing the <i>Hertzian</i> sphere and the <i>Hertzian</i> cylinder. . . . .	174
5.48	Numerical results for 3D sphere. Stress distribution in the vertical direction.	174
5.49	Numerical results for 3D sphere. Strain distribution in the vertical direction.	175
5.50	Numerical results 3D sphere. Contact normal forces on contact surfaces. . .	175
5.51	Compounded elastic ring in contact with a foundation. . . . .	176
5.52	Mesh used for the discretization of the elastic ring and the foundation. . . .	177
5.53	Vertical stress distribution and deformation state. . . . .	178
5.54	Vertical strain, contact force distribution and deformation state. . . . .	179
5.55	Elastic block on a elastic foundation. . . . .	180
5.56	Unstructured mesh used for the discretization of the two elastic domains. .	181
5.57	Evolution of the friction force in the two-dimensional case. . . . .	181
5.58	Evolution of the normal contact forces when the progressive tangential load is applied. . . . .	182
5.59	Shrinkage of the contact boundaries, contact elements and contact forces. .	183
5.60	Vertical stresses and forces with the block resting and tangential forces when the imposed load is applied. . . . .	183
5.61	Unstructured mesh used for the discretization of the two elastic domains of Figure 5.55 in 3D. . . . .	184
5.62	Evolution of the friction force in the 3D case. . . . .	184
5.63	Shrinkage of the contact boundaries, contact elements and contact forces. .	185
5.64	Evolution of the normal contact forces when the progressive tangential load is applied. . . . .	186

6.1	Schematic view of an interface showing the apparent and real areas of contact	188
6.2	A much simplified abrasive wear model showing how a cone removes material from a surface . . . . .	191
6.3	Problem geometry and mesh discretization for the two blocks of steel and rock. The initial mesh has 1426 triangular elements and 840 nodes. . . . .	194
6.4	Volume worn compared with the hardness of the rock surfaces under analysis.	196
6.5	A simplified excavation model for a removal of a solid material by means of a cutting tool. . . . .	197
6.6	Common types of boring machines (Herrenknecht). . . . .	199
6.7	Disc force and geometry for kerf cutting . . . . .	199
6.8	Disc cutter force variation with penetration for high and low strength rocks	200
6.9	A simplified excavation model for removal of solid material by means of a cutting disc. . . . .	201
6.10	Characteristic section for the cutting ring of a disc (Herrenknecht profiles).	201
6.11	Ripping and milling cutting systems. . . . .	205
6.12	Types of picks . . . . .	206
6.13	Drag tools parallel to the cutting direction. $\alpha$ : rake angle; $\beta$ : clearance angle. . . . .	206
6.14	<i>Particle release</i> excavation strategy. When the associated volume of a boundary particle reaches the critical null value the particle is set free. . . . .	208
6.15	Assignment of the volume from elements to particles: (a) With overlapping (b) Without overlapping. . . . .	209
6.16	Error induced in the volume reduction due to the release of consecutive particles. . . . .	210
6.17	Excavation strategy consisting in a <i>Shaping the Surface</i> using the computed volume loss . . . . .	210
6.18	Error induced in the volume reduction due to the inwards movement of consecutive particles. . . . .	211
6.19	Excavation scheme for the PFEM . . . . .	215
6.20	2D model of an excavation problem with a roadheader. . . . .	217
6.21	Initial mesh for the model of a 2D roadheader and the ground. The mesh has 15498 triangular elements and 7919 nodes. . . . .	217
6.22	Computed contact forces on the 2D roadheader. . . . .	218
6.23	Computed contact forces on the 2D roadheader and averaged contact interaction curve. . . . .	219
6.24	2D excavation with a roadheader. . . . .	219
6.25	Stresses, strains and accelerations on the ground due to excavation. Accumulated wear on the roadheader surface at the end of the simulation. . . . .	220
6.26	Geometry of the cutting disc. . . . .	221
6.27	Mesh discretization of the disc and the ground. Initial mesh of 243835 elements and 51109 nodes, including both domains. . . . .	222
6.28	Contact force comparison for the linear cutting test with and without ground damage. . . . .	223

6.29	Number of active contact elements along the analysis. . . . .	224
6.30	Instantaneous wear computed on the surface of the cutting disc. . . . .	224
6.31	Acumulated wear computed on the surface of the cutting disc. . . . .	225
6.32	Stresses, strains and damage on the ground after 10 cm cutting. The disc is hidden but in fact is pushing the last part of the cutting channel. . . . .	226
6.33	Wear on the ring of the disc and rate of excavated volume on the ground with(b) and without(a) a damage constitutive law for the ground modeling. (c) Final state after cutting the whole piece of geo-material. . . . .	227
6.34	Model of a roadheader cutting a ground. . . . .	228
6.35	Discretization of the roadheader excavation model. The initial mesh has 270463 tetrahedra and 52551 nodes. The used time step is $\Delta t = 0.04 s$ . . .	229
6.36	Evolution of the contact forces at the excavation front. . . . .	230
6.37	Evolution of the contact surface. . . . .	230
6.38	Wear progression in time due to the contact interaction machine-ground. . .	231
6.39	Wear on the cutting tool after 20.9 seconds penetrating on the massive rock and at 158.12 seconds, when the roadheader is moving towards the side. . .	232
6.40	Normal forces in the initial contact. Excavation profiles after the roadheader penetration and after ripping along the rock front. . . . .	233
6.41	Contour lines of accelerations produced by the excavation forces. . . . .	234
6.42	Geometry of a real rock cutting head, measures in mm (Herrenknecht). . .	235
6.43	Ground model in the TBM excavation problem. . . . .	236
6.44	Model of the real rock cutting head of the TBM. . . . .	236
6.45	Excavation parameters for the TBM cutting head. . . . .	237
6.46	Discretization of the model. The initial mesh has 704475 tetrahedra and 127299 nodes. . . . .	238
6.47	Normal contact forces produced during the TBM excavation. . . . .	238
6.48	Instantaneous wear computed on the surface of the TBM cutting discs. . . .	239
6.49	Acumulated wear computed on the surface of the TBM cutting discs. . . .	239
6.50	Volume loss due to wear on the TBM discs at time $t = 1.79s$ . . . . .	240
6.51	Excavation rate and stresses on the massive ground in the initial interaction. . . . .	240
6.52	Normal forces and strains produced due to the excavation in the initial interaction. . . . .	241
B.1	Program scheme for the PFEM . . . . .	253



# List of Tables

3.1	Overview of time integration schemes [68]	75
5.1	Friction coefficient for different material pairings [134]	123
6.1	Abrasive Wear Coefficient Values $k_{abr}$ [104]	192
6.2	Mohs's Hardness Values [104]	192
6.3	Impact Wear Coefficient Values $K_{imp}$ [104]	194
6.4	Material properties for the wear test.	195
6.5	Volume loss due to wear on steel.	195
6.6	Volume loss due to wear on rocks.	196
6.7	Comparison of TBM Case Study and Predicted Penetration Rates [38].	202
6.8	Results of $C_c$ prediction equations for a 432-mm-diam cutters [38].	204
6.9	Material properties for the linear cutting test example	222
6.10	Material properties for the roadheader excavation example	229
6.11	Material properties for the TBM excavation example	237
B.1	Time consuming with the programmed code for the PFEM (2,4 GHz processor)	256



# List of Boxes

3.1	Flowchart of the information transfer: Particles-Elements-Particles . . . . .	56
3.2	Modified flowchart of the information transfer: Particles-Elements-Particles	57
3.3	Flowchart for the material assignment from elements to particles . . . . .	64
3.4	Material assignment from particles to elements . . . . .	64
3.5	Flowchart for the numerical implementation of the isotropic damage model	78
3.6	Flowchart for the numerical implementation of the perturbed constitutive tensor for a damage model . . . . .	80
3.7	Rules for updating the constitutive internal variables in the PFEM. . . . .	81
4.1	Flowchart for the boundary refinement. The refining scheme is complemen- tary to the $\alpha$ -shape process. . . . .	102
4.2	The offset algorithm to obtain the contracted or expanded surface . . . . .	110
5.1	Conditions to detect a self-contact element . . . . .	130
5.2	Contact algorithm for the detection of active elements . . . . .	133
5.3	Adaptative time step prediction . . . . .	161
6.1	Flowchart for shaping the worn and excavated surfaces. . . . .	214





*Plans are only good intentions  
unless they immediately degenerate  
into hard work.*

Peter Drucker (1909-2005)

# 1

## Introduction

Underground construction is an important sector in civil engineering. Going underground is sometimes the only or the better building solution. This is an usual alternative in crowded cities where the surface is crowded by facilities, buildings and historical heritage. Gaining space to this environment can only be done going underground. At present, the most common underground facilities which are metro networks, railway stations and city roads.

In future, industry, the big shopping areas and several other facilities will be located below the cities. The surface must be clean, large and pleasant for the daily living of people. This is a particular vision of a future world which aims for a sustainable development for society. But that will be affordable only if a big revolution in the underground construction takes place. It seems it has started. Around the world, hundreds of projects of underground construction are put into practice nowadays.

Scientists and researchers have to contribute to this process and find new solutions to the underground construction requests. With this work we want to do our bit by presenting a new method for the simulation of excavation processes. The excavation of a tunnel or a cavern is only the first part of the building process, but is the most complex and the most important in terms of space production.

An excavation is a complex problem that faces several difficulties. It includes the modeling of the complex tool-soil/rock interactions. This needs adequate constitutive models for representing the evolution of the material properties of the different interacting continua during the excavation process. This also requires a method able to compute large deformations and displacements and multiple contacts between solid domains. All these phenomena have a non-linear behavior which substantially increases the complexity of the solution.

Previous attempts to model numerically ground excavation processes have been made with the discrete element method (DEM) (see [33], [127], [87]). This method discretizes the continua with a finite number of single elements of a certain geometrical shape. These elements try to reproduce the shape of soil grains or disaggregated rock particles. Special contact laws with an important cohesive behavior govern the constitutive material response. One of the handicaps of the DEM technique is the lack of a fundamental base in the constitutive modeling. Another important handicap is the size of the discrete elements. Small sizes are needed in order to converge to the realistic size of the material

particles and this requires a high computational effort. At this time, much research is being done in the field of the DEM. The method is becoming faster and the results are quite promising (see [69], [70]).

An alternative to the DEM is the method presented in this monograph. The Particle Finite Element Method (PFEM) is a new procedure for modeling ground excavation processes which overcomes most of the problem difficulties mentioned above. The objective of this work is to describe the method, to explain all the new advances and show its capabilities for simulating excavation problems. The PFEM is not competing directly with the DEM because their characteristics and applicability are different. The PFEM objective is to be applicable to the modeling of the integral ground excavation process and let the DEM to be applied to single specific parts of the process. Nowadays the DEM is more suitable for local problems. However the PFEM can face all the range of excavation problems and its features will be indeed extended in the future.

The PFEM is an original conjunction of solution strategies. The method uses all the previous background of the standard Finite Element Method (FEM) and introduces new tools to increase the geometrical adaptability to the model via automatic remeshing. PFEM has its foundation on the Lagrangian description of the motion of a continuum built from a set of particles with known physical properties. The FEM is used to solve the governing equations of the continuum in the global domain.

At first sight one gets the impression that the PFEM is the same as the FEM with a remeshing process between time steps. Actually this is a wrong approximation to the definition. A more accurate description is: "*The PFEM is a particle based method supported by a mesh*". A group of particles defines a domain (in an average of smeared sense) and a mesh is used to compute its kinematics which are transferred back to the particles. That cyclic scheme gives to the method excellent capabilities for modeling large displacement and large deformation problems. Globally is a very good method for modeling rapid changes in the domain geometries. In this work the method is applied to non-linear solid mechanics although the original idea of the PFEM comes from Computational Fluid Dynamics (CFD).

The capabilities of the PFEM makes it very suitable for modeling fluid motions and moving free surfaces. That is the reason why most of the research and applications of the PFEM can be found in the context of CFD instead of solid mechanics [57], [86], [36], [77]. The satisfying results in modeling fluids have been the motivation to use this method in dynamic solid mechanics.

Current research is focused in applications of the PFEM to several problems that go beyond the fluid motion description [57], [83], [85], [77]. Fluid-Structure Interaction problems (FSI) with rigid bodies, erosion processes (bed erosion in unsteady flows), mixing processes, coupled thermo-viscous problems and thermal diffusion are some of the applications of the PFEM. A first application of the PFEM in solid mechanics can be found in [94].

This work is the first application of the PFEM for simulating ground excavation. The overall difficulty is to model a problem that has rapidly dynamical changes in a system full of non-linearities including geometrical, material and frictional contact non-linearities.

---

The PFEM is very suitable for the treatment of the large material deformations and rapidly changing boundaries, which typically occurs in excavation processes.

The monograph is structured as follows. First a description the method for solid mechanics is presented, defining the problems involved in the methodology and giving a solution to them. Several new features are added to the PFEM in the field of solid mechanics and contact mechanics. Every characteristic of the method is studied and improved. Wear and excavation theory is coupled to the method culminating with an original application to tunneling processes.

The major contributions of this work are:

- The development of the PFEM in the field of solid mechanics and the adaptation for modeling ground excavation. A new particle definition of the domains and materials as well as new remeshing schemes are presented.
- The development of contact mechanics in the PFEM. A new treatment of the contact problem is presented in this work as a new method termed the *Continuum Constraint* method.
- The adaptation of standart material wear theory to model numerically tool wear and ground excavation, via coupling the FEM with automatic geometry shaping.
- A global PFEM solution scheme has been developed by assembling all these individual contributions. They set a powerful tool for simulating a wide range of excavation processes, from a full tunneling machine to a single cutting tool, including the wear of the excavation tools.

In summary, the emphasis of this work is placed on the demonstration of applicability of the PFEM in solid mechanics and in particular to the simulation of ground excavation. Several examples are presented in order to validate the proposed solutions and schemes. All the advances in this research has been programmed in a new C++ code which has been used to solve all the examples presented in the monograph.

In Chapter 2 the basic mathematical theory concerning the continuum solid mechanics is presented. It is a summary of the standard non-linear FEM where the *Lagrangian* description of motion and the *finite element* resolution is explained. These are very important concepts, but also well-known for those already introduced in the field. For this reason is recommended to avoid this chapter if these concepts are known, as its the purpose is to describe the key particularities of the PFEM only.

In Chapter 3 the main characteristics of the PFEM for solid mechanics are presented. The description of the PFEM is presented from the basis of a particle based method. It provides an answer to all stages of the solution scheme for non-linear solid mechanics. Some validation examples about variable transfer and constitutive modeling with the PFEM are included.

In Chapter 4 the meshing process is described. The importance of the meshing in the PFEM is stated. *Delaunay* triangulations, *alpha-shapes* and several geometrical techniques

are included. The boundary recognition and contact detection via a remeshing process is presented.

In Chapter 5 the theory concerning computational contact mechanics is presented. The classical theory is explained in order to introduce a new contact treatment for the PFEM. The *Contact Constraint* method is explained and validated through different case studies.

In Chapter 6 the friction and wear theories are presented. From the wear mechanics point of view the excavation theory is presented. This provides the basis for the treatment of the excavation with the PFEM. The schemes for the excavation and the geometrical shaping are described and coupled within the global scheme for the PFEM. Several examples showing the potentials of the PFEM for simulating complex excavation processes are presented.

In Annex A and for the sake of completeness, some basics of geometry are included. They provide the numerical basis for the most important criteria for the boundary recognition presented in Chapter 4.

In Annex B the basic structure of PFEM code developed in the monograph is explained. It gives a general view of the organization and the main classes. It is presented from a high conceptual level and not entering in the programming details.

*As far as the laws of mathematics refer to reality,  
they are not certain; and as far as they are certain,  
they do not refer to reality.*

Albert Einstein (1879-1955)

# 2

## Continuum solid mechanics within the PFEM

### 2.1 Introduction

This chapter introduces the general concepts of continuum mechanics theory. This work is focused in nonlinear solid mechanics. The continuum theory and the Finite Element Method (FEM) is the background theory for the Particle Finite Element Method (PFEM). Therefore, the general theory for the PFEM is founded in a well known method, the FEM. That validates the new method which acquires all the important features that derive from the FEM. That characteristics are presented in the next chapters. Next sections are a review of the basic principles of continuum solid mechanics that are essential for the understanding the functioning of the new method.

### 2.2 Continuum mechanics

The objective of continuum mechanics is to provide models for the macroscopic behavior of fluids, solids and structures. This chapter summarizes the main equations which govern the deformation of solids. The extended theory can be found in the literature [12], [13], [81], [132], [137] and [139].

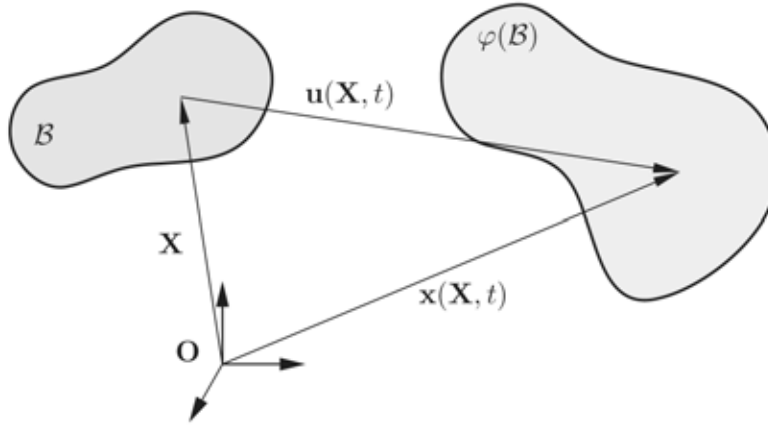
#### 2.2.1 Motion and deformation

The motion and deformation of continua is described by kinematic relations of spatial variables. A body  $\mathcal{B}$  can be defined for a certain number of points which are in a region of the Euclidean space. To describe the behavior of the body the kinematic evolution of a particle in that space has to be analyzed. The position of a particle  $X$  of  $\mathcal{B}$  in the configuration  $\varphi$  is defined as  $\mathbf{x} = \varphi(X)$ . Taking into account the motion of the body, the configuration  $\varphi$  has a time dependence. The position of a particle  $X$  at time  $t$  will be

$$\mathbf{x} = \varphi_t(X) = \varphi(X, t) \quad (2.1)$$

$\varphi_t$  is called the mapping form between configurations. The reference configuration of the body  $\mathcal{B}$  is defined by  $\mathbf{X} = \varphi_0(X) = \varphi(X, t = 0)$ , where  $\mathbf{X}$  is the position of the particle  $X$  in this configuration. For practical applications there is no difference between  $\mathbf{X}$  and  $X$ , the particle is described by its position. This simplifies the notation which can be written as

$$\mathbf{x} = \varphi(\mathbf{X}, t) \quad (2.2)$$



**Figure 2.1:** Reference configuration and current configuration [131].

The positions  $\mathbf{x}$  and  $\mathbf{X}$  are vectors in  $\mathbb{E}^3$ . Point  $X$  is denoted in the reference configuration  $\mathcal{B}$  by the position  $\mathbf{X} = X_A \mathbf{E}_A$  and in the current configuration by  $\mathbf{x} = x_i \mathbf{e}_i$ . Here  $\mathbf{E}_A$  defines an orthogonal basis in the reference configuration and  $\mathbf{e}_i$  is the orthogonal basis in the spatial or current configuration.  $X_A$  are called the *material* or *Lagrangian coordinates* of the particle  $X$  and  $x_i$  are the *spacial* or *Eulerian coordinates*. The reference and current configurations of a body are shown in Figure 2.1.

From now on the notation is: capital letters in indices and tensors will refer to the reference configuration and small letters will be used to denote the current configuration.

The equations of the continuum mechanics can be formulated with respect to the deformed or the undeformed configurations of a body  $\mathcal{B}$ . The theory does not change whether the equations refer to one or another. The differences are considerable in terms of the efficiency when numerical methods for continua are formulated. Strain measures will be defined with respect to both configurations.

Let us define now the displacement of a material point as the difference between its current position and its original position. The displacement vector  $\mathbf{u}(\mathbf{X}, t)$  can be written as:

$$\mathbf{u}(\mathbf{X}, t) = \boldsymbol{\varphi}(\mathbf{X}, t) - \mathbf{X} \quad (2.3)$$

This is shown in Figure 2.1. Using equation (2.2) the displacement is often written as  $\mathbf{u} = (\mathbf{x} - \mathbf{X})$ .

### Time derivatives

Most nonlinear problems are time dependent. There are complete time-dependent processes, like an impact problem, but also nonlinear problems where the constitutive relations are time-dependent or history-dependent, such as in the case of friction between solid materials. Therefore the dependency of the deformation  $\boldsymbol{\varphi}(X, t)$  with time must be considered.

Computing the material time derivatives of the kinematic quantities, the velocity and the acceleration of a material point are obtained:

$$\dot{\mathbf{u}}(\mathbf{X}, t) = \mathbf{v}(\mathbf{X}, t) = \frac{\partial \boldsymbol{\varphi}}{\partial t}(\mathbf{X}, t) = \dot{\boldsymbol{\varphi}}(\mathbf{X}, t) \quad (2.4)$$

$$\ddot{\mathbf{u}}(\mathbf{X}, t) = \mathbf{a}(\mathbf{X}, t) = \ddot{\boldsymbol{\varphi}}(\mathbf{X}, t) = \dot{\mathbf{v}}(\mathbf{X}, t) \quad (2.5)$$

In the current configuration the velocity and the acceleration of a particle  $\mathbf{x}$  at time  $t$  in  $\boldsymbol{\varphi}(\mathcal{B})$ , are written as  $\hat{\mathbf{v}}$ ,  $\hat{\mathbf{a}}$ :

$$\hat{\mathbf{v}}(\mathbf{x}, t) = \hat{\mathbf{v}}(\boldsymbol{\varphi}(\mathbf{X}, t), t) = \mathbf{v}(\mathbf{x}, t) \quad (2.6)$$

$$\hat{\mathbf{a}}(\mathbf{x}, t) = \frac{\partial}{\partial t}[\hat{\mathbf{v}}(\boldsymbol{\varphi}(\mathbf{X}, t), t)] = \frac{\partial \hat{\mathbf{v}}}{\partial t} + \text{grad} \hat{\mathbf{v}} \hat{\mathbf{v}} \quad (2.7)$$

The first term of (2.7) is known as the *local derivative* and the second one is the *convective part* of the time derivative.

### Description of the deformation

The description of the deformation is an essential part of nonlinear mechanics. To describe the deformation process locally the deformation gradient  $\mathbf{F}$  is introduced. The deformation gradient is the tensor that associates to a material line element  $d\mathbf{X}$  in  $\mathcal{B}$  the line element  $d\mathbf{x}$  in  $\boldsymbol{\varphi}(\mathcal{B})$ :

$$d\mathbf{x} = \mathbf{F} d\mathbf{X} \quad (2.8)$$

In the terminology of mathematics, the deformation gradient  $\mathbf{F}$  is the *Jacobian matrix* of the motion  $\boldsymbol{\varphi}(\mathbf{X}, t)$ .  $\mathbf{F}$  maps tangent vectors of the reference configuration to tangent vectors in the spatial configuration.

$$\mathbf{F} = \text{Grad} \boldsymbol{\varphi}(\mathbf{X}, t) = \frac{d\mathbf{x}}{d\mathbf{X}} \quad (2.9)$$

The deformation gradient can be also written as

$$\mathbf{F} = \text{Grad}[\mathbf{X} + \mathbf{u}(\mathbf{X}, t)] = \mathbb{1} + \text{Grad} \mathbf{u} = \mathbb{1} + \mathbf{H} \quad (2.10)$$

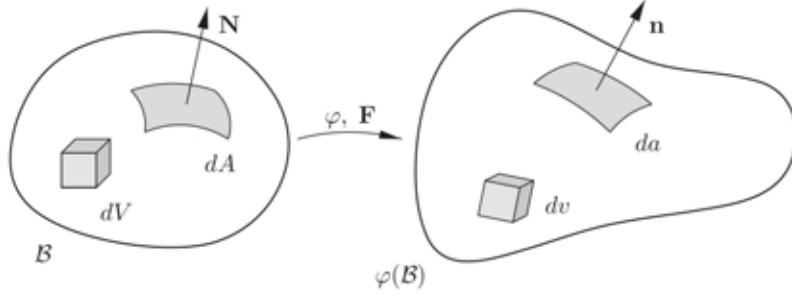
where  $\mathbb{1}$  is the unit matrix and  $\mathbf{H} = \text{Grad} \mathbf{u}$  is the displacement gradient with respect to  $\mathbf{X}$ .

To preserve the continuous structure of  $\mathcal{B}$  during the deformation, the mapping has to be one-to-one; this means that  $\mathbf{F}$  can not be singular. This is equivalent to the condition

$$J = \det \mathbf{F} \neq 0 \quad (2.11)$$

where  $J$  is the *Jacobian determinant*. Furthermore, to exclude self-penetration of the body,  $J$  has to be greater than 0. The deformation gradient is invertible. The inverse is denoted by  $\mathbf{F}^{-1}$  and allows to invert equation (2.8), i.e.





**Figure 2.2:** Transformation between area and volume elements [131]

$$d\mathbf{X} = \mathbf{F}^{-1}d\mathbf{x} \quad (2.12)$$

The deformation gradient  $\mathbf{F}$  can be decomposed into a stretching and rotational part using the polar decomposition theorem, see [76]:

$$\mathbf{F} = \mathbf{R}\mathbf{U} = \mathbf{V}\mathbf{R}, \quad F_{iA} = R_{iB}U_{BA} = V_{ik}R_{kA} \quad (2.13)$$

where  $\mathbf{U}$  is the right stretch tensor with a basis in the reference configuration, and  $\mathbf{V}$  is the left stretch tensor which is an object in the current configuration. The rotation tensor  $\mathbf{R}$  is a two-field tensor which connects both configurations.

It is well known that the transformations of area and volume between  $\mathcal{B}$  and  $\varphi(\mathcal{B})$  can be derived using the deformation gradient  $\mathbf{F}$ . The transformation of area is given by *Nanson formula*,

$$d\mathbf{a} = \mathbf{n} da = J\mathbf{F}^{-T}\mathbf{N}dA = J\mathbf{F}^{-T}d\mathbf{A} \quad (2.14)$$

In this equation  $\mathbf{n}$  is the normal to the surface of  $\varphi(\mathcal{B})$  and  $\mathbf{N}$  is the normal to the surface  $\mathcal{B}$ , (see Figure 2.2). For the transformation of the volume the *Jacobian* determinant is used:

$$dv = JdV \quad (2.15)$$

Equations (2.14) and (2.15) will be also used to relate the integrals in the current and reference configuration in the derivation of the weak forms.

### 2.2.2 Strain measures

Different strain measures are used in non linear continuum mechanics. In this section the more usual measures of strain will be explained. One of the most common ones is the right *Cauchy-Green* tensor  $\mathbf{C}$ , defined in the initial configuration  $\mathcal{B}$  as

$$\mathbf{C} = \mathbf{F}^T\mathbf{F} \quad (2.16)$$

In the initial state  $\mathbf{F} = \mathbf{1}$  and hence this strain measure is  $\mathbf{C} = \mathbf{1}$ . To express the strain in the initial configuration it is convenient to introduce the *Green-Lagrange* strain tensor  $\mathbf{E}$

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T \mathbf{F} - \mathbf{1}) = \frac{1}{2}(\mathbf{C} - \mathbf{1}) \quad (2.17)$$

With respect to the current configuration the *Almansi* strain tensor is often used. This is defined as:

$$\mathbf{e} = \frac{1}{2}(\mathbf{1} - \mathbf{b}^{-1}) \quad (2.18)$$

where tensor  $\mathbf{b}$  is denoted as the left *Cauchy-Green* tensor:

$$\mathbf{b} = \mathbf{F} \mathbf{F}^T \quad (2.19)$$

The *Almansi* strain tensor related to the *Green-Lagrange* strain tensor via the following transformation:

$$\mathbf{E} = \mathbf{F}^T \mathbf{e} \mathbf{F} \quad (2.20)$$

### Vector and tensor transformations

Two operations must be defined: the *pull back* operation; that means the transformation between the differential elements from the current to the reference configuration, and the *push forward* operation which is the transformation in the opposite direction, from the initial to the current configuration.

Let us consider a gradient of a scalar field  $G(\mathbf{X}) = g(\mathbf{x}) = g[\boldsymbol{\varphi}(\mathbf{X})]$  and the following transformations:

$$\text{Grad } G = \mathbf{F}^T \text{grad } g \iff \frac{\partial G}{\partial X_A} = \frac{\partial g}{\partial x_i} \frac{\partial x_i}{\partial X_A} \quad (2.21)$$

$$\text{grad } g = \mathbf{F}^{-T} \text{Grad } G \quad (2.22)$$

In an analogous way, considering the gradient of a vector field  $\mathbf{H}(\mathbf{X}) = \mathbf{h}(\mathbf{x}) = \mathbf{h}[\boldsymbol{\varphi}(\mathbf{X})]$  (see equation (2.9)), the equivalent transformations are

$$\text{Grad } \mathbf{H} = \text{grad } \mathbf{h} \mathbf{F} \quad (2.23)$$

$$\text{grad } \mathbf{h} = \text{Grad } \mathbf{H} \mathbf{F}^{-1} \quad (2.24)$$

Applying these general results in the computation of the deformation gradient in terms of the displacement field  $\mathbf{u}[\boldsymbol{\varphi}(\mathbf{X})]$  the following expressions are obtained:

$$\mathbf{F} = \mathbf{1} + \text{Grad } \mathbf{u} \quad (2.25)$$

$$\mathbb{1} = \mathbf{F}^{-1} + \text{Grad } \mathbf{u} \mathbf{F}^{-1} \quad (2.26)$$

$$\mathbf{F}^{-1} = \mathbb{1} - \text{grad } \mathbf{u} \quad (2.27)$$

The last expression shows that the inverse of the deformation gradient can be computed from the displacement field using the gradient of the current configuration.

### Time derivatives

The time derivative of the deformation gradient  $\mathbf{F}$  is obtained from equation (2.9) with (2.4) and (2.24). It yields to

$$\dot{\mathbf{F}} = \text{Grad } \dot{\boldsymbol{\varphi}}(\mathbf{X}, t) = \text{Grad } \mathbf{v} = \text{grad } \hat{\mathbf{v}} \mathbf{F} \quad (2.28)$$

where  $\hat{\mathbf{v}}$  was defined in equation (2.6).

The spatial velocity gradient in (2.28), is often denoted by  $\boldsymbol{\ell}$ . Then the velocity gradient can be described using  $\mathbf{F}$  as follows:

$$\boldsymbol{\ell} = \dot{\mathbf{F}} \mathbf{F}^{-1} \quad (2.29)$$

Another important time derivative is the *Green-Lagrangian* strain tensor derivative. Applying (2.28) to (2.17) the following expression is obtained

$$\dot{\mathbf{E}} = \frac{1}{2}(\dot{\mathbf{F}}^T \mathbf{F} + \mathbf{F}^T \dot{\mathbf{F}}) \quad (2.30)$$

The time derivative of  $\mathbf{E}$  can be written as

$$\dot{\mathbf{E}} = \mathbf{F}^T \frac{1}{2}(\boldsymbol{\ell} + \boldsymbol{\ell}^T) \mathbf{F} = \mathbf{F}^T \mathbf{d} \mathbf{F} \quad (2.31)$$

This expression has a similar structure as (2.20) and thus can be interpreted as a *pull back* of the symmetrical spatial velocity gradient  $\mathbf{d} = \frac{1}{2}(\boldsymbol{\ell} + \boldsymbol{\ell}^T)$  into the initial configuration.

The *Lie-derivative* of a contravariant tensor applied to the *Almansi* strain tensor gives [66], [65]

$$\mathcal{L}_v \mathbf{e} = \mathbf{F}^{-T} \left\{ \frac{\partial}{\partial t} [\mathbf{F}^T \mathbf{e} \mathbf{F}] \right\} \mathbf{F}^{-1} = \mathbf{F} \dot{\mathbf{E}} \mathbf{F}^T \quad (2.32)$$

and consequently

$$\dot{\mathbf{E}} = \mathbf{F}^T \mathcal{L}_v \mathbf{e} \mathbf{F} \quad (2.33)$$

### 2.2.3 Stress measures

The most known stress tensor is the *Cauchy* stress tensor that is denoted by  $\boldsymbol{\sigma}$ . It is a magnitude expressed in the current configuration  $\boldsymbol{\varphi}(\mathcal{B})$ . The *Cauchy theorem* defines the meaning of stresses. The relation between the traction vector  $\mathbf{t}$  and the surface normal vector  $\mathbf{n}$  is

$$\mathbf{t} = \boldsymbol{\sigma}^T \mathbf{n}, \quad \begin{Bmatrix} t_1 \\ t_2 \\ t_3 \end{Bmatrix} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix} \begin{Bmatrix} n_1 \\ n_2 \\ n_3 \end{Bmatrix} \quad (2.34)$$

From the local balance of angular momentum we obtain:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^T \quad (2.35)$$

i.e.  $\boldsymbol{\sigma}$  is a *symmetric* tensor.

There are other stress measures equivalent to the *Cauchy* stress tensor and expressed in the initial configuration  $\mathcal{B}$ . To define them, an equivalence of a force defined in  $\mathcal{B}$  and  $\boldsymbol{\varphi}(\mathcal{B})$  is done. A *pull back* of the stress measures is written as:

$$\int_{\partial\boldsymbol{\varphi}(\mathcal{B})} \boldsymbol{\sigma} \mathbf{n} da = \int_{\partial\mathcal{B}} \boldsymbol{\sigma} J \mathbf{F}^{-T} \mathbf{N} dA = \int_{\partial\mathcal{B}} \mathbf{P} \mathbf{N} dA \quad (2.36)$$

Above relation defines the first *Piola-Kirchhoff* stress tensor  $\mathbf{P}$ . This tensor is in general nonsymmetric ( $\mathbf{P}\mathbf{F}^T \neq \mathbf{F}\mathbf{P}^T$ ) and expresses the actual stresses in terms of the area of the reference configuration. In fact it is a *two-field tensor*, where one base vector lies on  $\mathcal{B}$  and the other lies on  $\boldsymbol{\varphi}(\mathcal{B})$ .

$$\mathbf{P} = J \boldsymbol{\sigma} \mathbf{F}^{-T} \quad (2.37)$$

There is a symmetric stress tensor which is defined with regard to the reference configuration. This is the second *Piola-Kirchhoff* stress tensor  $\mathbf{S}$ :

$$\mathbf{S} = \mathbf{F}^{-1} \mathbf{P} = J \mathbf{F}^{-1} \boldsymbol{\sigma} \mathbf{F}^{-T} \quad (2.38)$$

$\mathbf{S}$  does not represent an experimentally measurable stress but it is an essential stress measure that plays an important role in the constitutive theory. Firstly it is the complete *pull back* of the *Cauchy* stress tensor to the reference configuration  $\mathcal{B}$ . Secondly, and most important property, it is "work conjugate" (duality paired) with the *Green-Lagrange* strain tensor of equation (2.17).

Instead of the *Cauchy* stress tensor  $\boldsymbol{\sigma}$ , the *Kirchhoff* stress tensor  $\boldsymbol{\tau}$  is often employed. It is defined as the *push forward* of the second *Piola-Kirchhoff* stress tensor  $\mathbf{S}$  to the current configuration:

$$\boldsymbol{\tau} = \mathbf{F} \mathbf{S} \mathbf{F}^T, \quad \boldsymbol{\tau} = J \boldsymbol{\sigma} \quad (2.39)$$

## 2.3 Lagrangian formulation

### 2.3.1 Balance laws

The physical system must satisfy a group of fundamental laws. The equations that describe the mechanical behavior of the continuum arise from these laws. They are called *balance laws* or *conservation laws* and are usually expressed as partial differential equations. The balance laws of a thermomechanical system are considered and summarized in this section.

#### Balance of mass

The mass conservation principle requires that the mass of any material domain remains constant. There is no material flow through the boundaries of a material domain and there is no mass loss to energy conversion. The balance of the mass  $m$  of a body is given by the equation

$$m = \int_{\partial\mathcal{B}} \rho_0 dV = \int_{\partial\varphi(\mathcal{B})} \rho dV = \text{const.} \quad (2.40)$$

where  $\rho_0$  is the density in the initial configuration and  $\rho$  the density in the current configuration. With a *Lagrangian* description and assuming sufficient smoothness, the next equation for mass conservation is obtained:

$$\rho_0 = J \rho \quad (2.41)$$

This equation yields a relation between the volume differentials in the initial and current configurations:

$$dv = \frac{\rho_0}{\rho} dV = J dV \quad (2.42)$$

#### Local balance of momentum and angular momentum

The balance of momentum is the equivalent of Newton's second law of motion which relates the forces acting on a body to its acceleration. The equation of local balance of momentum with respect to a volume in the current configuration  $\varphi(\mathcal{B})$  can be written as [13]

$$\text{div } \boldsymbol{\sigma} + \rho \bar{\mathbf{b}} = \rho \ddot{\mathbf{u}} \quad (2.43)$$

where  $\boldsymbol{\sigma}$  is the *Cauchy* stress tensor (2.35),  $\rho \bar{\mathbf{b}}$  defines the volume or body forces (e.g. gravitational force), and  $\rho \ddot{\mathbf{u}}$  is the inertial force term. For static analysis, where the loads are applied slowly, the inertial forces can be neglected. The resulting equation is called the *equilibrium equation*:

$$\text{div } \boldsymbol{\sigma} + \rho \bar{\mathbf{b}} = 0 \quad (2.44)$$

The local balance of angular momentum in the absence of micropolar stresses, which is usually the case in non-magnetic materials, yields

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^T \quad (2.45)$$

This equation re-expresses the symmetry of the *Cauchy* stress tensor (see equation (2.35)).

### First law of thermodynamics

This law is the principle of conservation of energy for a thermodynamical process. The first law of thermodynamics postulates that the rate of change of total energy  $E$  is induced by the work done by body forces and surface forces, the mechanical power  $P$  and the heat transfer  $Q$  into the system. The statement of the conservation of energy is:

$$\dot{E} = P + Q \quad (2.46)$$

where the mechanical power  $P$  is defined by

$$P = \frac{d}{dt} \int_{\varphi(\mathcal{B})} \frac{1}{2} \rho \mathbf{v} \cdot \mathbf{v} \, dv + \int_{\varphi(\mathcal{B})} \boldsymbol{\sigma} \cdot \mathbf{d} \, dv \quad (2.47)$$

The first integral expresses the material time derivative of the kinetic energy and the second one is the work performed by the *Cauchy* stress tensor over the symmetrical spatial velocity gradient  $\mathbf{d}$  (see Section 2.2.2). The term  $\boldsymbol{\sigma} \cdot \mathbf{d}$  contributes to the internal energy.

The heat transfer input into the system has the expression:

$$Q = - \int_{\varphi(\partial\mathcal{B})} \mathbf{q} \cdot \mathbf{n} \, da + \int_{\varphi(\mathcal{B})} \rho r \, dv \quad (2.48)$$

The integrals represent two sources of energy. The first one is the heat transfer through the surface of the body, described by the heat flux vector  $\mathbf{q}$  and the surface normal  $\mathbf{n}$ . The second integral is the heat induced by the internal heat source  $r$ .

The synthesis of energy balance can be explained as: The total energy consists on the sum of the *kinetic energy*  $K = \int_{\varphi(\mathcal{B})} \frac{1}{2} \rho \mathbf{v} \cdot \mathbf{v} \, dv$  and the *internal energy*  $W = \int_{\varphi(\mathcal{B})} \rho w \, dv$ , where  $w$  is the specific internal energy. Introducing these relations into (2.46) and after some transformations, the local form of the first law of thermodynamics becomes

$$\rho \dot{w} = \boldsymbol{\sigma} \cdot \mathbf{d} + \rho r - \operatorname{div} \mathbf{q} \quad (2.49)$$

If the heat flux  $\mathbf{q}$  and heat source  $r$  vanish (e.g. purely mechanical process) the only term that remains in the left hand side of the equation is  $\boldsymbol{\sigma} \cdot \mathbf{d}$ , which denotes the specific stress power.

$$\rho \dot{w} = \boldsymbol{\sigma} \cdot \mathbf{d} \quad (2.50)$$

### 2.3.2 Weak form of balance of momentum

Variational principles are employed in numerical methods for the solution of boundary value problems. To find a solution for the strong form of balance laws associated formulations are needed. The principle of virtual work yields an equivalent formulation for the balance of momentum, often called weak form of equilibrium. It holds for all problem classes, including plasticity, friction or non-conservative loading. The derivation of the weak form starts from the local balance of the momentum equation (2.43) which is multiplied by a valued function  $\boldsymbol{\eta} = \{\boldsymbol{\eta} | \boldsymbol{\eta} = \mathbf{0} \text{ on } \partial B_u\}$ , often called virtual displacement or test function.

Integrating the balance momentum equation over the volume of the body under consideration and transforming the associated tensors by *push forward* operations to the current configuration gives:

$$\int_{\boldsymbol{\varphi}(\mathcal{B})} \text{div} \boldsymbol{\sigma} \cdot \boldsymbol{\eta} \, dv + \int_{\boldsymbol{\varphi}(\mathcal{B})} \rho (\bar{\mathbf{b}} - \dot{\mathbf{v}}) \cdot \boldsymbol{\eta} \, dv \quad (2.51)$$

The partial integration of the first term of this equation using the divergence theorem gives the spatial form of the weak formulation:

$$g(\boldsymbol{\varphi}, \boldsymbol{\eta}) = \int_{\boldsymbol{\varphi}(\mathcal{B})} \boldsymbol{\sigma} \cdot \text{grad} \boldsymbol{\eta} \, dv - \int_{\boldsymbol{\varphi}(\mathcal{B})} \rho (\bar{\mathbf{b}} - \dot{\mathbf{v}}) \cdot \boldsymbol{\eta} \, dv - \int_{\boldsymbol{\varphi}(\partial \mathcal{B}_\sigma)} \bar{\mathbf{t}} \cdot \boldsymbol{\eta} \, da = 0 \quad (2.52)$$

The symmetry of the *Cauchy* stress tensor enables us to replace the spatial gradient of  $\boldsymbol{\eta}$  by its symmetric part. Defining:

$$\nabla^S \boldsymbol{\eta} = \frac{1}{2} (\text{grad} \boldsymbol{\eta} + \text{grad}^T \boldsymbol{\eta}) \quad (2.53)$$

the expression (2.52) becomes

$$g(\boldsymbol{\varphi}, \boldsymbol{\eta}) = \int_{\boldsymbol{\varphi}(\mathcal{B})} \boldsymbol{\sigma} \cdot \nabla^S \boldsymbol{\eta} \, dv - \int_{\boldsymbol{\varphi}(\mathcal{B})} \rho (\bar{\mathbf{b}} - \dot{\mathbf{v}}) \cdot \boldsymbol{\eta} \, dv - \int_{\boldsymbol{\varphi}(\partial \mathcal{B}_\sigma)} \bar{\mathbf{t}} \cdot \boldsymbol{\eta} \, da = 0 \quad (2.54)$$

This relation has exactly the same structure as the *principle of virtual work* in the geometrically linear theory. However, the difference is that all integrals, stresses and gradients have to be computed with respect to the current configuration (current coordinates). This shows the nonlinearity of the weak form of the balance of momentum.

### 2.3.3 Minimum of total potential energy

There are special cases for which an alternative solution for the boundary value problem exist. That occurs when the elastic energy stored in a body  $\mathcal{B}$  can be described by a strain energy function  $W$ . That happens for hyperelastic materials. They are presented in Section 2.4.2.

Based on this function and considering the potential energy of the applied forces, the classical minimum principle of the total elastic potential can be formulated. The assumption is that the applied forces are conservative (meaning they are path-independent).

Neglecting dynamic effects, the minimum total potential energy principle is written as

$$\Pi(\boldsymbol{\varphi}) = \int_{\boldsymbol{\varphi}(\mathcal{B})} \rho [W(\mathbf{C}) - \bar{\mathbf{b}} \cdot \boldsymbol{\varphi}] dv - \int_{\boldsymbol{\varphi}(\partial\mathcal{B}_\sigma)} \bar{\mathbf{t}} \cdot \boldsymbol{\varphi} da \Rightarrow \text{MIN} \quad (2.55)$$

Considering all possible deformation states, the one which minimizes  $\boldsymbol{\varphi}$  fulfils the equilibrium equations. The minimum can be computed by a variation of (2.55). It is related to the weak form (2.54) and can be derived by applying the directional derivative. This leads to the so-called first variation of  $\Pi$ :

$$\delta\Pi = D\Pi(\boldsymbol{\varphi}) \cdot \boldsymbol{\eta} = \left. \frac{d}{d\alpha} \Pi(\boldsymbol{\varphi} + \alpha\boldsymbol{\eta}) \right|_{\alpha=0} \quad (2.56)$$

In explicit form

$$D\Pi(\boldsymbol{\varphi}) \cdot \boldsymbol{\eta} = \int_{\boldsymbol{\varphi}(\mathcal{B})} \rho \left[ \frac{\partial W(\mathbf{b})}{\partial \mathbf{b}} \cdot \delta \mathbf{b} - \bar{\mathbf{b}} \cdot \boldsymbol{\eta} \right] dv - \int_{\boldsymbol{\varphi}(\partial\mathcal{B}_\sigma)} \bar{\mathbf{t}} \cdot \boldsymbol{\eta} da = g(\boldsymbol{\varphi}, \boldsymbol{\eta}) = 0 \quad (2.57)$$

To see the equivalence between (2.57) and (2.54), the partial derivative of  $W$  with respect to the left *Cauchy-Green* tensor  $\mathbf{b}$  leads to the Cauchy stress tensor  $2\rho \frac{\partial W(\mathbf{b})}{\partial \mathbf{b}} \mathbf{b} = \boldsymbol{\sigma}$ . It is presented in equation (2.84). The variation of the left *Cauchy-Green* tensor  $\mathbf{b}$  can be expressed in terms of the variation of the *Almansi* strain tensor as  $\delta \mathbf{b} \simeq 2\delta \mathbf{e} \mathbf{b} \simeq 2\nabla^S \boldsymbol{\eta} \mathbf{b}$ . This equivalence is not direct but can be deduced from equations (2.18) and (2.33), using (2.167) and (2.168) which are presented in Section 2.5. The variation becomes after a *push forward* operation applied to the right *Cauchy-Green* tensor  $C$ .

The construction of a minimal total potential energy principle is important in several aspects, since it enables investigations regarding the existence and uniqueness of solutions. Furthermore, special solution methods can be developed on the basis of this principle which are efficient and reliable.

## 2.4 Constitutive models

The mathematical characterization of the material behavior is described by constitutive laws. The constitutive equation of a body gives the stress as a function of the deformation history of the body. Different constitutive relations permit to distinguish between a viscous fluid, rubber, metal or concrete, for example. In the following section different classes of material behavior are introduced. The multiaxial constitutive equations for large deformation elasticity, such as the well-know *Neo-Hookean* or *Saint Venant-Kirchhoff* hyperplastic materials, are considered.



### 2.4.1 Nonlinear Elasticity

Elastic materials have the fundamental property that the stress depends only on the current level of strain. At the same time, strains are *reversible*. This means that the loading and unloading stress-strain curves are identical. The original shape of the material is recovered upon unloading the material. An elastic material is also *rate-independent* and there is a one-to-one correspondence between stress and strain. Reversibility and path-independence in a purely mechanical theory imply absence of energy dissipation in deformation. For a *non-dissipative* material all energy expended in deformation is stored in the material and can be recovered upon unloading. For simplicity the problem is presented in one dimension and then is extended to the general case. For a nonlinear elastic material the uniaxial stress-strain law can be written as:

$$\boldsymbol{\sigma}_x = \mathbf{s}(\boldsymbol{\varepsilon}_x) \quad (2.58)$$

Due to the non-dissipative characteristic a *potential* function  $w(\boldsymbol{\varepsilon}_x)$  exist, called the specific strain energy density, such that

$$\boldsymbol{\sigma}_x = \mathbf{s}(\boldsymbol{\varepsilon}_x) = \frac{dw(\boldsymbol{\varepsilon}_x)}{d\boldsymbol{\varepsilon}_x} \quad (2.59)$$

Function  $w(\boldsymbol{\varepsilon}_x)$  can be identified with the specific internal energy density  $w(\boldsymbol{\varepsilon}_x) \equiv \rho w^{int}$  under adiabatic conditions (see equation (2.49)). From (2.59)

$$dw(\boldsymbol{\varepsilon}_x) = \boldsymbol{\sigma}_x d\boldsymbol{\varepsilon}_x = \boldsymbol{\sigma}_x \dot{\boldsymbol{\varepsilon}}_x = \boldsymbol{\sigma}_x d\boldsymbol{x} \quad (2.60)$$

It can be seen that this is the one-dimensional version of the multiaxial expression (2.50).

For a linear elastic material, the stress-strain law can be written as

$$\boldsymbol{\sigma}_x = E \boldsymbol{\varepsilon}_x \quad (2.61)$$

where  $E$  is the *Young modulus*. This relation is the classical *Hooke* law. The specific strain energy density is given by

$$w = \frac{1}{2} E \boldsymbol{\varepsilon}_x^2 \quad (2.62)$$

which is a quadratic function of the strains.

Last expressions hold for small strain regimes. The extension to large strains requires to choose a measure of strain and define an elastic potential for the stress. It means that the stress measure will be *work conjugate* with the strain measure. The existence of a potential implies reversibility, path-independence and non dissipation of the deformation process.

Choosing the *Green-Strain*  $E_x$  in one dimension as the strain measure, the work conjugate second *Piola-Kirchhoff* stress becomes:

$$S_x = \frac{dw}{dE_x} \quad (2.63)$$

The potential function is quadratic in the Green strain, i.e.

$$w = \frac{1}{2} \check{E} E_x^2 \quad (2.64)$$

where the modulus  $\check{E}$  is constant. Clearly  $\check{E} = E$  for small strains. Then the relation between these stress and strain measures is linear.

$$S_x = \check{E} E_x \quad (2.65)$$

All elastic stress-strain relationships in which the stresses can be obtained from a potential function of the strains are called *hyperelastic*. Equation (2.65) is the simplest hyperplastic relation.

#### 2.4.2 Multiaxial nonlinear elasticity: Hyperelastic response

The extension of non-linear elasticity to the three dimensional case is called multiaxial elasticity. There are many different constitutive relations for finite strains. The most straightforward generalization of linear elasticity to large deformations is the *Saint Venant-Kirchhoff* material.

##### Saint Venant-Kirchhoff material

This model is suitable for applications where the materials experience small strains and large rotations. When the effects of large deformations are due to rotations the material can be modeled by a simple extension of the linear elastic laws. The most general *Kirchhoff* material model is

$$S_{AB} = \check{C}_{ABCD} E_{CD}, \quad \mathbf{S} = \check{\mathbf{C}} : \mathbf{E} \quad (2.66)$$

where  $\check{\mathbf{C}}$  is the fourth-order tensor of elastic moduli, which is constant for a *Kirchhoff* material. For an isotropic material, tensor  $\check{\mathbf{C}}$  is also isotropic. This means that it has the same components in any (rectangular Cartesian) coordinate system. The general expression for the material tensor  $\check{\mathbf{C}}$  for an isotropic material is given by

$$\check{C}_{ABCD} = \lambda \delta_{AB} \delta_{CD} + \mu (\delta_{AB} \delta_{CD} + \delta_{AC} \delta_{BD} + \delta_{AD} \delta_{BC}), \quad \check{\mathbf{C}} = \lambda \mathbf{1} \otimes \mathbf{1} + 2\mu \mathbf{I} \quad (2.67)$$

where the two independent constants  $\lambda$  and  $\mu$  are called the *Lame* constants. The fourth-order symmetric identity tensor  $\mathbf{I}$  has the components  $I_{ABCD} = \frac{1}{2}(\delta_{AC} \delta_{BD} + \delta_{AD} \delta_{BC})$

The stress-strain relation in (2.66) can be written now as

$$\mathbf{S} = \lambda (\text{tr } \mathbf{E}) \mathbf{1} + 2\mu \mathbf{E} \quad (2.68)$$

The *Lame* constants can be expressed in terms of other constants, which are more closely related to physical measurements; i.e.

$$G = \mu = \frac{E}{2(1 + \nu)}, \quad \lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)}, \quad K = \lambda + \frac{2\mu}{3} \quad (2.69)$$

where  $G$  is the *shear modulus*,  $K$  is the *bulk modulus*,  $E$  is the *Young modulus* and  $\nu$  is the *Poisson ratio*.

In two-dimensional problems the stress-strain relations have a more particular expression depending on the simplification of the problem to *plane stress* or *plane strain* situations, see [93], [81], [137].

The equivalent definition of (2.66) for the current configuration is obtained by the *push-forward* operation, which gives

$$\tau_{ik} = F_{iA} F_{lC} F_{mD} F_{kB} \check{\mathbf{C}}_{ABCD} e_{lm} \quad (2.70)$$

where  $\tau_{ik}$  is the *Kirchhoff* stress tensor of (2.39) and  $e_{lm}$  the *Almansi* strain tensor of (2.20). We define the spatial incremental constitutive tensor as

$$\check{\mathbf{c}}_{iklm} = F_{iA} F_{lC} F_{mD} F_{kB} \check{\mathbf{C}}_{ABCD} \quad (2.71)$$

yielding

$$\tau_{ik} = \check{\mathbf{c}}_{iklm} e_{lm} \quad \boldsymbol{\tau} = \check{\mathbf{c}} : \mathbf{e} \quad (2.72)$$

### Hyperelastic materials

As previously mentioned *hyperplastic* materials are characterized by the existence of a strain energy function that is a potential of the stress. The constitutive equation or response function for the second *Piola-kirchhoff* stress is the partial derivative of the strain energy  $W$  function with respect to the right *Cauchy-Green* tensor, which was previously introduced in (2.63).

$$\mathbf{S} = 2\rho_0 \frac{\partial W(\mathbf{C}, \mathbf{X})}{\partial \mathbf{C}} \quad (2.73)$$

For a homogeneous isotropic material, the strain energy  $W$  does not depend upon  $\mathbf{X}$ . Restricting the potential to an isotropic material respect to the initial unstressed configuration, the strain energy function can be represented by an isotropic tensor function. It can be written as a function of the principal invariants of the right (or left) *Cauchy-Green* deformation (2.16) (2.19), as

$$W(\mathbf{F}) = W(I_C, II_C, III_C) = W(I_b, II_b, III_b) \quad (2.74)$$

where  $I_C, II_C, III_C$  are defined using the eigenvalues  $\lambda_i^2$  of the right *Cauchy-Green* tensor  $\mathbf{C}$  as

$$\begin{aligned}
I_C &= \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \\
II_C &= \lambda_1^2 \lambda_2^2 + \lambda_2^2 \lambda_3^2 + \lambda_3^2 \lambda_1^2 \\
III_C &= \lambda_1^2 \lambda_2^2 \lambda_3^2
\end{aligned} \tag{2.75}$$

The same applies for the invariants of the left *Cauchy-Green* tensor  $\mathbf{b}$ , considering now  $\lambda_i^2$  as the eigenvalues of  $\mathbf{b}$ .

Using (2.73) and the chain rule, the second *Piola-Kirchhoff* stress tensor obtained as

$$\mathbf{S} = 2 \rho_0 \left[ \left( \frac{\partial W}{\partial I_C} + I_C \frac{\partial W}{\partial II_C} \right) \mathbf{1} - \frac{\partial W}{\partial II_C} \mathbf{C} + III_C \frac{\partial W}{\partial III_C} \mathbf{C}^{-1} \right] \tag{2.76}$$

where the derivative of the invariants respect to the tensors has been used; i.e.

$$\left( \frac{\partial I_C}{\partial \mathbf{C}} = \mathbf{1}, \quad \frac{\partial II_C}{\partial \mathbf{C}} = I_C \mathbf{1} - \mathbf{C}, \quad \frac{\partial III_C}{\partial \mathbf{C}} = III_C \mathbf{C}^{-1} \right) \tag{2.77}$$

From the definition (2.39) the *Cauchy* stress tensor can be written in terms of variables defined in the initial configuration as:

$$\boldsymbol{\sigma} = 2 \rho \mathbf{F} \frac{\partial W(\mathbf{C})}{\partial \mathbf{C}} \mathbf{F}^T \tag{2.78}$$

From this form, the *Cauchy* stress is given using (2.73) with (2.76) by

$$\boldsymbol{\sigma} = 2 \rho \left[ \left( \frac{\partial W}{\partial I_C} + I_C \frac{\partial W}{\partial II_C} \right) \mathbf{F} \mathbf{F}^T - \frac{\partial W}{\partial II_C} \mathbf{F} \mathbf{C} \mathbf{F}^T + III_C \frac{\partial W}{\partial III_C} \mathbf{F} \mathbf{C}^{-1} \mathbf{F}^T \right] \tag{2.79}$$

Since the invariants of  $\mathbf{C}$  and  $\mathbf{b}$  coincide, using  $\mathbf{F} \mathbf{C}^{-1} \mathbf{F}^T = \mathbf{1}$ , they can be written as

$$I_C = \text{tr } \mathbf{C} = \text{tr } \mathbf{b} \tag{2.80}$$

$$II_C = \frac{1}{2} [(\text{tr } \mathbf{C})^2 - \text{tr } \mathbf{C}^2] = \frac{1}{2} [(\text{tr } \mathbf{b})^2 - \text{tr } \mathbf{b}^2] \tag{2.81}$$

$$III_C = \det \mathbf{C} = \det \mathbf{b} \tag{2.82}$$

Equation (2.79) is modified as:

$$\boldsymbol{\sigma} = 2 \rho \left[ \left( \frac{\partial W}{\partial I_b} + I_b \frac{\partial W}{\partial II_b} \right) \mathbf{b} - \frac{\partial W}{\partial II_b} \mathbf{b}^2 + III_b \frac{\partial W}{\partial III_b} \mathbf{1} \right] \tag{2.83}$$

Comparing (2.83) with (2.79) we can see that the *Cauchy* stress tensor can be written in terms of the left *Cauchy-Green* tensor as

$$\boldsymbol{\sigma} = 2 \rho \frac{\partial W(\mathbf{b})}{\partial \mathbf{b}} \mathbf{b} \quad (2.84)$$

The most simple *Neo-Hookean* material law consists of two parts: one depends on the first invariant of the right (or left) *Cauchy-Green* tensor and the second one depends on the *Jacobi* determinant, i.e.

$$W(I_b, J) = g(J) + \frac{1}{2} \mu (I_b - 3) \quad (2.85)$$

Function  $g(J)$  has to be convex for compressible materials. Different approaches for the strain energy function  $W$  are possible as long as they fulfill poly-convexity and the three growth conditions for a compressible hyperplastic material:

$$\lim_{J \rightarrow +\infty} W \rightarrow \infty \quad , \quad \lim_{J \rightarrow 0} W \rightarrow -\infty \quad , \quad W(J = 1) = 0 \quad (2.86)$$

For the compressible part of  $g(J)$  we have

$$g(J) = c(J^2 - 1) - d \ln J - \mu \ln J \quad \text{with} \quad c > 0, \quad d > 0 \quad (2.87)$$

The following are two examples of strain energy functions:

$$W(I_b, J) = \frac{1}{2} \mu (I_b - 3) + \frac{1}{4} \lambda (J^2 - 1) - \frac{1}{4} \lambda \ln J - \mu \ln J \quad (2.88)$$

with  $c = \frac{\lambda}{4}$  and  $d = \frac{\lambda}{2}$ , or

$$W(I_b, J) = \frac{1}{2} \mu (I_b - 3) + \frac{1}{2} \lambda (\ln J)^2 - \mu \ln J \quad (2.89)$$

Equation (2.89) comes from another expression of  $g(J)$ . In above equations  $\mu$  and  $\lambda$  are the *Lame* constants (2.69).

The *Cauchy* stress tensor associated to (2.88) and (2.89) is developed from (2.84), being

$$\boldsymbol{\sigma} = \frac{\lambda}{2} (J^2 - 1) \mathbf{1} + \frac{\mu}{J} (\mathbf{b} - \mathbf{1}) \quad (2.90)$$

the response function associated to the strain energy function (2.88) and

$$\boldsymbol{\sigma} = \frac{\lambda}{J} \ln J \mathbf{1} + \frac{\mu}{J} (\mathbf{b} - \mathbf{1}) \quad (2.91)$$

the response function associated to (2.89).

### 2.4.3 Incremental constitutive tensor

Taking in account that the constitutive laws will be linearized and integrated in time, the rate of stresses is needed. This is obtained from the differentiation with respect to time of the 2nd *Piola-Kirchhoff* stress tensor function (2.73), i.e.

$$\dot{\mathbf{S}} = \frac{\partial S(\mathbf{X}, t)}{\partial t} = 2 \frac{\partial^2 W(\mathbf{C})}{\partial \mathbf{C} \partial \mathbf{C}} [\dot{\mathbf{C}}] \quad (2.92)$$

or the analogue form with the *Cauchy* stress tensor (2.84)

$$\dot{\boldsymbol{\sigma}} = \frac{\partial \boldsymbol{\sigma}}{\partial t} + \text{grad } \boldsymbol{\sigma} \cdot \mathbf{v} \quad (2.93)$$

where the two components of the material derivative can be seen, as described in (2.7). The convective part of the tensor can be eliminated by using the derivative in the velocity direction. This is the *Lie-derivative* of the covariant tensor and is applied to the *Kirchhoff* stress tensor as

$$\mathcal{L}_v \boldsymbol{\tau} = \mathbf{F} \left\{ \frac{\partial}{\partial t} [\mathbf{F}^{-1} \boldsymbol{\tau} \mathbf{F}^{-T}] \right\} \mathbf{F}^T = \mathbf{F} \dot{\mathbf{S}} \mathbf{F}^T \quad (2.94)$$

The incremental constitutive tensor is defined as the incremental relation between the rate of the 2nd *Piola-Kirchhoff* stress tensor  $\mathbf{S}$  and the right *Cauchy-Green* tensor  $\mathbf{C}$ . This tensor is defined by

$$\mathbf{C} = 4 \frac{\partial^2 W}{\partial \mathbf{C} \partial \mathbf{C}}, \quad \mathbf{C}_{ABCD} = 4 \frac{\partial^2 W}{\partial C_{AB} \partial C_{CD}} \quad (2.95)$$

Then  $\dot{\mathbf{S}}$  can be obtained as

$$\dot{\mathbf{S}} = \mathbf{C} \left[ \frac{1}{2} \dot{\mathbf{C}} \right], \quad \dot{S}_{AB} = \mathbf{C}_{ABCD} \left[ \frac{1}{2} \dot{C}_{CD} \right] \quad (2.96)$$

From (2.17) and (2.31) the next equality is defined:

$$\dot{\mathbf{C}} = 2 \dot{\mathbf{E}} = 2 \mathbf{F}^T \mathbf{d} \mathbf{F}, \quad \dot{C}_{CD} = 2 F_{lC} d_{lm} F_{mD} \quad (2.97)$$

Doing the *push-forward* of (2.96) the equivalent definition in the reference configuration is obtained. Operation with equations (2.94) and (2.97) yields

$$(\mathcal{L}_v \boldsymbol{\tau})_{ik} = F_{iA} F_{lC} F_{mD} F_{kB} \mathbf{C}_{ABCD} d_{lm} \quad (2.98)$$

where  $\mathbf{d}$  is the symmetrical spatial velocity gradient. Defining the spatial incremental constitutive tensor as

$$\mathbf{c}_{iklm} = F_{iA} F_{lC} F_{mD} F_{kB} \mathbf{C}_{ABCD} \quad (2.99)$$

(2.98) can be written in a more compact form as

$$(\mathcal{L}_v \boldsymbol{\tau})_{ik} = \mathbf{c}_{iklm} d_{lm}, \quad \mathcal{L}_v \boldsymbol{\tau} = \mathbf{c} [\mathbf{d}] \quad (2.100)$$

Comparing (2.33) and (2.31) a similar expression can be written as

$$(\mathcal{L}_v \boldsymbol{\tau})_{ik} = \mathbf{c}_{iklm} (\mathcal{L}_v e)_{lm}, \quad \mathcal{L}_v \boldsymbol{\tau} = \mathbf{c} \mathcal{L}_v \mathbf{e} \quad (2.101)$$

The fourth order incremental constitutive tensor (2.99) in the current configuration  $\varphi(\mathcal{B})$ , considering a *Neo-Hookean* law (2.90) is given after some algebra by:

$$\mathbf{c} = \lambda J^2 \mathbf{1} \otimes \mathbf{1} + [2\mu - \lambda(J^2 - 1)] \mathbb{I} \quad (2.102)$$

where  $\mathbf{1}$  is the second order unit tensor and  $\mathbb{I}$  is the fourth order unit tensor. Both tensors are expressed in the current configuration.

In next sections the tangent constitutive tensors for each specific constitutive law will be presented. They will be needed later when the linearization and the integration of the constitutive laws are performed in order to express relationship between the increment of stresses and strains.

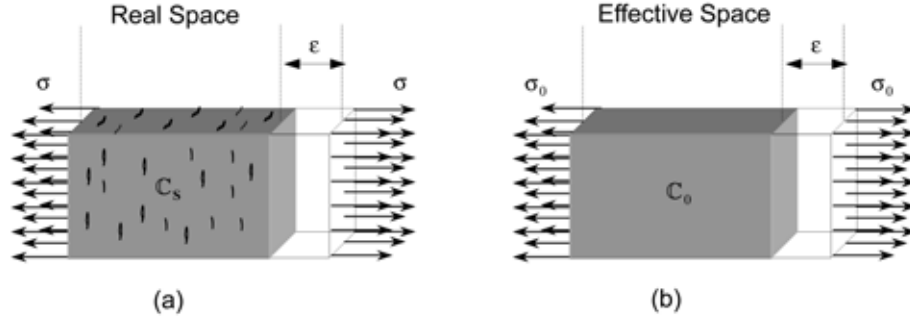
#### 2.4.4 Isotropic damage models

The theory of continuum damage was presented first in 1958 for *Kachanov* [61]. Damage affects only to the elastic properties of the material, experimenting a loss of stiffness after overpassing a damage threshold. There are several formulations to describe the behavior of damaged materials. The one selected here is a model with only a scalar internal variable that characterizes the loss of effective area. This simple model has a big potential for representing the non-linear behavior of metallic materials and geomaterials [98].

The first important concept is the effective stress  $\bar{\boldsymbol{\tau}}$ . It represents the stresses in a non-degraded domain. For an isotropic material it is expressed as:

$$\boldsymbol{\tau} = (1 - d) \bar{\boldsymbol{\tau}} \quad (2.103)$$

where  $d$  is the internal damage variable and  $\boldsymbol{\tau}$  is the *Kirchhoff* stress tensor given in (2.39). It measures the loss of material stiffness, ( $0 \leq d \leq 1$ ). When  $d = 1$  the domain is completely degraded and when  $d = 0$  the domain is not damaged. The hypothesis of effective stress was formulated by means of the strain equivalence. The strain related to a damaged domain under an applied stress  $\boldsymbol{\tau}$  is equivalent to the strain related with the not-damaged domain under an applied effective stress  $\bar{\boldsymbol{\tau}}$ . Figure 2.3 illustrates this hypothesis.



**Figure 2.3:** (a) The real damaged solid and (b) the equivalent non-damaged solid [98]

### Helmholtz free energy

The density of *Helmholtz* free energy per unit volume of a damage model for an isothermal process is written as

$$\Psi(\boldsymbol{\varepsilon}, d) = (1 - d) \bar{\Psi}(\boldsymbol{\varepsilon}) \quad (2.104)$$

where  $\bar{\Psi}(\boldsymbol{\varepsilon})$  is the classical *Helmholtz* free energy for the non-damaged material. This energy can be written in terms of the invariants of the right *Cauchy-Green* tensor  $\mathbf{C}$  (2.16) or the left *Cauchy-Green* tensor  $\mathbf{b}$  (2.19). Therefore  $\bar{\Psi}(\boldsymbol{\varepsilon})$  is an analogous expression of the strain energy  $W$  for an hyperelastic material (see equation (2.74)).

For a *Saint Venant-Kirchhoff* material the classical *Helmholtz* free energy for the non-damaged material  $\bar{\Psi}(\boldsymbol{\varepsilon})$  can be characterized using a quadratic function of the strains  $\bar{\Psi}(\mathbf{e})$ . In that case, the expression is the following

$$\bar{\Psi}(\mathbf{e}) = \frac{1}{2} \mathbf{e} : \check{\mathbf{c}} : \mathbf{e} \quad (2.105)$$

Note that  $\Psi(\mathbf{e}, d)$  is expressed in the current configuration in an isotropic space without plastic strains.  $\check{\mathbf{c}}$  is the spatial elastic constitutive tensor of the non-damaged *Saint Venant-Kirchhoff* material (2.71) and  $\mathbf{e}$  is the almanzi strain tensor (2.18). For stable thermal problems the mechanical dissipation of the energy is defined by the *Clausius-Plank* inequality:

$$\Xi = \left( \boldsymbol{\tau} - \frac{\partial \Psi}{\partial \mathbf{e}} \right) : \dot{\mathbf{e}} - \frac{\partial \Psi}{\partial d} \dot{d} \geq 0 \quad (2.106)$$

After setting the *Coleman* conditions [98] which are sufficient to guarantee a non-negative dissipation  $\Xi \geq 0$ , the constitutive equation for the solid domain (2.72) becomes

$$\boldsymbol{\tau} = \frac{\partial \Psi}{\partial \mathbf{e}} = (1 - d) \check{\mathbf{c}} : \mathbf{e} \quad (2.107)$$

The parte of the energy dissipation is defined as



$$\frac{\partial \Psi}{\partial d} = -\bar{\Psi} \leq 0 \quad \Rightarrow \quad -\bar{\Psi} \text{ is } d\text{-conjugated} \quad (2.108)$$

Equation (2.107) defines an isotropic damage model, affected by a scalar variable only, which can be decomposed by the sum of the elastic and inelastic parts as:

$$\boldsymbol{\tau} = (1 - d) \check{\boldsymbol{\epsilon}} : \mathbf{e} = [\check{\boldsymbol{\epsilon}} : \mathbf{e}] - [d \check{\boldsymbol{\epsilon}} : \mathbf{e}] = \bar{\boldsymbol{\tau}} - \boldsymbol{\tau}_d \quad (2.109)$$

For a more complex free energy function the secant constitutive equation has a general form as

$$\boldsymbol{\tau} = (1 - d) \frac{\partial \bar{\Psi}(\mathbf{b})}{\partial \mathbf{b}} \quad (2.110)$$

The resolution of  $\frac{\partial \bar{\Psi}(\mathbf{C})}{\partial \mathbf{C}}$  was developed in Section 2.4.2 for hyperelastic materials. That is analogous to the resolution of  $\frac{\partial \bar{\Psi}(\mathbf{b})}{\partial \mathbf{b}}$  (see equation (2.84)). The resulting secant constitutive law from equation (2.90) is

$$\boldsymbol{\tau} = (1 - d) \left[ J \frac{\lambda}{2} (J^2 - 1) \mathbf{1} + \mu (\mathbf{b} - \mathbf{1}) \right] \quad (2.111)$$

We note that in our work the integration of the constitutive model is explicit, hence the evolution of the damage variable  $d$  must be known during all the mechanical process. The damage threshold is defined by means of a homogeneous scalar function which is first order in stresses and has tensorial arguments.

### Damage threshold

The criterium to define the damage threshold has to distinguish states of elastic and degraded behavior. This criterium depends on the material and can be defined with flow functions as in plasticity [120].

$$\mathbb{F}(\bar{\boldsymbol{\tau}}, \mathbf{q}) = f(\bar{\boldsymbol{\tau}}) - c(d) \leq 0, \quad \mathbf{q} \equiv d \quad (2.112)$$

where  $f(\bar{\boldsymbol{\tau}})$  is a function of the stress tensor  $\bar{\boldsymbol{\tau}} = \check{\boldsymbol{\epsilon}} : \mathbf{e}$  and  $c(d)$  is the function that defines the damage threshold.

The initial non-linear behavior is established by  $c(d)$ , which also defines loading, unloading and reloading states. It is a scalar function that is positive in the strained state and null in the unstrained one. The initial value of the damage threshold  $c(d_0) = c^{\max} = \boldsymbol{\tau}^{\max}$  is a material property and usually is related to the compressive stress.

Equation (2.112) is the limit surface in the non-damaged strain or stress spaces. By means of  $\mathbb{F}(\bar{\boldsymbol{\tau}}, \mathbf{q})$  the evolution of the damage internal variable is described. Damage occurs when the value of  $f(\bar{\boldsymbol{\tau}})$  is equal or greater than  $c^{\max} = \boldsymbol{\tau}^{\max}$ . An equivalent expression of (2.112) is

$$\mathbb{F}(\bar{\boldsymbol{\tau}}, \mathbf{q}) = G[f(\bar{\boldsymbol{\tau}})] - G[c(d)] \leq 0, \quad \mathbf{q} \equiv d \quad (2.113)$$

where  $G[\cdot]$  is a scalar function that is positive, invertible and its derivative is monotonic and increasing. Function  $G[\cdot]$  is used to define the evolution of the damage threshold function  $c(d)$ .

### Internal variable evolution

In a mechanical problem the interval variables are defined by an evolution law. For the damage model we have:

$$\dot{d} = \dot{\mu} \frac{\partial \mathbb{F}(\bar{\boldsymbol{\tau}}, \mathbf{q})}{\partial [f(\bar{\boldsymbol{\tau}})]} \equiv \dot{\mu} \frac{\partial G[f(\bar{\boldsymbol{\tau}})]}{\partial [f(\bar{\boldsymbol{\tau}})]} \quad (2.114)$$

where  $\mu$  is a non-negative scalar called damage consistency parameter used to define the loading, unloading and reloading through the *Kuhn-Tucker* conditions:

$$\begin{cases} \dot{\mu} \geq 0 \\ \mathbb{F}(\bar{\boldsymbol{\tau}}, \mathbf{q}) \leq 0 \\ \dot{\mu} \cdot \mathbb{F}(\bar{\boldsymbol{\tau}}, \mathbf{q}) = 0 \end{cases} \quad (2.115)$$

The *Kuhn-Tucker* conditions must be verified to describe the unilateral constrains of the evolution law (2.114). For example, if  $\mathbb{F}(\bar{\boldsymbol{\tau}}, \mathbf{q}) < 0$  the fulfilment of the *Kuhn-Tucker* conditions repairs  $\dot{\mu} = 0$ . Replacing this parameter into equation (2.114) the temporal damage evolution becomes  $\dot{d} = 0$ . That means that the material is not experiencing any damage and is under an elastic mechanical process.

Imposing the consistency conditions and using the properties of  $G[\cdot]$  gives

$$\mathbb{F}(\bar{\boldsymbol{\tau}}, \mathbf{q}) = 0 \Rightarrow G[f(\bar{\boldsymbol{\tau}})] = G[c(d)] \Rightarrow f(\bar{\boldsymbol{\tau}}) = c(d) \Rightarrow \frac{\partial G[f(\bar{\boldsymbol{\tau}})]}{\partial [f(\bar{\boldsymbol{\tau}})]} = \frac{\partial G[c(d)]}{\partial [c(d)]} \quad (2.116)$$

From the condition of permanence on the damage surface we can deduce

$$\mathbb{F}(\bar{\boldsymbol{\tau}}, \mathbf{q}) = 0 \Rightarrow \frac{\partial G[f(\bar{\boldsymbol{\tau}})]}{\partial [f(\bar{\boldsymbol{\tau}})]} \dot{f}(\bar{\boldsymbol{\tau}}) - \frac{\partial G[c(d)]}{\partial c(d)} \dot{c}(d) = 0 \Rightarrow \dot{f}(\bar{\boldsymbol{\tau}}) = \dot{c}(d) \quad (2.117)$$

Observing the temporal derivative  $\partial G[f(\bar{\boldsymbol{\tau}})]/\partial t = \dot{G}[f(\bar{\boldsymbol{\tau}})]$  and doing an analogy with the damage evolution  $\dot{d}$  (2.114), the damage consistency parameter becomes

$$\left. \begin{aligned} \dot{G}[f(\bar{\boldsymbol{\tau}})] &= \frac{\partial G[f(\bar{\boldsymbol{\tau}})]}{\partial [f(\bar{\boldsymbol{\tau}})]} \dot{f}(\bar{\boldsymbol{\tau}}) \\ \dot{d} &= \dot{\mu} \frac{\partial G[f(\bar{\boldsymbol{\tau}})]}{\partial [f(\bar{\boldsymbol{\tau}})]} \end{aligned} \right\} \Rightarrow \dot{d} \equiv \dot{G}[f(\bar{\boldsymbol{\tau}})] \Rightarrow \dot{\mu} \equiv \dot{f}(\bar{\boldsymbol{\tau}}) \quad (2.118)$$

Developing  $\dot{\mu}$  gives

$$\dot{\mu} = \dot{f}(\bar{\boldsymbol{\tau}}) = \dot{c}(d) = \frac{\partial f(\bar{\boldsymbol{\tau}})}{\partial \bar{\boldsymbol{\tau}}} : \dot{\bar{\boldsymbol{\tau}}} = \frac{\partial f(\bar{\boldsymbol{\tau}})}{\partial \bar{\boldsymbol{\tau}}} : \boldsymbol{\epsilon} : \dot{\boldsymbol{\epsilon}} \quad (2.119)$$

From the last definitions the damage threshold  $c$  is deduced as

$$c = \max \{ c^{\max}, \max\{f(\bar{\tau})|_s\} \} \quad \forall 0 \leq s \leq t \quad (2.120)$$

Integrating the time derivative of the damage variable of equation (2.118), the explicit form of  $d$  representing damage at a point of the domain is obtained as

$$d = \int_t \dot{d} dt = \int_t \dot{G}[f(\bar{\tau})] dt = G[f(\bar{\tau})] \quad (2.121)$$

The definition of damage depends on the evolution of function  $G$ . In [98] two different softening models are proposed, which are defined by different  $G$  functions.

### Softening models

Function  $G[\cdot]$  that defines the evolution of the damage has to be monotonic and bounded between 0 and 1. Depending on that function the model is going to represent hardening or softening of the material. These phenomena can also have different evolutions (usually  $G$  varies lineally or exponentially).

The evolution of the damage parameter, in case of considering *exponential softening*, is defined by the following expression:

$$d = G[c(d)] = 1 - \frac{c^{\max}}{c(d)} e^{-A \left(1 - \frac{c(d)}{c^{\max}}\right)} \quad \text{with} \quad 0 \leq c^{\max} \leq c(d) \quad (2.122)$$

Parameter  $A$  depends on the fracture energy of the material and is defined in [95]. The deduction of  $A$  depends on the function that defines the threshold for the damage model. The expression (2.122) was first proposed in [95].

For the *linear softening* case the following scalar function is defined:

$$d = G[c(d)] = \frac{1 - \frac{c^{\max}}{c(d)}}{1 + A} \quad \text{with} \quad 0 \leq c^{\max} \leq c(d) \quad (2.123)$$

note that  $G[c^{\max}] = 0$  and  $G[\infty] = 0$

### Hardening models

In the case of hardening behavior of the material,  $G[\cdot]$  can also be represented by a linear or an exponential function. A new internal variable  $q(c)$  is introduced to characterize hardening. The damage evolution function can be rewritten as

$$d = G[c(d)] = 1 - \frac{q(c)}{c} \quad (2.124)$$

Where  $q(c)$  is the hardening function that can be defined for *exponential hardening* as follows:

$$q(c) = q_{\infty} - (q_{\infty} - c^{\max}) e^{-A \left(1 - \frac{c(d)}{c^{\max}}\right)} \quad \text{with} \quad c^{\max} \leq q_{\infty} \quad (2.125)$$

For the *linear hardening* the following scalar function is proposed:

$$q(c) = \begin{cases} c^{\max} + H(c - c^{\max}) & \text{with } c \in [c^{\max}, c^{\lim}] \\ q_{\infty} & \text{with } c \geq [c^{\lim}] \end{cases}$$

for  $H > 0$  and  $c^{\max} \leq q_{\infty}$  (2.126)

where  $c^{\lim}$  is the limit of the damage threshold. The slope defined by  $H$  ends when  $c^{\lim}$  is reached.  $q_{\infty}$  is defined as the hardening asymptotic limit when  $c(d)$  tends to infinity:

$$q(c)|_{t=0} = c^{\max}$$

$$q(c)|_{t=\infty} = \begin{cases} q_{\infty} & \text{hardening} \\ 0 & \text{softening} \end{cases}$$
(2.127)

Note that in last equation both softening and hardening evolution are considered. Depending on the slope defined by  $H = \partial q / \partial r$  the laws will describe a hardening behavior ( $H > 0$  and  $q_{\infty} > c$ ) or a softening behavior ( $H < 0$  and  $q_{\infty} < c$ ). This is illustrated in Figure 2.4.

In the case of *exponential hardening* the tendency of  $q(c)$  is defined by an equivalent hardening slope  $H$ :

$$\frac{\partial q(c)}{\partial c} = H(c) = A \frac{q_{\infty} - c^{\max}}{c^{\max}} e^{-A \left(1 - \frac{c(d)}{c^{\max}}\right)}$$
(2.128)

### Tangent constitutive tensor for damage

The tangent constitutive tensor for damage is obtained considering the time derivative of the secant constitutive equation (2.109).

$$\dot{\boldsymbol{\tau}} = (1 - d) \check{\boldsymbol{\epsilon}} : \dot{\boldsymbol{\epsilon}} - \dot{d} \check{\boldsymbol{\epsilon}} : \mathbf{e}$$
(2.129)

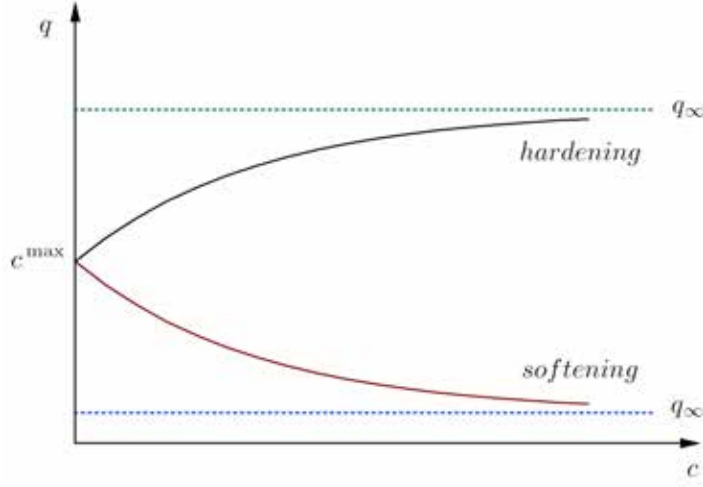
Substituting the equation of the damage evolution (2.114) into (2.129) gives

$$\dot{\boldsymbol{\tau}} = (1 - d) \check{\boldsymbol{\epsilon}} : \dot{\boldsymbol{\epsilon}} - \frac{\partial G[f(\bar{\boldsymbol{\tau}})]}{\partial [f(\bar{\boldsymbol{\tau}})]} \dot{f}(\bar{\boldsymbol{\tau}}) \cdot [\check{\boldsymbol{\epsilon}} : \mathbf{e}]$$
(2.130)

The time derivative of the threshold function is given by

$$\dot{f}(\bar{\boldsymbol{\tau}}) = \frac{\partial f(\bar{\boldsymbol{\tau}})}{\partial \bar{\boldsymbol{\tau}}} : \dot{\bar{\boldsymbol{\tau}}} = \frac{\partial f(\check{\boldsymbol{\epsilon}} : \dot{\boldsymbol{\epsilon}})}{\partial \mathbf{e}} : \dot{\boldsymbol{\epsilon}}$$
(2.131)

Replacing (2.131) into (2.130) yields to



**Figure 2.4:** Exponential hardening and softening functions.

$$\dot{\tau} = (1 - d) \check{\mathbf{c}} : \dot{\mathbf{e}} - \frac{\partial G[f(\bar{\tau})]}{\partial [f(\bar{\tau})]} \left[ \frac{\partial f(\check{\mathbf{c}} : \dot{\mathbf{e}})}{\partial \mathbf{e}} : \dot{\mathbf{e}} \right] \cdot [\check{\mathbf{c}} : \mathbf{e}] \quad (2.132)$$

Finally from (2.132) the tangent constitutive tensor is obtained

$$\mathbf{c} = (1 - d) \check{\mathbf{c}} - \frac{\partial G[f(\bar{\tau})]}{\partial [f(\bar{\tau})]} [\check{\mathbf{c}} : \mathbf{e}] \otimes \left[ \frac{\partial f(\check{\mathbf{c}} : \mathbf{e})}{\partial \mathbf{e}} \right] \quad (2.133)$$

where  $\mathbf{c}$  is the fourth order incremental constitutive tensor for the damage model defined in the current configuration  $\varphi(B)$ .

#### 2.4.5 Damage surfaces

The flow function that defines the damage threshold of the damage model (2.112) is a homogenous scalar function with tensorial arguments and first degree in stresses. That is the case for many flow functions usually considered in plasticity. The model used in this work is based in the presumption that any flow function with those characteristics can define the damage threshold.

#### Simo and Ju model

The *Simo and Ju* damage model was formulated in 1987 and the detailed characteristics can be found in [118] and [119]. It is a simple model founded in the stress space giving a damage threshold in terms of stresses as follows

$$\zeta = f(\bar{\tau}) = \sqrt{2\bar{\Psi}(\mathbf{e})} = \sqrt{\mathbf{e} : \check{\mathbf{c}} : \mathbf{e}} \quad (2.134)$$

The tangent constitutive tensor that results from (2.133) is

$$\mathbf{c} = (1 - d) \check{\mathbf{c}} - \frac{\partial G[\zeta]}{\partial \zeta} \frac{1}{\zeta} [\check{\mathbf{c}} : \mathbf{e}] \otimes [\check{\mathbf{c}} : \mathbf{e}] \quad (2.135)$$

Deducing the parameter  $A$  for the softening laws of equations (2.122) and (2.123) and substituting on the *Simo and Ju* model, the resulting tangent constitutive tensors are

$$\mathbf{c} = (1 - d) \check{\mathbf{c}} - \frac{\bar{\zeta} + A\zeta}{\zeta^2} e^A \left(1 - \frac{\zeta}{\bar{\zeta}}\right) \left[\frac{1}{\zeta} [\check{\mathbf{c}} : \mathbf{e}]\right] \otimes [\check{\mathbf{c}} : \mathbf{e}] \quad \textit{Exponential softening} \quad (2.136)$$

$$\mathbf{c} = (1 - d) \check{\mathbf{c}} - \frac{\bar{\zeta}}{\zeta^2(1 + A)} \left[\frac{1}{\zeta} [\check{\mathbf{c}} : \mathbf{e}]\right] \otimes [\check{\mathbf{c}} : \mathbf{e}] \quad \textit{Linear softening} \quad (2.137)$$

where  $\bar{\zeta} = \bar{f}(\bar{\boldsymbol{\tau}})$  is the initial value for the damage criteria and  $\zeta \geq \bar{\zeta}$ . The process for obtaining (2.136) and (2.137) can be found in [98].

There is an evolution of the *Simo and Ju* model that takes in account that some materials have a different degradation in tension and compression. In this model the threshold function is expressed as

$$\zeta = f(\bar{\boldsymbol{\tau}}) = \left(\theta + \frac{1 - \theta}{n}\right) \sqrt{2\bar{\Psi}(\mathbf{e})} = \left(\theta + \frac{1 - \theta}{n}\right) \sqrt{\mathbf{e} : \check{\mathbf{c}} : \mathbf{e}} \quad (2.138)$$

where  $n$  is the ratio between the compressive strength  $f'_c$  and the tensile strength  $f'_t$

$$n = \frac{f'_c}{f'_t} \quad (2.139)$$

and  $\theta$  is a weighting factor depending on the state of stress  $\bar{\boldsymbol{\tau}}$ . An acceptable definition for  $\theta$  is

$$\theta = \frac{\sum_{i=1}^3 \langle \bar{\tau}_i \rangle}{\sum_{i=1}^3 \|\bar{\tau}_i\|} \quad (2.140)$$

The parameter  $A$  for the softening laws (2.122) and (2.123) deduced for this model is

$$A = \left(\frac{G_f}{l_e(\zeta_0)^2} - \frac{1}{2}\right)^{-1} \geq 0 \quad (2.141)$$

where  $G_f$  is the specific fracture energy per unit area,  $l_e$  is the characteristic length for the fractured domain, usually taken as characteristic length of the finite elements [92] and  $\zeta_0$  is the initial value for the damage criteria. For the *Simo and Ju* model

$$\zeta_0 = f(\bar{\boldsymbol{\tau}}_0) = \frac{f'_t}{\sqrt{E}} \quad (2.142)$$

where  $E$  is the elasticity modulus of the material.

Other parameters used in this model and its implementation are explained in [95].

### Von Mises model

This model was formulated by *Von Mises* in 1913 and it also depends on a single parameter. This is the maximum strength of the octahedric shear  $\tau_{oct}^{max}$ . The damage threshold is reached when the hardening function  $k(d) = \tau_{oct}^{max}(d)$  takes the value of the maximum shear strength  $\tau_{oct}^{max}$ .

This model considers only the second invariant of the deviatoric stress tensor  $II'_\tau = II_{\tau^{dev}} = II_{\tau'}$  without considering the influence of the first invariant of the stress tensor  $I_\tau$  and the third invariant of the deviatoric stress tensor  $III'_\tau$ . The stress invariants are

$$\begin{aligned} I_\tau &= \tau_{ii} \\ II_\tau &= \frac{1}{2}\tau_{ij}\tau_{ji} \\ III_\tau &= \frac{1}{3}\tau_{ij}\tau_{jk}\tau_{ki} \end{aligned} \quad (2.143)$$

For the deviatoric stress tensor  $\tau^{dev}$

$$\begin{aligned} I'_\tau &= I_{\tau'} = \tau'_{ii} = 0 \\ II'_\tau &= II_{\tau'} = \frac{1}{2}\tau'_{ij}\tau'_{ji} \\ III'_\tau &= III_{\tau'} = \frac{1}{3}\tau'_{ij}\tau'_{jk}\tau'_{ki} \end{aligned} \quad (2.144)$$

Mathematically the *Von Mises* criterion can be expressed in terms of principal stresses as

$$\mathbb{F}(\boldsymbol{\tau}, \tau_{oct}^{max}) = \frac{1}{6} [(\tau_1 - \tau_2)^2 + (\tau_2 - \tau_3)^2 + (\tau_3 - \tau_1)^2] - [\tau_{oct}^{max}(d)]^2 = 0 \quad (2.145)$$

where  $\tau_1$  and  $\tau_3$  are the largest and smallest principal stresses, respectively. The same expression (2.145) expressed in function of the second invariant of the deviatoric stress tensor is given by

$$\mathbb{F}(II'_\tau, \tau_{oct}^{max}) = \underbrace{f(II'_\tau)}_{\zeta} - \bar{\tau}(d) = \sqrt{3 III'_\tau} - \bar{\tau}(d) = 0 \quad (2.146)$$

This model is suitable to describe the damage threshold for homogenous materials like metals.

### Mohr-Coulomb model

This model is suitable for establishing the limit where damage starts in frictional materials and geomaterials. The basic principles were formulated by *Coulomb* in 1773 and developed later by *Mohr* in 1882. This model depends on two material parameters, the cohesion  $c$  and the angle of internal friction between the particles of the material  $\phi$ . The strength of a point grows with the friction between particles. Formulated in principal stresses gives

$$\mathbb{F}(\boldsymbol{\tau}, c, \phi) = \underbrace{f(\boldsymbol{\tau}, \phi)}_{\zeta} - c(d) \cos \phi = \left( \frac{\tau_1 - \tau_3}{2} \right) + \left( \frac{\tau_1 + \tau_3}{2} \right) \sin \phi - c(d) \cos \phi = 0 \quad (2.147)$$

From last equation it can be seen that the middle principal stress is not considered in the *Mohr-Coulomb* model. This is a drawback that can be overcome by expressing (2.147) in terms of the invariants of the stress and the deviatoric stress tensors. The resultant expression is

$$\begin{aligned} \mathbb{F}(I_\tau, II'_\tau, \theta, c, \phi) &= \underbrace{f(I_\tau, II'_\tau, \theta, \phi)}_{\zeta} - c(d) \\ &= \frac{I_\tau}{3\sqrt{6}} \tan \phi + \frac{\sqrt{II'_\tau}}{\sqrt{6} \cos \phi} \left( \cos \theta - \frac{\sin \theta \sin \phi}{\sqrt{3}} \right) - c(d) = 0 \end{aligned} \quad (2.148)$$

where  $\theta$  is defined as the *Lode* angle of similarity

$$\theta = \frac{1}{3} \arcsin \left[ \frac{3\sqrt{3} III'_\tau}{2 II'_\tau \left(\frac{3}{2}\right)} \right] \quad (2.149)$$

This model is widely used for predicting damage in geomaterials.

### Drucker-Prager model

This criteria was formulated by *Drucker* and *Prager* in 1952, it is considered as a smoothed approximation of the Mohr-Coulomb model. It includes the first invariant of the stress tensor  $I_\tau$ , the second invariant of the deviatoric stress tensor  $II'_\tau$  and the angle of internal friction  $\phi$ . The influence of the third invariant of the deviatoric stress tensor  $III'_\tau$  is neglected. The mathematical expression for this model is given by:

$$\mathbb{F}(I_\tau, II'_\tau, c, \phi) = \underbrace{f(I_\tau, II'_\tau, \phi)}_{\zeta} - c(d) = \bar{\alpha}(\phi) I_\tau + \sqrt{II'_\tau} - \bar{\mathcal{K}}(d, \phi) = 0 \quad (2.150)$$

Functions  $\bar{\mathcal{K}}(d, \phi) = 6c(d) \cos \phi / (3\sqrt{3} + \sqrt{3} \sin \phi)$  and  $\bar{\alpha}(\phi) = 2 \sin \phi / (3\sqrt{3} + \sqrt{3} \sin \phi)$  describe the inscribed cone in the *Mohr-Coulomb* pyramid. Both models coincide along



the tensile meridians. When the cone is circumscribed to the *Mohr-Coulomb* pyramid, the coincidence affects the compression meridians of both surfaces. In that case the functions are  $\bar{\mathcal{K}}(d, \phi) = 6c(d) \cos \phi / (3\sqrt{3} - \sqrt{3} \sin \phi)$  and  $\bar{\alpha}(\phi) = 2 \sin \phi / (3\sqrt{3} - \sqrt{3} \sin \phi)$ . Both particular cases describe a very different behavior.

### Initial damage threshold and parameter A

For the damage threshold surfaces of *Von Mises*, *Mohr-Coulomb* and *Drucker-Prager* models the deduction of the  $A$  parameter for the softening laws (2.122) and (2.123) yields

$$A = \left( \frac{G_f E}{l_e (\zeta_0)^2} - \frac{1}{2} \right)^{-1} \geq 0 \quad (2.151)$$

where  $G_f$  is the specific fracture energy for unit of area,  $l_e$  is the characteristic length for the fractured domain [92],  $E$  is the elasticity modulus of the material and  $\zeta_0$  is the initial value for the damage criteria. The initial damage threshold that will be adopted for these models is

$$\zeta_0 = f(\bar{\boldsymbol{\tau}}_0) = f'_t \quad (2.152)$$

### Tangent constitutive tensor

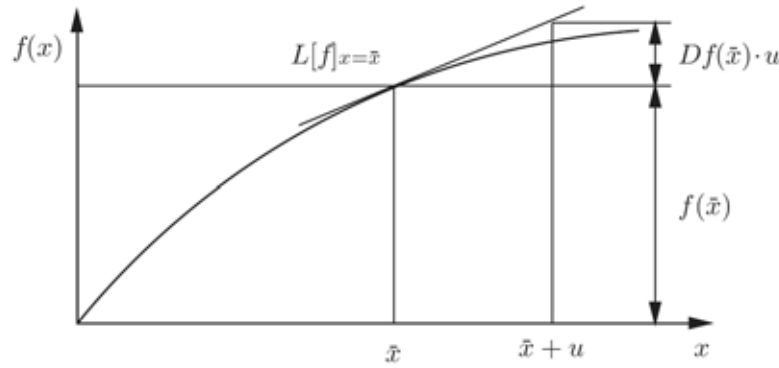
For the *Von Mises*, *Mohr-Coulomb* and *Drucker-Prager* damage surfaces the tangent constitutive tensors, after considering the softening laws (2.122) and (2.123) are given by the general form

$$\boldsymbol{\mathfrak{c}} = (1 - d) \check{\boldsymbol{\mathfrak{c}}} - \frac{\bar{\zeta} + A\zeta}{\zeta^2} e^A \left( 1 - \frac{\zeta}{\bar{\zeta}} \right) \left[ \frac{1}{\zeta} [\check{\boldsymbol{\mathfrak{c}}} : \mathbf{e}] \right] \otimes \left[ \check{\boldsymbol{\mathfrak{c}}} : \frac{\partial \bar{\boldsymbol{\tau}}}{\partial \mathbf{e}} \right] \quad \textit{Exponential softening} \quad (2.153)$$

$$\boldsymbol{\mathfrak{c}} = (1 - d) \check{\boldsymbol{\mathfrak{c}}} - \frac{\bar{\zeta}}{\zeta^2(1 + A)} \left[ \frac{1}{\zeta} [\check{\boldsymbol{\mathfrak{c}}} : \mathbf{e}] \right] \otimes \left[ \check{\boldsymbol{\mathfrak{c}}} : \frac{\partial \bar{\boldsymbol{\tau}}}{\partial \mathbf{e}} \right] \quad \textit{Linear softening} \quad (2.154)$$

Both expressions of the constitutive tensor are no-symmetric. Also, both expressions depend on the damage surfaces. When they are simple it is possible to obtain an analytical expression to compute the tangent constitutive tensor (for example in (2.136) and (2.137)). In other cases an analytical expression can not be obtained due to the difficulty of the expressions.

We should note that in a large displacement setting the above tangent constitutive tensors are not suitable. The expressions given only hold for infinitesimal deformations, giving a bad response in other cases. It is therefore necessary to develop a numerical procedure to obtain the tangent constitutive tensor for large displacements and for any damage surface. The most commonly used method is the *Perturbation* method, see [22] and [78]. This method is presented in Section 3.7.1 dealing with the numerical implementation.



**Figure 2.5:** Linearization of  $f$  at  $\bar{x}$  [131].

## 2.5 Linearizations

Linearization of the nonlinear mathematical models is necessary once the associated boundary value problems have to be solved. Nonlinearities in continuum mechanics come from the geometry, the constitutive equations and the unilateral boundary conditions, such as in contact. *Newton* method is the most known and is a very efficient solution algorithm for nonlinear continuum problems, especially for numerical methods, like the finite element method.

To explain the idea of linearization we consider for an scalar valued function  $f$  which is continuous and has continuous first derivatives ( $C^{-1}$  continuity). This assumption permits to express  $f$  by a *Taylor* series expansion at  $x$ :

$$f(\bar{x} + u) = f(\bar{x}) + Df(\bar{x}) \cdot u + R \quad (2.155)$$

The operator  $D$  denotes the derivative of  $f$  with respect to  $x$ . The "." is a simple multiplication in this case and  $u$  is an increment. The residual term  $R = R(u)$  has the property  $\lim_{u \rightarrow 0} \frac{R}{|u|} \rightarrow 0$ . Figure 2.5 depicts the geometrical interpretation of (2.155).

The tangent to the curve described by  $f$  at  $\bar{x}$  in the point  $(\bar{x}, f(\bar{x}))$  is

$$f(u) = f(\bar{x}) + Df(\bar{x}) \cdot u \quad (2.156)$$

The linear part of  $f(x)$  in  $x = \bar{x}$  defines the linearization

$$L[f]_{x=\bar{x}} \equiv f(u) \quad (2.157)$$

This is a result for the one-dimensional case but can be extended to a scalar valued functions in three dimensions, or to infinite dimensional functions. Let us consider the following  $C^1$ -mapping  $\mathbf{G} : \mathcal{E} \rightarrow \mathcal{F}$ . The *Taylor* series expansion is

$$\mathbf{G}(\bar{\mathbf{x}} + \mathbf{u}) = \bar{\mathbf{G}} + \bar{D}\mathbf{G} \cdot \mathbf{u} + \mathbf{R} \quad (2.158)$$

where  $\bar{\mathbf{x}}$  and  $\mathbf{u}$  are points in the associated space. The notation used is:

$$\bar{\mathbf{G}} = \mathbf{G}(\bar{\mathbf{x}}), \quad \bar{D}\mathbf{G} = D\mathbf{G}(\bar{\mathbf{x}}) = \left. \frac{\partial \mathbf{G}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\bar{\mathbf{x}}} \quad (2.159)$$

now " $\cdot$ " denotes the inner product of the elements characterizing the associated space. The directional derivative is:

$$\left. \frac{d}{d\epsilon} [\mathbf{G}(\bar{\mathbf{x}} + \epsilon \mathbf{u})] \right|_{\epsilon=0} = \bar{D}\mathbf{G} \cdot \mathbf{u} \quad (2.160)$$

For simplicity the directional derivative  $\bar{D}\mathbf{G} \cdot \mathbf{u}$  will be written as  $\Delta \bar{\mathbf{G}}$ . Thus the linear part of the mapping at  $\bar{\mathbf{x}}$  is

$$L[\mathbf{G}]_{\mathbf{x}=\bar{\mathbf{x}}} \equiv \bar{\mathbf{G}} + \Delta \bar{\mathbf{G}} \quad (2.161)$$

Usually the expressions (e.g. tensors) which refer to the current configuration are linearized in the reference configuration by performing first a *pull-back* transformation. There the linearization is computed according to the rules exposed above. Finally, the result is transformed back to the current configuration by a *push-forward* operation.

It is important to note that the *pull-back* and *push-forward* transformations depend on the description of the tensors. Thus some tensors (e.g. stress tensors) refer to a *covariant* base, where the directional derivative has the form:

$$D\boldsymbol{\tau} \cdot \mathbf{u} = \mathbf{F} \{ D[\mathbf{F}^{-1} \boldsymbol{\tau} \mathbf{F}^{-T}] \cdot \mathbf{u} \} \mathbf{F}^T \quad (2.162)$$

Tensors that refer to a *contravariant* base, like a strain tensor, have the directional derivative as

$$D\mathbf{e} \cdot \mathbf{u} = \mathbf{F}^{-T} \{ D[\mathbf{F}^T \mathbf{e} \mathbf{F}] \cdot \mathbf{u} \} \mathbf{F}^{-1} \quad (2.163)$$

### 2.5.1 Linearization of the kinematic equations

In this section we will linearize the strain measures for the reference and current configuration. Using (2.161) the linear part of the *Green-Lagrange* strain tensor of (2.17) is given as:

$$L[\mathbf{E}]_{\varphi=\bar{\varphi}} = \bar{\mathbf{E}} + \Delta \bar{\mathbf{E}} \quad (2.164)$$

where the directional derivative  $\Delta \bar{\mathbf{E}} = \bar{D}\mathbf{E} \cdot \mathbf{u}$  has to be computed according to (2.160)

$$\begin{aligned} \Delta \bar{\mathbf{E}} &= \bar{D}\mathbf{E} \cdot \mathbf{u} = \left. \frac{d}{d\epsilon} \left[ \frac{1}{2} \mathbf{F}^T (\bar{\boldsymbol{\varphi}} + \epsilon \mathbf{u}) \mathbf{F} (\bar{\boldsymbol{\varphi}} + \epsilon \mathbf{u}) - \mathbf{1} \right] \right|_{\epsilon=0} \\ &= \frac{1}{2} [\bar{\mathbf{F}}^T \text{Grad } \mathbf{u} + \text{Grad}^T \mathbf{u} \bar{\mathbf{F}}] \end{aligned} \quad (2.165)$$

Last result is linear in  $\mathbf{u}$  and depends on the deformation at  $\bar{\boldsymbol{\varphi}}$ , which is represented by  $\bar{\mathbf{F}}$ . At  $\boldsymbol{\varphi} = \mathbf{X}$  the computation of (2.165) yields the linear strain tensor

$$L[\mathbf{E}]_{\varphi=\mathbf{x}} = \mathbf{0} + \frac{1}{2}[\text{Grad } \mathbf{u} + \text{Grad}^T \mathbf{u}] \quad (2.166)$$

The next strain measure is the *Almansi* strain tensor  $\mathbf{e}$  (2.18), which is expressed in the current configuration. As mentioned in (2.163), an initial *pull-back* of  $\mathbf{e}$  is needed. After applying the directional derivative (2.165) a *push-forward* gives the result in the current configuration, i.e.

$$\begin{aligned} D\mathbf{e} \cdot \mathbf{u} &= \bar{\mathbf{F}}^{-T} \{[\bar{D}\mathbf{E} \cdot \mathbf{u}]\} \bar{\mathbf{F}}^{-1} = \frac{1}{2}(\text{Grad } \mathbf{u} \bar{\mathbf{F}}^{-1} + \bar{\mathbf{F}}^{-T} \text{Grad}^T \mathbf{u}) \\ &= \frac{1}{2}(\overline{\text{grad}} \mathbf{u} + \overline{\text{grad}}^T \mathbf{u}) = \bar{\nabla}^S \Delta \mathbf{u} \end{aligned} \quad (2.167)$$

Comparing the result with (2.165) gives

$$\Delta \bar{\mathbf{E}} = \bar{\mathbf{F}}^T \bar{\nabla}^S \Delta \mathbf{u} \bar{\mathbf{F}} \quad (2.168)$$

It can be seen that the linearization of the *Almansi* strain tensor with respect to the *Green-Lagrange* strain tensor has the same structure as in equation (2.33) for the time derivative of the *Green-Lagrange* strain tensor.

## 2.5.2 Linearization of the constitutive models

The linearization of the constitutive law depends on the algorithm used to integrate the evolution equations. The linearization can only be computed once the integration algorithm is known.

Let us consider an hyperplastic constitutive response function expressed in terms of the *Cauchy* stress tensor, that depends on the left *Cauchy-Green* tensor. Linearization yields to:

$$L[\boldsymbol{\sigma}]_{\varphi=\bar{\varphi}} = \bar{\boldsymbol{\sigma}} + \bar{D} \boldsymbol{\sigma} \cdot \mathbf{u} \quad (2.169)$$

According to (2.161) and starting from the reference configuration with the 2nd Piola-Kirchhoff tensor the expression (2.169) is

$$\begin{aligned} L[\mathbf{S}]_{\varphi=\bar{\varphi}} &= \bar{\mathbf{S}} + \bar{D}\mathbf{S} \cdot \mathbf{u} = \bar{\mathbf{S}} + \delta \bar{\mathbf{S}} \\ &= \bar{\mathbf{S}} + \left. \frac{\partial \mathbf{S}}{\partial \mathbf{C}} \right|_{\varphi=\bar{\varphi}} [\bar{D}\mathbf{C} \cdot \mathbf{u}] \end{aligned} \quad (2.170)$$

Last result can be reformulated using (2.97) and (2.165) as

$$L[\mathbf{S}]_{\varphi=\bar{\varphi}} = \bar{\mathbf{S}} + \bar{\mathbf{C}} [\Delta \bar{\mathbf{E}}] \quad (2.171)$$

Comparing with (2.170) yields

$$\Delta \bar{\mathbf{S}} = \bar{\mathbf{C}} [\Delta \bar{\mathbf{E}}] \quad (2.172)$$

where the relation (2.172) has the same structure as the incremental constitutive equations (2.96). The time derivatives are replaced by directional derivatives in this case.

To obtain the result of expression (2.169), equation (2.172) has to be *push-forward* to the current configuration. Using (2.168) and (2.94) yields

$$(\mathcal{L}_v \boldsymbol{\tau})_{ik} = F_{iA} F_{lC} F_{mD} F_{kB} \bar{\mathbf{C}}_{ABCD} \bar{\nabla}^S \Delta \mathbf{u}_{lm} \quad \mathcal{L}_v \boldsymbol{\tau} = \bar{\mathbf{c}} [\bar{\nabla}^S \Delta \mathbf{u}] \quad (2.173)$$

This has the same structure as the incremental constitutive equation (2.100). Using the *Cauchy* stress tensor  $\bar{\boldsymbol{\sigma}} = \frac{1}{J} \bar{\boldsymbol{\tau}}$  the incremental constitutive tensor for that relation is defined by

$$\bar{\bar{\mathbf{c}}} = \frac{1}{J} \bar{\mathbf{c}} \quad (2.174)$$

After all these considerations (2.169) can be re-written as

$$L[\boldsymbol{\sigma}]_{\varphi=\bar{\varphi}} = \bar{\boldsymbol{\sigma}} + \bar{\bar{\mathbf{c}}} [\bar{\nabla}^S \Delta \mathbf{u}] \quad (2.175)$$

### 2.5.3 Linearization of the weak form

As was seen in Section 2.3.2, the solution of the non-linear boundary problem was based on the variational formulation by the weak form of the principle of virtual work. For the solution of a nonlinear problem an iterative scheme has to be developed. That means that the discretization of the weak form results in a non-linear system of equations.

*Newton* method is the iterative algorithm chosen here for solving the non-linear system. Usually it is the most efficient scheme because it has quadratic convergence near the solution point. The method requires linearizing the non-linear equations using the directional derivative.

The linearization of the weak form with respect to the current configuration is based on equation (2.52). It is assumed that the linearization is computed at a deformation state  $\bar{\varphi}$  where the body is under equilibrium.

$$L[\mathbf{g}]_{\varphi=\bar{\varphi}} = \bar{\mathbf{g}}(\bar{\varphi}, \boldsymbol{\eta}) + D \bar{\mathbf{g}}(\bar{\varphi}, \boldsymbol{\eta}) \cdot \Delta \mathbf{u} \quad (2.176)$$

$\mathbf{g}$  is the same function as in (2.52) exchanging  $\varphi$  by  $\bar{\varphi}$ . The linearization is applied only to the first term because when the loading is assumed to be conservative and all other terms do not depend on the deformation. This gives

$$D g(\bar{\varphi}, \boldsymbol{\eta}) \cdot \Delta \mathbf{u} = \int_{\bar{\varphi}(\mathcal{B})} [D \boldsymbol{\sigma}(\bar{\varphi}) \cdot \Delta \mathbf{u}] \cdot \overline{\text{grad}} \boldsymbol{\eta} \, dv \quad (2.177)$$

The linearization of the *Cauchy* stress yields

$$D g(\bar{\varphi}, \boldsymbol{\eta}) \cdot \Delta \mathbf{u} = \int_{\bar{\varphi}(\mathcal{B})} \{ \overline{\text{grad}} \Delta \mathbf{u} \bar{\boldsymbol{\sigma}} + [D \bar{\boldsymbol{\sigma}} \cdot \Delta \mathbf{u}] \} \cdot \overline{\text{grad}} \boldsymbol{\eta} \, dv \quad (2.178)$$

where the quantities labeled with a bar have to be evaluated at  $\bar{\varphi}$ . Equation (2.175) is used for the linearization of the *Cauchy* stresses:

$$D \bar{\boldsymbol{\sigma}} \cdot \Delta \mathbf{u} = \bar{\mathbf{c}} [\bar{\nabla}^S \Delta \mathbf{u}] \quad (2.179)$$

where  $\bar{\mathbf{c}}$  is the incremental elasticity tensor in the current configuration evaluated at  $\bar{\varphi}$ . Using the trace operation and considering the symmetry of  $\bar{\mathbf{c}}$ , a compact form of (2.178) is obtained as

$$D g(\bar{\boldsymbol{\varphi}}, \boldsymbol{\eta}) \cdot \Delta \mathbf{u} = \int_{\bar{\boldsymbol{\varphi}}(\mathcal{B})} \{ \overline{\text{grad}} \Delta \mathbf{u} \bar{\boldsymbol{\sigma}} \cdot \overline{\text{grad}} \boldsymbol{\eta} + \bar{\nabla}^S \boldsymbol{\eta} \bar{\mathbf{c}} [\bar{\nabla}^S \Delta \mathbf{u}] \} dv \quad (2.180)$$

This equation is known as the *updated Lagrangian* formulation ([12], [13], [16], [131]). The deformation state  $\bar{\varphi}$  is always updated during the nonlinear incremental solution procedure. Equation (2.180) is the basis for the discretization using the finite element method.

## 2.6 Discretization of the continuum

The discrete equations for the updated lagrangian formulation are obtained with a finite element model which uses interpolants for the test and trial functions. The domain  $\boldsymbol{\varphi}(\mathcal{B})$  is subdivided into elements  $e$  from 1 to  $n_e$ . The domain of each element is denoted by  $\boldsymbol{\varphi}(\Omega_e)$ .

$$\boldsymbol{\varphi}(\mathcal{B}) = \bigcup_{e=1}^{n_e} \boldsymbol{\varphi}(\Omega_e) \quad (2.181)$$

Each element  $\boldsymbol{\varphi}(\Omega_e)$  have a certain number of nodes  $n_n$ . Each node has a spatial position in the current configuration  $x_I$ , where  $I = 1$  to  $n_n$ . Nodes define the connection between elements for the assembly of the domain. Most boundary conditions are imposed on nodes.

### 2.6.1 Isoparametric concept

The interpolation of any variable  $u$  is written in term of the nodal values of the shape functions  $N_I$  in the standard manner [137].

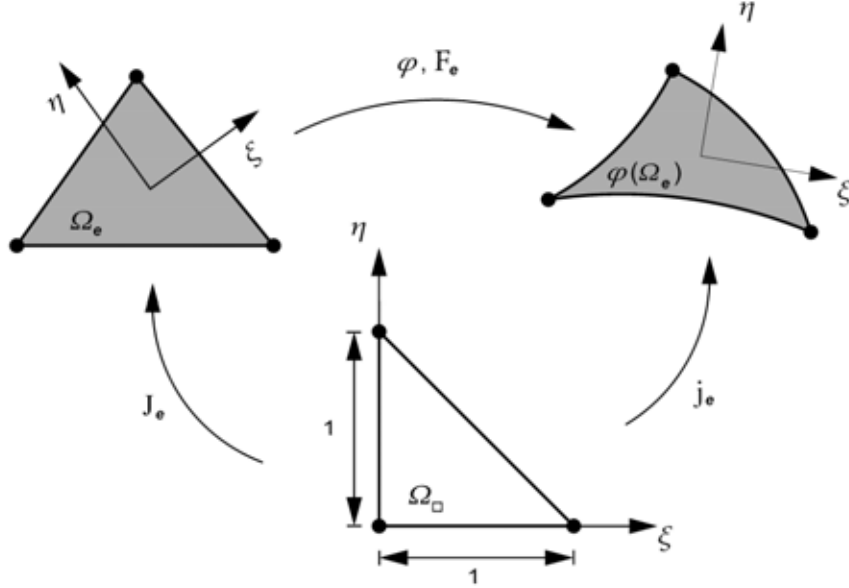
$$u(X, t) = \sum_{I=1}^{n_n} N_I(X) u_I(t) \quad (2.182)$$

where  $N_I(X)$  are  $C^0$  interpolants which satisfy the condition

$$N_I(X_J) = \delta_{IJ} \quad (2.183)$$

and  $\delta_{IJ}$  is the *Kronecker* delta.

The finite element model uses the isoparametric concept for the geometry. That means that shape functions are expressed in the parent configuration instead of the current one. The parent configuration is an elemental configuration defined in terms of the natural



**Figure 2.6:** Isoparametric description of a deformed element.

coordinates of the element. The natural coordinates define the geometry of the element by an orthogonal base that normalizes the shape functions.

The coordinates in the current configuration are defined as

$$\mathbf{x}_e = \sum_{I=1}^{n_n} N_I(\boldsymbol{\xi}) \mathbf{x}_I \quad (2.184)$$

Equation (2.184) is the basis for transforming the geometry from a configuration to another following standard transformation rules [131] (see Figure 2.6).

### Derivatives of the functions

The spatial derivatives are obtained by implicit differentiation because the functions are usually not explicitly invertible respect to the natural coordinates of the parent configuration. The chain rule is used in this case for computing the derivatives:

$$\nabla_{\boldsymbol{\xi}} N_I = \frac{\partial N_I}{\partial \boldsymbol{\xi}} = \frac{\partial N_I}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \quad (2.185)$$

where

$$\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} = \sum_{I=1}^{n_n} N_{I,\xi}(\boldsymbol{\xi}) \mathbf{x}_I \otimes \mathbf{E}_{\xi} \quad (2.186)$$

being  $\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}$  the Jacobian of the map between the current configuration of the element and the parent element configuration. It is denoted as:

$$\mathbf{j}_e = \text{Grad}_\xi \mathbf{x}_e = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} = \sum_{I=1}^{n_n} \mathbf{x}_I \otimes N_{I,\xi}(\boldsymbol{\xi}) \mathbf{E}_\xi \quad (2.187)$$

The derivatives of the functions between configurations can be written as

$$\nabla_\xi N_I = \mathbf{j}_e^T \nabla_x N_I \quad (2.188)$$

or the inverse relation

$$\nabla_x N_I = \mathbf{j}_e^{-T} \nabla_\xi N_I \quad (2.189)$$

The gradient of a vector field  $\mathbf{u}_e$  with respect to the current configuration  $\mathbf{x}$  will be expressed as

$$\text{grad } \mathbf{u}_e = \sum_{I=1}^{n_n} \mathbf{u}_I \otimes \mathbf{j}_e^{-T} \nabla_\xi N_I \quad (2.190)$$

Note that in all cases  $\mathbf{j}_e$  is the Jacobian between the parent and the current configuration. The transformation also can be made from the parent to the reference configuration. The Jacobian in this case is expressed as

$$\mathbf{J}_e = \text{Grad}_\xi \mathbf{X}_e = \frac{\partial \mathbf{X}}{\partial \boldsymbol{\xi}} = \sum_{I=1}^{n_n} \mathbf{X}_I \otimes N_{I,\xi}(\boldsymbol{\xi}) \mathbf{E}_\xi \quad (2.191)$$

The gradient of vector field  $\mathbf{u}_e$  respect to the reference configuration  $\mathbf{X}$  has the form:

$$\text{Grad } \mathbf{u}_e = \sum_{I=1}^{n_n} \mathbf{u}_I \otimes \mathbf{J}_e^{-T} \nabla_\xi N_I \quad (2.192)$$

where

$$\nabla_\xi N_I = \mathbf{J}_e^T \nabla_X N_I \quad \text{and} \quad \nabla_X N_I = \mathbf{J}_e^{-T} \nabla_\xi N_I \quad (2.193)$$

The deformation gradient of an element can be expressed as a function of the transformation jacobians between the parent and the spatial configurations as

$$\mathbf{F}_e = \mathbf{j}_e \mathbf{J}_e^{-1} \quad \text{and} \quad J_e = \det \mathbf{F}_e = \frac{\det \mathbf{j}_e}{\det \mathbf{J}_e} \quad (2.194)$$

The definition of (2.194) is shown in Figure 2.6.



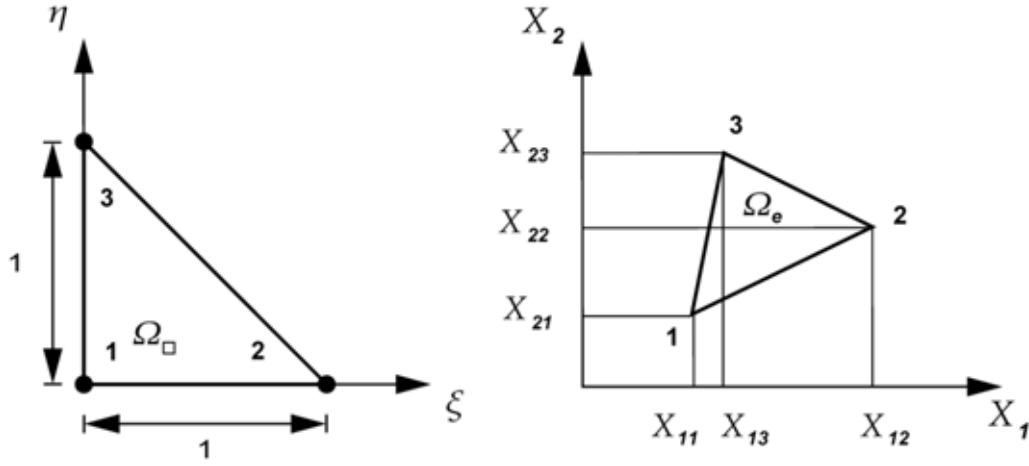


Figure 2.7: Linear Triangle, 3 nodes element.

### 2.6.2 Shape Functions

Vector fields in the discretization with finite elements are approximated with shape functions. The shape functions, or interpolation functions, are related to the dimension of the problem and the characteristics of the element chosen. The most simple and common element in two dimensions is the linear triangle. In three dimensions the simplest element is the linear tetrahedron.

#### Linear Triangle

The linear triangle has three nodes. Figure 2.7 shows the element and the  $\xi$  and  $\eta$  natural coordinates in the parent configuration  $\Omega_{\square}$ .

The shape functions for the linear triangle are

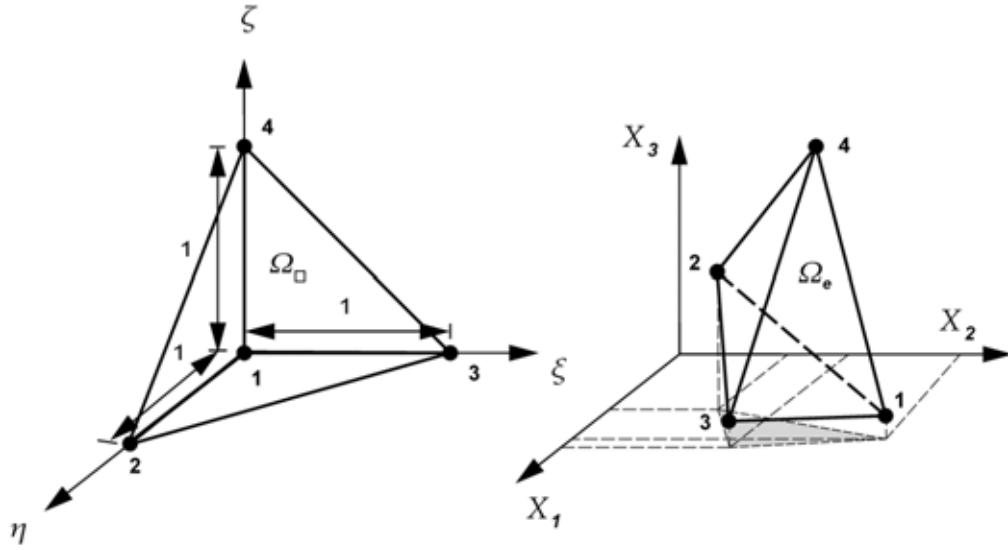
$$N_1 = 1 - \xi - \eta, \quad N_2 = \xi, \quad N_3 = \eta \quad (2.195)$$

The computation of the derivatives of the shape functions with respect to the current configuration for an element  $\Omega_e$  is expressed by

$$\nabla_x N_I = \begin{Bmatrix} N_{I,1} \\ N_{I,2} \end{Bmatrix} = \mathbf{j}_e^{-T} \begin{Bmatrix} N_{I,\xi} \\ N_{I,\eta} \end{Bmatrix} \quad (2.196)$$

where  $\mathbf{j}_e$  is the jacobian matrix for the transformation between the current and the parent configurations given by equation (2.187). This transformation is needed for computing the gradient of a vectorial field. Using (2.187) we obtain

$$\begin{Bmatrix} N_{I,1} \\ N_{I,2} \end{Bmatrix} = \frac{1}{\det \mathbf{j}_e} \begin{bmatrix} \mathbf{x}_{2,\eta} & -\mathbf{x}_{2,\xi} \\ -\mathbf{x}_{1,\eta} & \mathbf{x}_{1,\xi} \end{bmatrix} \begin{Bmatrix} N_{I,\xi} \\ N_{I,\eta} \end{Bmatrix} \quad (2.197)$$



**Figure 2.8:** Four-noded tetrahedron.

where the derivatives of the linear shape functions (2.195) are given by

$$\begin{aligned} N_{1,\xi} &= -1 & N_{2,\xi} &= 1 & N_{3,\xi} &= 0 \\ N_{1,\eta} &= -1 & N_{2,\eta} &= 0 & N_{3,\eta} &= 1 \end{aligned} \quad (2.198)$$

and the coordinates derivatives are computed by

$$\mathbf{x}_{\alpha,\beta} = \sum_{I=1}^{n_n} \mathbf{N}_{I,\beta} \mathbf{x}_{\alpha,I} \quad (2.199)$$

### Linear tetrahedra

The linear tetrahedra has four nodes. The natural coordinates for the linear tetrahedra are shown in Figure 2.8. The shape functions are

$$N_1 = 1 - \xi - \eta - \zeta, \quad N_2 = \xi, \quad N_3 = \eta, \quad N_4 = \zeta \quad (2.200)$$

The derivatives of the shape functions with respect to the current configuration are completed by

$$\nabla_x N_I = \begin{Bmatrix} N_{I,1} \\ N_{I,2} \\ N_{I,3} \end{Bmatrix} = \mathbf{j}_e^{-T} \begin{Bmatrix} N_{I,\xi} \\ N_{I,\eta} \\ N_{I,\zeta} \end{Bmatrix} \quad (2.201)$$

In three dimensions the jacobian matrix (2.187) has the expression:

$$\mathbf{j}_e = \sum_{I=1}^{n_n} \mathbf{x}_I \otimes \nabla_{xi} N_I = \begin{bmatrix} \mathbf{x}_{1,\xi} & \mathbf{x}_{1,\eta} & \mathbf{x}_{1,\zeta} \\ \mathbf{x}_{2,\xi} & \mathbf{x}_{2,\eta} & \mathbf{x}_{2,\zeta} \\ \mathbf{x}_{3,\xi} & \mathbf{x}_{3,\eta} & \mathbf{x}_{3,\zeta} \end{bmatrix} \quad (2.202)$$

The components of (2.202) are computed by

$$\mathbf{x}_{r,k} = \sum_{I=1}^{n_n} \mathbf{N}_{I,k} \mathbf{x}_{r,I} \quad (2.203)$$

### 2.6.3 Integration in the parameterized space

For the computation of the weak form (2.180) an integration of the functions or their derivatives over the element domain  $\Omega_e$  is needed. They will be properly exported to the parameterized parent configuration of the element  $\Omega_\square$ . That also requires a transformation of the integrals

$$\int_{\varphi(\Omega_e)} g(\mathbf{x}) d\Omega_e = \int_{\varphi(\Omega_\square)} g(\boldsymbol{\xi}) \det \mathbf{j}_e d\Omega_\square \quad (2.204)$$

In two dimensions the integral is performed over a plane domain. The triangle equation (2.204) is expressed in the natural coordinate system as

$$\int_{\varphi(\Omega_e)} g(\mathbf{x}) da = \int_{\varphi(\Omega_\square)} g(\boldsymbol{\xi}) \det \mathbf{j}_e d\Omega_\square = \int_0^1 \int_0^{1-\xi} g(\xi, \eta) \det \mathbf{j}_e d\eta d\xi \quad (2.205)$$

The integration is computed numerically. The product  $(g(\xi, \eta) \det \mathbf{j}_e)$  is represented by a rational function. That allows to write the integral (2.205) with the following approximation:

$$\int_0^1 \int_0^{1-\xi} g(\xi, \eta) \det \mathbf{j}_e d\eta d\xi \approx \sum_{p=1}^{n_p} g(\xi_p, \eta_p) \det \mathbf{j}_e(\xi_p, \eta_p) W_p \quad (2.206)$$

where  $n_p$  is the number of integration points that will define the quadrature order.  $W_p$  are weights of the integration points. Typically a *Gauss* quadrature is used ([102], [51]). For the three dimensional space, the integration is done over a volume. Equation (2.204) becomes

$$\int_{\varphi(\Omega_e)} g(\mathbf{x}) dv = \int_{\varphi(\Omega_\square)} g(\boldsymbol{\xi}) \det \mathbf{j}_e d\Omega_\square = \int_0^1 \int_0^{1-\xi} \int_0^{1-\eta-\zeta} g(\xi, \eta, \zeta) \det \mathbf{j}_e d\xi d\eta d\zeta \quad (2.207)$$

The approximation of (2.207) for the numerical integration is

$$\int_0^1 \int_0^{1-\xi} \int_0^{1-\eta-\zeta} g(\xi, \eta, \zeta) \det \mathbf{j}_e d\xi d\eta d\zeta \approx \sum_{p=1}^{n_p} g(\xi_p, \eta_p, \zeta_p) \det \mathbf{j}_e(\xi_p, \eta_p, \zeta_p) W_p \quad (2.208)$$

The *Gauss* integration quadrature to solve (2.208) can be found in [81], [137].

#### 2.6.4 Discretization of the weak form

The continuum is discretized with finite elements and the functions that describe the geometry and the movement of the domain are approximated numerically. To define the general equations in the finite element domain the integrals of the weak form are computed as

$$\int_{\varphi(\mathcal{B})} (\dots) dv = \bigcup_{e=1}^{n_e} \int_{\varphi(\Omega_e)} (\dots) dV = \bigcup_{e=1}^{n_e} \int_{\Omega_{\square}} (\dots) d\Omega \quad (2.209)$$

The operator  $\bigcup$  represents the assembly of all elemental quantities to a global algebraic system of equations. The methodology for the assembly of the finite elements can be found in the literature (e.g. [81]).

The weak form of the balance of momentum (2.54) is composed by three integrals. They can be analyzed separately and discretized with the finite element method.

#### Internal forces

The first term of (2.54)

$$\int_{\varphi(\mathcal{B})} \boldsymbol{\sigma} \cdot \nabla^S \boldsymbol{\eta} dv \quad (2.210)$$

is the virtual internal work. This expresses the integration of the internal nodal forces over the continuum. The variation of the virtual valued function  $\nabla^S \boldsymbol{\eta}$  in the elemental approximation can be written as:

$$\nabla^S \boldsymbol{\eta}_e = \frac{1}{2} \sum_{I=1}^{n_n} [(\boldsymbol{\eta}_I \otimes \nabla_x N_I) + (\nabla_x N_I \otimes \boldsymbol{\eta}_I)] \quad (2.211)$$

using the index notation, equation (2.211) becomes

$$(\nabla^S \boldsymbol{\eta}_e)_{ik} = \frac{1}{2} \sum_{I=1}^{n_n} [(\boldsymbol{\eta}_{iI} N_{I,k}) + (N_{I,i} \boldsymbol{\eta}_{kI})] \quad (2.212)$$

where  $N_{I,m} = \frac{\partial N_I}{\partial x_m}$ .

Using the isoparametric concept, the shape functions are approximated as in equation (2.193). In the three dimensional case this yields

$$N_{I,k} = \{j_e^{-1}\}_{1k} N_{I,\xi} + \{j_e^{-1}\}_{2k} N_{I,\eta} + \{j_e^{-1}\}_{3k} N_{I,\zeta} \quad (2.213)$$

where  $\{j_e^{-1}\}_{ik}$  is the  $ik$  component of the inverse of the *Jacobian* matrix  $\mathbf{j}_e$ .

Due to the symmetry of  $\nabla^S \boldsymbol{\eta}_e$ , equation (2.212) can be written as

$$\nabla^S \boldsymbol{\eta} = \sum_{I=1}^{n_n} \begin{bmatrix} N_{I,1} & 0 & 0 \\ 0 & N_{I,2} & 0 \\ 0 & 0 & N_{I,3} \\ N_{I,2} & N_{I,1} & 0 \\ 0 & N_{I,3} & N_{I,2} \\ N_{I,3} & 0 & N_{I,1} \end{bmatrix} \begin{Bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{Bmatrix}_I = \sum_{I=1}^{n_n} \mathbf{B}_{0I} \boldsymbol{\eta}_I \quad (2.214)$$

subscript "0" in matrix  $\mathbf{B}_{0I}$  denotes that the matrix is independent of the displacement field, i.e. it does not contain any displacements. The difference between  $\mathbf{B}_{0I}$  and the standard  $\mathbf{B}$  matrix of the linear theory (see [137]) is that here all derivatives in  $\mathbf{B}_{0I}$  have to be computed with respect to the coordinates  $\mathbf{x}$  of the current configuration.

Expressing the *Cauchy* stress tensor in a vector form  $\boldsymbol{\sigma} = \{\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{23}, \sigma_{13}\}^T$  the virtual internal forces (2.210) become

$$\begin{aligned} \boldsymbol{\eta}^T \mathbf{F}^{int} &= \int_{\boldsymbol{\varphi}(\mathcal{B})} \boldsymbol{\sigma} \cdot \nabla^S \boldsymbol{\eta} \, dv = \bigcup_{e=1}^{n_e} \int_{\boldsymbol{\varphi}(\Omega_e)} (\nabla^S \boldsymbol{\eta})^T \boldsymbol{\sigma}_e \, dV \\ &= \bigcup_{e=1}^{n_e} \sum_{I=1}^{n_n} \boldsymbol{\eta}_I^T \int_{\boldsymbol{\varphi}(\Omega_e)} \mathbf{B}_{0I}^T \boldsymbol{\sigma}_e \, dV \\ &= \bigcup_{e=1}^{n_e} \sum_{I=1}^{n_n} \boldsymbol{\eta}_I^T \int_{\Omega_{\square}} \mathbf{B}_{0I}^T \boldsymbol{\sigma}_e \det \mathbf{j}_e \, d\Omega_{\square} \end{aligned} \quad (2.215)$$

where  $dV = J_e d\Omega$  (2.194), as the integration is performed in the element current configuration  $\boldsymbol{\varphi}(\Omega_e)$ .

### External Forces

Analyzing the other terms of (2.54) the external contributions to the system are given by

$$\int_{\boldsymbol{\varphi}(\mathcal{B})} \rho \bar{\mathbf{b}} \cdot \boldsymbol{\eta} \, dv - \int_{\boldsymbol{\varphi}(\partial \mathcal{B}_\sigma)} \bar{\mathbf{t}} \cdot \boldsymbol{\eta} \, d\Gamma \quad (2.216)$$

Using (2.184), the valued function  $\boldsymbol{\eta}$  is approximated as

$$\boldsymbol{\eta}_e = \sum_{I=1}^{n_n} N_I \boldsymbol{\eta}_I \quad (2.217)$$

The above approximation allows (2.216) to write

$$\begin{aligned} \boldsymbol{\eta}^T \mathbf{F}^{ext} &= \int_{\boldsymbol{\varphi}(\mathcal{B})} \rho \bar{\mathbf{b}} \cdot \boldsymbol{\eta} \, dv - \int_{\boldsymbol{\varphi}(\partial \mathcal{B}_\sigma)} \bar{\mathbf{t}} \cdot \boldsymbol{\eta} \, d\Gamma = \bigcup_{e=1}^{n_e} \sum_{I=1}^{n_n} \boldsymbol{\eta}_I^T \int_{\boldsymbol{\varphi}(\Omega_e)} \rho \bar{\mathbf{b}} N_I \, dV \\ &\quad + \bigcup_{s=1}^{n_s} \sum_{I=1}^{n_m} \boldsymbol{\eta}_I^T \int_{\boldsymbol{\varphi}(\Gamma_s)} N_I \bar{\mathbf{t}} \, dS \end{aligned} \quad (2.218)$$

where  $n_r$  is the number of segments that discretize the surface of the domain and  $n_m$  the number of nodes for each surface element. The integration domains are  $dV = J_e d\Omega$  and  $dS = J_s d\Gamma$  being "e" and "s" subindexes indicate *volume element* and *surface element*, respectively.

### Dynamic Forces

The last term in expression (2.54) expresses the inertial virtual work of the forces within the system as

$$\int_{\varphi(\mathcal{B})} \rho \dot{\mathbf{v}} \cdot \boldsymbol{\eta} dv \quad (2.219)$$

Equation (2.219) can be transformed using (2.220) and the approximation for the acceleration field as

$$\dot{\mathbf{v}}(\mathbf{x}, t) = \sum_{K=1}^{n_n} N_K(\boldsymbol{\xi}) \dot{\mathbf{v}}_K \quad (2.220)$$

The resulting expression is:

$$\begin{aligned} \int_{\varphi(\mathcal{B})} \rho \dot{\mathbf{v}} \cdot \boldsymbol{\eta} dv &= \bigcup_{e=1}^{n_e} \int_{\varphi(\Omega_e)} \rho \boldsymbol{\eta}_I^T \dot{\mathbf{v}} dv \\ &= \bigcup_{e=1}^{n_e} \sum_{I=1}^{n_n} \sum_{K=1}^{n_n} \boldsymbol{\eta}_I^T \int_{\varphi(\Omega_e)} N_I \rho N_K dV \dot{\mathbf{v}}_K \end{aligned} \quad (2.221)$$

With the application of the unit matrix  $\dot{\mathbf{v}}_K = \mathbf{I} \dot{\mathbf{v}}_K$  the elemental mass matrix is obtained:

$$\mathbf{M}_{IK} = \int_{\varphi(\Omega_e)} N_I \rho N_K dV \mathbf{I} \quad (2.222)$$

Using the quadrature of the element and integrating numerically (2.222) yields

$$\begin{aligned} \mathbf{M}_{IK} = \int_{\varphi(\Omega_e)} N_I \rho N_K dV \mathbf{I} &= \int_{\varphi(\Omega_\square)} N_I \rho N_K \det \mathbf{J}_e d\square \mathbf{I} \\ &= \sum_{p=1}^{n_p} N_I(\boldsymbol{\xi}_p) \rho(\boldsymbol{\xi}_p) N_K(\boldsymbol{\xi}_p) \det \mathbf{J}_e(\boldsymbol{\xi}_p) W_p \mathbf{I} \end{aligned} \quad (2.223)$$

where  $\mathbf{M}_{IK}$  is the elemental mass matrix. The assembled expression is

$$\boldsymbol{\eta}^T \mathbf{M} \dot{\mathbf{v}} = \int_{\varphi(\mathcal{B})} \rho \dot{\mathbf{v}} \cdot \boldsymbol{\eta} dv = \bigcup_{e=1}^{n_e} \sum_{I=1}^{n_n} \sum_{K=1}^{n_n} \boldsymbol{\eta}_I^T \mathbf{M}_{IK} \dot{\mathbf{v}}_K \quad (2.224)$$

being  $\mathbf{M}$  the consistent mass matrix for the whole mesh.

The discretization of the weak form of the balance of momentum (2.54) gives a general expression for the nonlinear system of equations as

$$\boldsymbol{\eta}^T [\mathbf{M} \dot{\mathbf{v}} + \mathbf{F}^{int}(\mathbf{u}) - \mathbf{F}^{ext}] = 0 \quad (2.225)$$

For any virtual function  $\boldsymbol{\eta}$  the nonlinear discrete form for the differential system of equations becomes

$$\boxed{\mathbf{M} \dot{\mathbf{v}} + \mathbf{F}^{int}(\mathbf{u}) - \mathbf{F}^{ext} = 0} \quad , \quad \forall \mathbf{u} \in \mathbb{R}^N \quad (2.226)$$

### Tangent Stiffness Matrix

The solution of the non linear equation (2.226) requires the linearization of the discretized form of the weak equations. The discretized tangent matrix is obtained from the global equation (2.180) as

$$\int_{\bar{\varphi}(\mathcal{B})} \overline{\text{grad}} \Delta \mathbf{u} \bar{\boldsymbol{\sigma}} \cdot \overline{\text{grad}} \boldsymbol{\eta} + \bar{\nabla}^S \boldsymbol{\eta} \bar{\boldsymbol{\epsilon}} [\bar{\nabla}^S \Delta \mathbf{u}] dv \quad (2.227)$$

For each element the gradient function is discretized as follows:

$$\begin{aligned} \overline{\text{grad}} \Delta \mathbf{u}_e &= \sum_{K=1}^{n_n} \Delta \mathbf{u}_K \otimes \nabla_{\bar{x}} N_K \\ \overline{\text{grad}} \boldsymbol{\eta}_e &= \sum_{I=1}^{n_n} \boldsymbol{\eta}_I \otimes \nabla_{\bar{x}} N_I \end{aligned} \quad (2.228)$$

The first part of the integral (2.227) yields

$$\boldsymbol{\eta}^T \mathbf{K}^{geo} \Delta \mathbf{u} = \int_{\bar{\varphi}(\mathcal{B})} \overline{\text{grad}} \Delta \mathbf{u} \bar{\boldsymbol{\sigma}} \cdot \overline{\text{grad}} \boldsymbol{\eta} dv = \bigcup_{e=1}^{n_e} \sum_{I=1}^{n_n} \sum_{K=1}^{n_n} \boldsymbol{\eta}_I^T \int_{\boldsymbol{\varphi}(\Omega_e)} (\nabla_{\bar{x}} N_I^T) \bar{\boldsymbol{\sigma}}_e \nabla_{\bar{x}} N_K \mathbb{I} dV \Delta \mathbf{u}_K \quad (2.229)$$

where  $\mathbf{K}^{geo}$  is the *geometric stiffness matrix* (see [12] and [13]). For the second term

$$\int_{\bar{\varphi}(\mathcal{B})} \bar{\nabla}^S \boldsymbol{\eta} \bar{\boldsymbol{\epsilon}} [\bar{\nabla}^S \Delta \mathbf{u}] dv \quad (2.230)$$

The update of the material tensor in the current configuration  $\bar{\varphi}(\mathcal{B})$  introduces matrix  $\bar{\boldsymbol{\epsilon}}$  directly from the incremental constitutive equation, i.e. (2.102). Using the discretization of (2.214) for the virtual parent function  $\boldsymbol{\eta}$  and for the displacement increment  $\Delta \mathbf{u}$ , equation (2.230) gives:

$$\boldsymbol{\eta}^T \mathbf{K}^{mat} \Delta \mathbf{u} = \int_{\bar{\boldsymbol{\varphi}}(\mathcal{B})} \bar{\nabla}^S \boldsymbol{\eta} \bar{\boldsymbol{\epsilon}} [\bar{\nabla}^S \Delta \mathbf{u}] dv = \bigcup_{e=1}^{n_e} \sum_{I=1}^{n_n} \sum_{K=1}^{n_n} \boldsymbol{\eta}_I^T \int_{\boldsymbol{\varphi}(\Omega_e)} \bar{\mathbf{B}}_{0I}^T \bar{\mathbf{D}}^M \bar{\mathbf{B}}_{0K} dV \Delta \mathbf{u}_K \quad (2.231)$$

where  $\bar{\mathbf{D}}^M$  is the incremental material constitutive tensor  $\bar{\boldsymbol{\epsilon}}$  in the current configuration. An example for a *Neo-Hookean* material can be found in (2.102).  $\mathbf{K}^{mat}$  is the *material stiffness matrix*. In the infinitesimal theory  $\mathbf{K}^{mat}$  is the classical stiffness matrix  $\mathbf{K} = \mathbf{K}^{mat}$ . For large displacements, the tangent matrix for a non-linear problem is written in the current configuration as the sum of both stiffness matrices:

$$\begin{aligned} \boldsymbol{\eta}^T \mathbf{K}^M \Delta \mathbf{u} &= \boldsymbol{\eta}^T (\mathbf{K}^{mat} + \mathbf{K}^{geo}) \Delta \mathbf{u} \\ &= \bigcup_{e=1}^{n_e} \sum_{I=1}^{n_n} \sum_{K=1}^{n_n} \boldsymbol{\eta}_I^T \int_{\boldsymbol{\varphi}(\Omega_e)} [(\nabla_{\bar{x}} N_I^T) \bar{\boldsymbol{\sigma}}_e \nabla_{\bar{x}} N_K \mathbb{I} + \bar{\mathbf{B}}_{0I}^T \bar{\mathbf{D}}^M \bar{\mathbf{B}}_{0K}] dV \Delta \mathbf{u}_K \end{aligned} \quad (2.232)$$

$\mathbf{K}^M$  is the result of the linearization of the weak form (2.180) and is called the *tangent matrix* of the system of equations. In Section 3.6 these linearizations will be used in an implicit scheme for the resolution of the dynamic equation (2.226).





*The important thing in science  
is not so much to obtain new facts  
as to discover new ways  
of thinking about them.*

Sir William Bragg (1862-1942)

# 3

## Formulation of the Particle Finite Element Method

### 3.1 Introduction

In the Particle Finite Element Method (PFEM) the continuum is described as a cloud of particles of infinitesimal size. A particle is a material point that has some information associated to it. The particles contain all the properties of the continuum (displacements, velocities, strains, stresses, material properties, internal variables, etc.) That means that they have enough information to reedit the original geometry and to recover the properties of a given continuum domain.

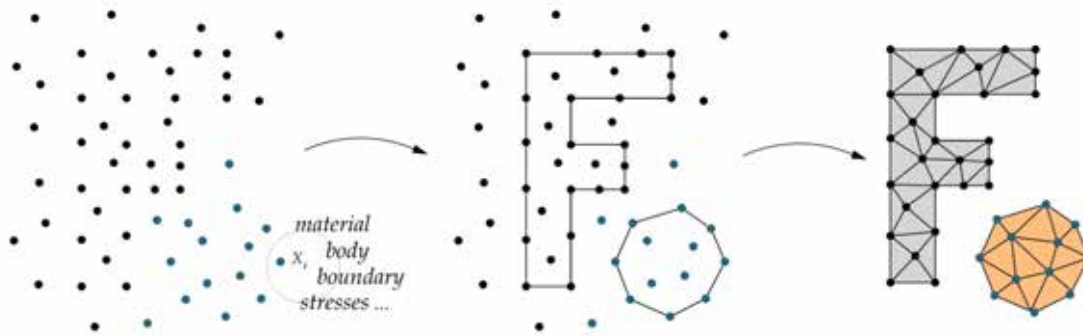
This cloud of particles is the basis of the PFEM. From them a finite element mesh is generated at any time step using a *Delaunay* triangulation [21]. The properties of the particles are transferred to the elements of the mesh for the computation.

In this chapter the particular properties of the PFEM applied to continua and structures will be discussed. There are many advantages in the use of this technique for the analysis of dynamic problems. To have the basis of the calculation in single particles allows us to construct the most suitable mesh for the required computation. At the same time it permits to capture rapidly changing boundaries of a continuum under large deformations. However, these are not significant advantages for an isolated non-linear analysis of a single structural domain. When a simple boundary problem is under study, some aspects of the method are more problematic than advantageous. The main problems emanate from the information transfer operations. In those cases, the use of the complete particle method is not needed for the analysis. A simplification is done automatically using only the features of the classical finite element analysis and activating the PFEM analysis only when is necessary. This will be explained in detail in the following sections.

### 3.2 Particle concept

The presented method is based on particles for one important reason: the type of problems that are going to be analyzed will have a changing morphology. The boundaries are not conserved and are eroded. There are several domains interacting with each other and experiencing geometrical changes. The PFEM is conceived to capture large displacements and deformations in such a changing multiple body system.

In order to not apply a computationally expensive discrete method, the particle definition is matched with a finite element mesh. With this mesh, the domain is fully discretized, although all the information is stored in the particles. A cloud of particles defines the dif-



**Figure 3.1:** Discrete body definition from a cloud of particles

ferent solid domains. See Figure 3.1.

Every time a mesh is regenerated for the computation, the element information is taken from the values at the particles. That means that particles must store the information and be a container for the variables. These features distinguish a particle from a node, which is only a geometric position. *A particle contains all the information about the key variables in the domain and also has the functionality of a node.*

In most cases particles coincide geometrically with the nodes of the mesh but this is not mandatory. Some problems require to introduce a larger number of particles, or alternatively to refine the mesh in an adaptative manner. Hence some particles have to be added or removed during the analysis process. Therefore the number of particles and nodes does not need to remain the same along time. The properties of the particles are interpolated to nodes when they do not coincide. Then, the information from the nodes is transferred to the integration points of a finite element mesh. This transfer is a critical process that can bring unexpected consequences. The scheme used for the transfer of the information and the containers of the particle values will be presented in Section 3.5.

### 3.3 Continuum formulation for the PFEM

In the PFEM the motion is described using the Updated Lagrangian formulation described in the previous chapter. There, it was mentioned that the fundamental equations of continuum mechanics arise from conservation laws. These equations must always be satisfied by the physical systems and they describe balance between the variables of the system. They are expressed by partial differential equations (PDEs). Applying the laws to a domain of a body an integral relation is obtained. PDEs are extracted from the integral relation by means of a proper derivation. In the Lagrangian description of the continuum, the conservation equations in the strong form are considered next. The equations are directly developed in terms of updated Lagrangian measures of stress and strain in its current configuration  $\varphi(\mathcal{B})$ . The major dependent variables are the initial density  $\rho_0(\mathbf{X}, t)$ , the displacement  $\mathbf{u}(\mathbf{X}, t)$ , and the Lagrangian measures of stress and strain.

*Equation of mass conservation* (continuity equation). From equation (2.41)

$$\rho(\mathbf{X}, t)J(\mathbf{X}, t) = \rho_0(\mathbf{X}, t) \quad (3.1)$$

*Equation of conservation of linear momentum*. From equation (2.43)

$$\frac{\partial \sigma_{ij}}{\partial x_j} + \rho b_i = \rho \ddot{u}_i \quad (3.2)$$

*Equation of conservation of angular momentum*. From equation (2.45)

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^T \quad (3.3)$$

*Constitutive equation*. In our work we will typically use a damage model (2.103)

$$\sigma_{ij} = (1 - d)C_{ijkl}E_{kl} \quad (3.4)$$

where  $E_{kl}$  is the linear part of the Green-Lagrange strain tensor and  $d$  is the damage internal variable. We note that any constitutive model can be used in the context of the PFEM.

The constitutive equations are usually formulated in terms of the PK2 stress  $\mathbf{S}$  (2.38) and the Green-Lagrange strain  $\mathbf{E}$  (2.17). This refers to the *Total Lagrangian* formulation (TL). However, the *Updated Lagrangian* measures are expressed in the current configuration. The Cauchy stress  $\boldsymbol{\sigma}$  (2.35) is used as well as the *Almansi* strain  $\mathbf{e}$  (2.18), or the symmetric gradient of displacements  $\nabla^S \mathbf{u}$  after the linearization of the equations.

The integral form of the equilibrium equation is represented by the virtual work principle. Substituting the finite element interpolation into the principle of virtual work leads to the standard residual equation

$$\mathbf{r}(\mathbf{u}, t) := \mathbf{M} \ddot{\mathbf{u}} - \mathbf{F}^{ext}(\mathbf{u}, t) + \mathbf{F}^{int}(\mathbf{u}, t) = 0 \quad (3.5)$$

This is the weak formulation of the weighted residual form for the displacement-based *Updated Lagrangian* formulation (UL). It was developed in Section 2.3 and expressed in (2.226)

The solution of above set of equations in time using the UL formulation requires the definition of boundary conditions and initial conditions for  $\mathbf{u}_0$ ,  $\dot{\mathbf{u}}_0$ ,  $\ddot{\mathbf{u}}_0$ , and also the external forces applied to the physical domain.

The particularities of the PFEM have not a direct influence in the solid mechanics formulation. They only affect the procedures for the resolution of (3.5).

In a dynamic finite element analysis the spatial domain  $\varphi(\mathcal{B}^t)$  is discretized at time  $t$ . The space is then described by finite elements and the time is expressed as the sum of small time steps. The integration, or sum, of small portions of space domains (finite elements) and time steps yields the solution of the continuum along time. The specific features of the PFEM affect the procedure for computing finite element variables during the transition between time steps.

The first process during the time step transition is the update of the variable values from the current configuration  $\varphi_t(\mathcal{B}^t)$  to the next reference configuration  $\mathcal{B}^{t+\Delta t}$ . This transition is defined by the incremental equations of motion, which are presented in the next section.

### 3.4 Incremental equations of motion based on particles

To describe a dynamic process, equation (3.5) has to be solved in time. When time changes, the position of the domain is not the same anymore. This is described by the UL formulation as a process that starts in the reference configuration  $\mathcal{B}^t$  and ends up at the current configuration  $\varphi_t(\mathcal{B}^t)$ . The spatial function  $\varphi(\mathbf{X}, t)$  defined in (2.1) maps the kinematic evolution of the particles of the body into a region of the Euclidean space.

In finite elements, the dynamic analysis is subdivided in time steps  $\Delta t$ . After the mapping of the body in one time step, the current configuration  $\varphi_t(\mathcal{B}^t)$  at time  $t$ , is updated and replaces the reference configuration  $\mathcal{B}^{t+\Delta t}$  for the next time step  $t + \Delta t$ . This update in the UL formulation can be expressed as follows

$$\mathcal{B}^{t+\Delta t} \simeq \varphi_t(\mathcal{B}^t) \quad (3.6)$$

Physical and mechanical variables change as the reference configuration is updated. The variables computed at each time step are updated and transformed to the new (current) reference configuration. This evolution of the variables is written in incremental form as

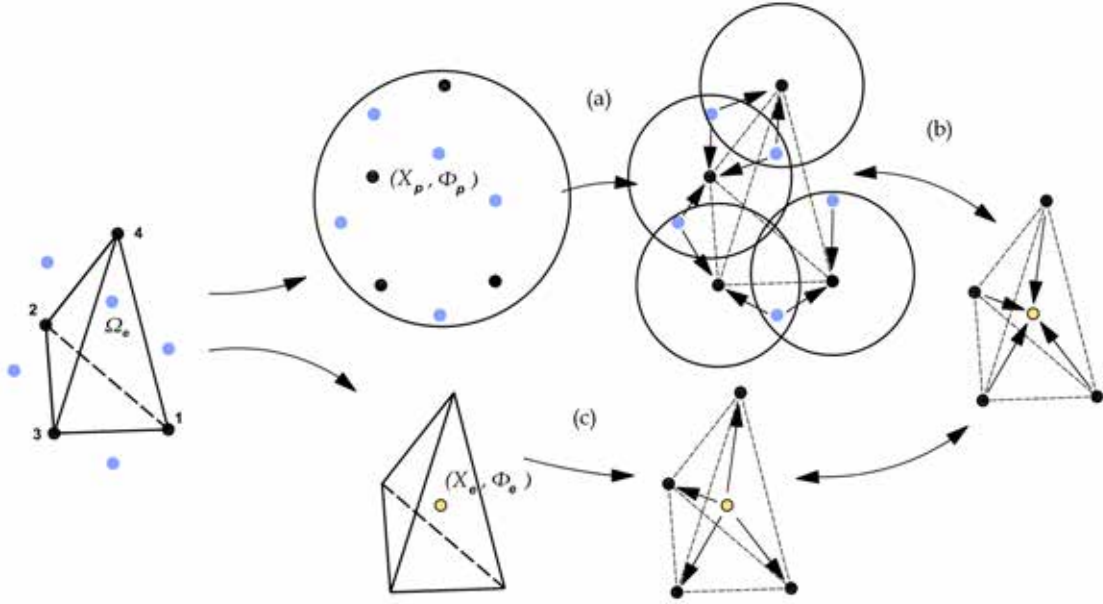
$$\mathbf{u}^{t+\Delta t} = \mathbf{u}^t + \Delta \mathbf{u} \quad (3.7)$$

$$\mathbf{S}^{t+\Delta t} = \boldsymbol{\sigma}^t + \Delta \mathbf{S} \quad (3.8)$$

where  $\mathbf{S}$  is the second *Piola-Kirchhoff* stress tensor (2.38) and  $\boldsymbol{\sigma}$  the *Cauchy* stress tensor (2.35). Note that  $\mathbf{S}_{\mathcal{B}^{t+\Delta t}}^t = \boldsymbol{\sigma}_{\varphi_t(\mathcal{B}^t)}^t$  expressed in the current reference configuration  $\mathcal{B}^{t+\Delta t}$ . The stress measure expressed in the current configuration  $\varphi(\mathcal{B}^{t+\Delta t})$  is a *push forward* of the second *Piola-Kirchhoff* (see (2.39); i.e.

$$\boldsymbol{\sigma}^{t+\Delta t} = \frac{1}{J} \mathbf{F} \mathbf{S}^{t+\Delta t} \mathbf{F}^T \quad (3.9)$$

The main feature of the PFEM is that *finite elements are only a mean for the computation, but they are not necessarily kept*. This means that the mesh that defines the domain for a given time step can change for the next time step. *This is the reason for assigning all the problem variables to the particles*. Consequently, the history of the continua remains in the particles. They become the spatial reference and all information of the analysis is kept on them. The solution of the incremental equations of motion of the UL formulation leads to the update of the particle variables in the PFEM.



**Figure 3.2:** The variables are segregated into elements and particles:(a) Transfer of information from particles to nodes, (b) Transfer from nodes to element and (c) Transfer from element to nodes. The way back is also performed when the information is updated and recovered.

This yields to expressing (3.7) and (3.8) as:

$$\mathbf{u}_p^{t+\Delta t} = \mathbf{u}_p^t + \Delta \mathbf{u}_p \quad (3.10)$$

$$\mathbf{S}_p^{t+\Delta t} = \sigma_p^t + \Delta \mathbf{S}_p \quad (3.11)$$

where  $p$  denotes the value that the variable has on a particle.

After a solution of a finite element mesh in one time step, the nodal variables and elemental variables are updated.

Focusing in nodal variables, for simplicity, is quite convenient to consider that every node coincides with a particle. Therefore displacements, velocities and accelerations are variables expressed directly on nodes. These variables are updated directly after the displacements are computed. If some particles do not coincide with nodes, an interpolation of the variables is done to assign the correct value to particles. This case is represented in Figure 3.2.

Elemental variables are more complex, as they are attributes that concern a spatial area of volume. Usually constitutive variables or measures of stress and strain are computed at the integration points of the elements. Therefore, these variables are computed for each single element and then, when the computation for one time step is finished, they are transferred to the particles and updated via the incremental equations (3.10) and (3.11). This introduces some loss of information due to the transfer operations from the

integration points to the particles and from the particles to the integration points. It is important to put special attention on these information transfers.

### 3.5 Updating and transferring information

A big deal in the PFEM is how to accurately transfer the information of the domain to single particles. The domain is described by finite elements and their variables are associated to the integration points. Therefore, the problem is how to transfer the information from finite elements to the particles and viceversa.

As explained in the previous section usually particles coincide with mesh nodes but this is not mandatory. Considering that particles and nodes coincide, the number of nodes can also change from one time step to another when mesh refinement is applied in some areas of the domain. *For simplicity we will assume here onwards that the number of finite elements and particles are kept constant during the time step.*

The first goal is how to transfer element information to particles. Typically variables for an element are: stresses, strains, constitutive variables, material properties, associated volumes, etc. The list increases depending on the type of problem under study.

The incremental equations of motion presented in (3.10) and (3.11) are defined for displacements and stresses. Displacements can be updated directly on nodes, because they are computed on them. On the other hand, element variables have to be transferred from the integration points to the nodes. Element variables are discontinuous from one element to another. It means that a node has a different value of the variable depending on the element that is taken as reference. Stresses are the best example of that ([137], [81]).

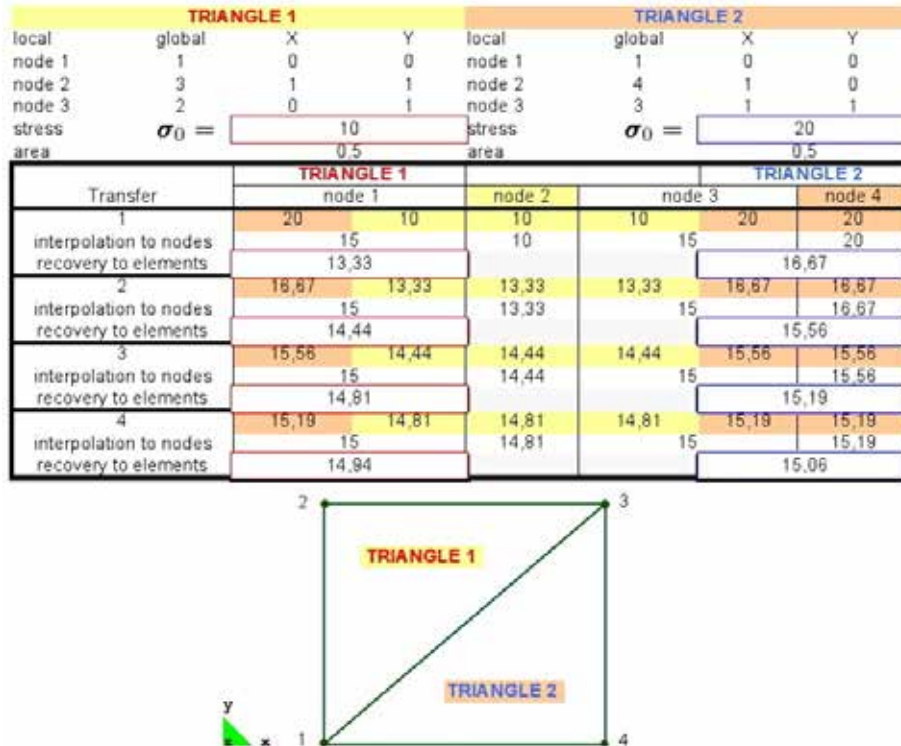
To define a characteristic stress for a node an interpolation of the stresses of the elements that share the node has to be done. This process has a negative consequence: the loss of accuracy. After transferring the information from a mesh to the nodes (or particles), the information that is subsequently recovered in the elements using the nodal information does not coincide with the original one. This is a consequence of the smoothing introduced by the transfer process. Stresses are affected by this smoothing and this endangers the equilibrium of the system. Figure 3.3 illustrates this phenomenon. The figure shows two triangles with the same area and an initial stress of  $\sigma_0^1 = 10$  and  $\sigma_0^2 = 20$ . After four transfer operations element-node-element (in the same mesh) the largest and lowest stress values are almost the same  $\sigma_f \simeq 15$ .

A procedure to minimize this smoothing effect using a conservation of the element information through meshes is presented in the next section.

#### 3.5.1 Smoothing minimization and variable recovery

After getting the solution in one time step, the update of the problem properties is the next thing to do. In the PFEM a new mesh is created and the element information changes. The explanation focuses in the stresses but is applicable to other element variables.

To get a continuous distribution of stresses in a domain, the values of the elements are interpolated onto the nodes. The smoothing reduces the extreme values. The large values



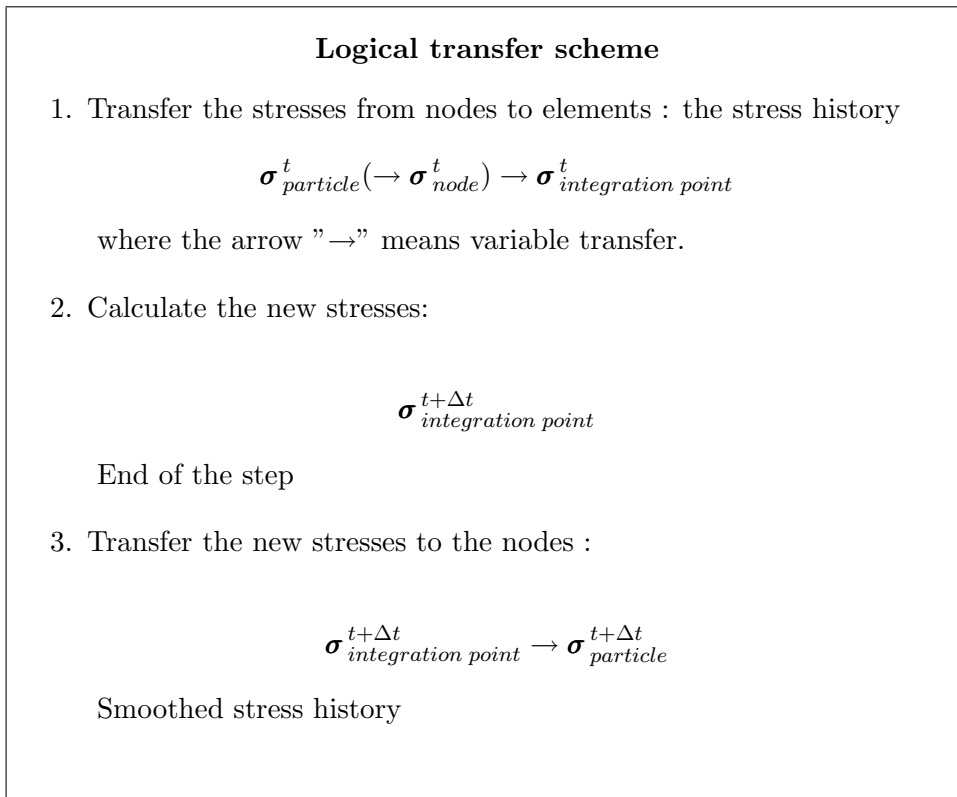
**Figure 3.3:** Interpolation of two triangular elements. Four element-node-element stress transfers are reduced and the low values increase. If the process is repeated many times the stress field will converge to a uniform value (see Figure 3.3).

The method developed in this work is formulated taking in account non-linear problems with large displacements and deformations. Under these considerations, there are not many options for performing the stress smoothing. A typical alternative is the projection of stresses from one mesh to another using former and new integration points. This alternative has not been applied for some reasons. The most important one is that it is not the philosophy of a particle based method. Integration points are associated to elements which are only the base for computing the continuum domain and they change from step to step.

Other reasons are due to the increase of complexity. The information of meshes has to be stored and associated to octree structures. There are also several problems when the geometry changes too much and the projection of the integration points does not find any correspondence between meshes.

For the PFEM a transfer based on particles has been developed. Focusing in the stresses at an intermediate analysis state, the transfer is reduced to the following steps:





Box 3.1: Flowchart of the information transfer: Particles-Elements-Particles

The global process has three steps and the result is a very dissipative transfer. This means that the history of stresses is modified and the variables are not recovered as in a reversible process. A significant error is introduced into the system, modifying substantially the results after some transfers.

Analyzing the transfer scheme of Box 3.1 it is possible to derive a simple modification of the scheme that reduces the error substantially.

Note that the update of the variables is done by adding the increment to the previous variable (3.8). For a particle based method the most natural updating scheme *is to add the increment of the variable on the particle* (3.11). That means, to conserve the information on the particles and transfer only the increment of the variable (i.e. stresses) on particles. There is an important advantage with this small modification. Now, the second transfer (element to node) is done only for the incremental value and not for the total historical amount of stress. This leads to a significant reduction of the smoothing. The trial scheme is:

### PFEM transfer SCHEME

1. Transfer the stresses from nodes to elements : the stress history

$$\sigma_{particle}^t (\rightarrow \sigma_{node}^t) \rightarrow \sigma_{integration\ point}^t$$

where the arrow "→" means variable transfer.

2. Calculate the new stresses:

$$\sigma_{integration\ point}^{t+\Delta t}$$

3. Calculate the stress increment for the step :

$$\sigma_{integration\ point}^{t+\Delta t} = \sigma_{integration\ point}^t + \Delta\sigma_{integration\ point}$$

End of the step

4. Transfer the stress increment to the nodes

$$\Delta\sigma_{integration\ point} \rightarrow \Delta\sigma_{particle}$$

5. Add the nodal stress increment to the historical nodal values :

$$\sigma_{particle}^{t+\Delta t} = \sigma_{particle}^t + \Delta\sigma_{particle}$$

**The process minimizes the smoothing of the stress history**

Box 3.2: Modified flowchart of the information transfer: Particles-Elements-Particles

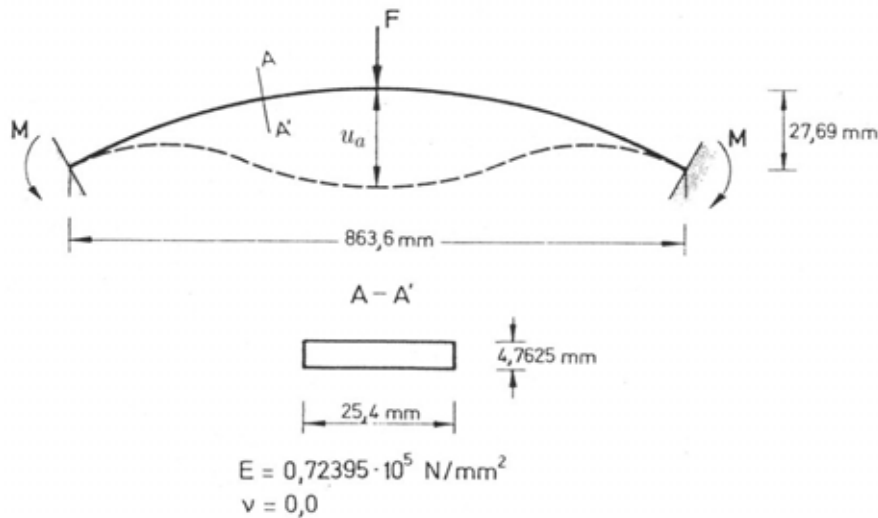
Using the scheme of Box 3.2 the introduced error is not important enough to reduce the accuracy of the analysis. The modified scheme does not reduce completely the smoothing. Nevertheless, the variable does not have a major loss of its history after the transfer and it conserves its properties.

In some specific problems where a high precision is needed it would be better to combine the second scheme with other transfer techniques. Some benchmark problems have been resolved in order to show that the weight of the variables is conserved using the

PFEM scheme. Examples are focused in stress conservation in solid mechanic problems. We compare the standard resolution with the FEM and the resolution with the PFEM. Note that in the PFEM a transfer of the variables is done using the scheme of Box 3.2.

### Clamped shallow circular arch

The shallow arc of Figure 3.4 has been studied with 2D solid elements. Only one half of the arc was discretized using triangular elements for symmetry. A point load acts vertically in the middle of the arc. Both ends of the arc are clamped.



**Figure 3.4:** Clamped shallow circular arch

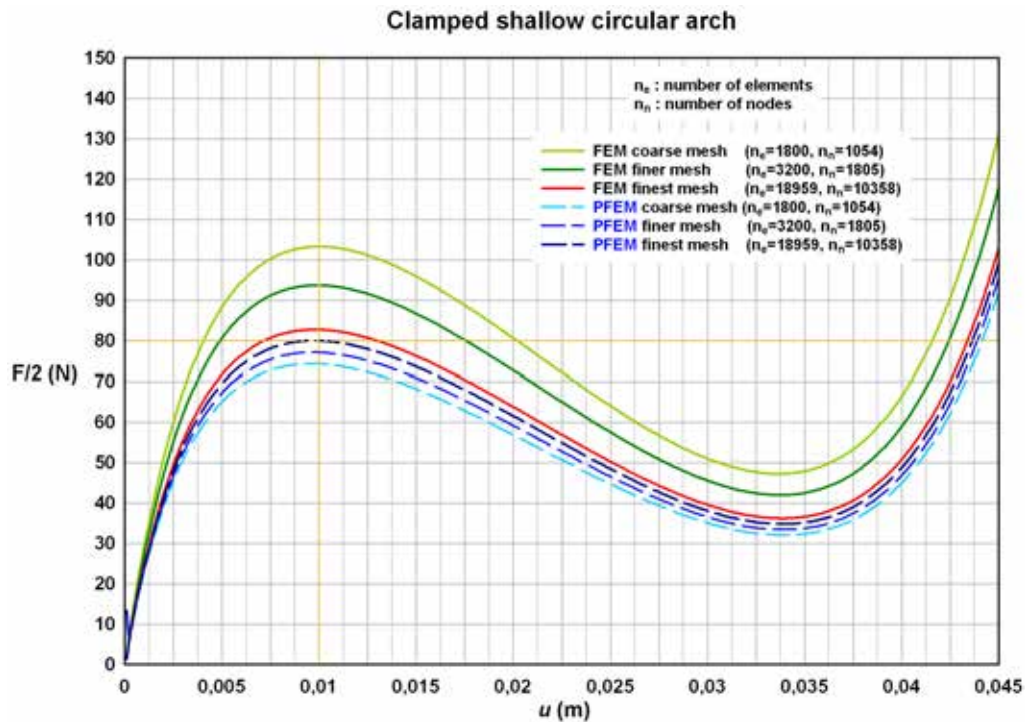
For the simulation instead of an increasing point load, a displacement control has been used. In the upper node in the apex of the arc a vertical displacement has been imposed, with steps of  $u = 0.1 \text{ mm}$ .

With a previous study of the arc the limit load is determined as  $F_{limit} = 160 \text{ N}$ . The computation is performed using one half of the arc only, consequently the reference limit load is  $\frac{F_{limit}}{2} = 80 \text{ N}$ . This load is reached when the vertical displacement in the apex of the arch is  $u_a = 10 \text{ mm}$ .

The diagrams of Figure 3.5 and Figure 3.6 show a plot of the vertical displacement of the upper node in the apex of the arc versus the applied load. As expected, a finer mesh approximates better the result. A comparison between the results with coarse and fine meshes is shown in Figure 3.5. The same problem has been computed using the standard FEM and the PFEM. The main difference between the two methods is that a constant remeshing and updating is done from step to step with the PFEM (450 steps were needed to reach  $u_a = 45 \text{ mm}$ ).

It can be observed in Figure 3.6 that the obtained force with the PFEM is always

lower than for the FEM. The progression when the mesh is refined is also different in the two methods. The FEM approaches the theoretical result with a under bound limit and the PFEM with a upper bound limit. This is due to the smoothing in the transfer of the mechanical variables. This smoothing is however not so important. It can be seen that does not distort the results so much when the mesh is good enough. In the analyzed problem this phenomenon yields a better approximation to the theoretical solution, see [89].



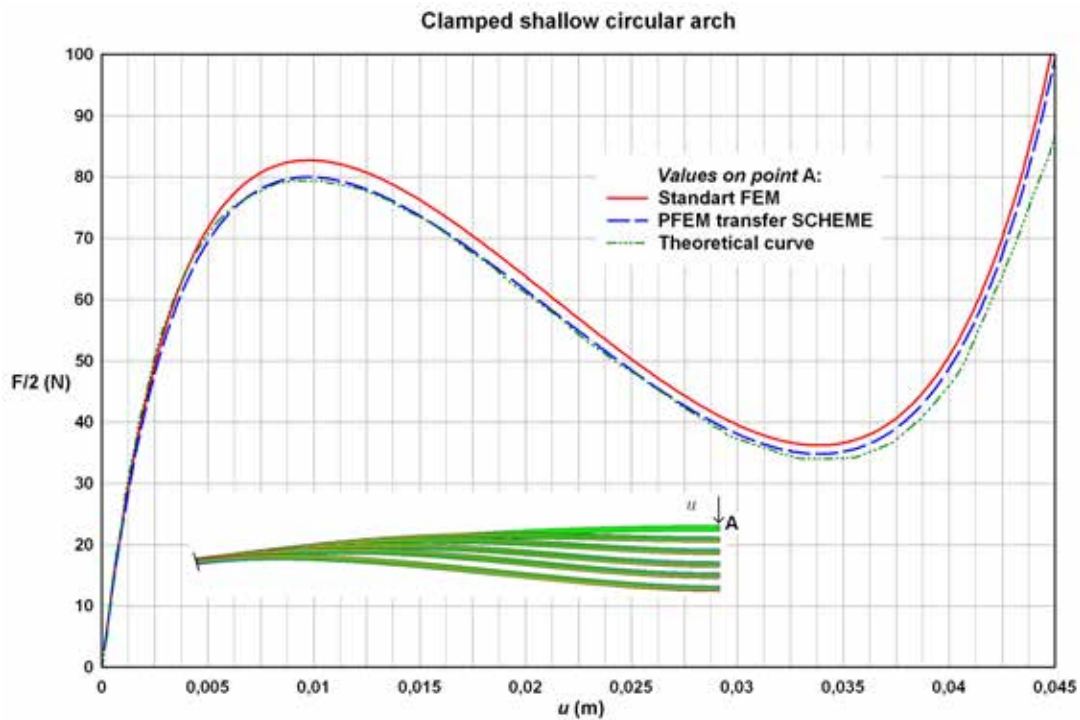
**Figure 3.5:** Six curves representing the solutions with the FEM for different meshes and the solution with the PFEM, for the same finer mesh and using the transfer scheme of Box 3.2

### Beam under a loading-unloading tensile force

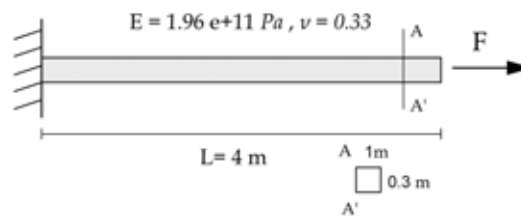
This example focuses in determining how the stress is conserved when the transfer scheme of Box 3.2 is used. In this case the analysis is an linear elastic beam modeled in two dimensions. An horizontal load  $F$  acts at one of the ends and the other end is clamped. The scheme of the problem is shown in Figure 3.7.

The problem consists in applying a static loading and unloading to see the evolution of the stresses. If the stress information is not lost, the initial configuration must be recovered. In Figures 3.8 and 3.9 the computations using the FEM and the PFEM are presented in terms of the displacement. Slightly differences can be appreciated.

Focussing on the stress conservation, a point in the middle of the clamped edge is



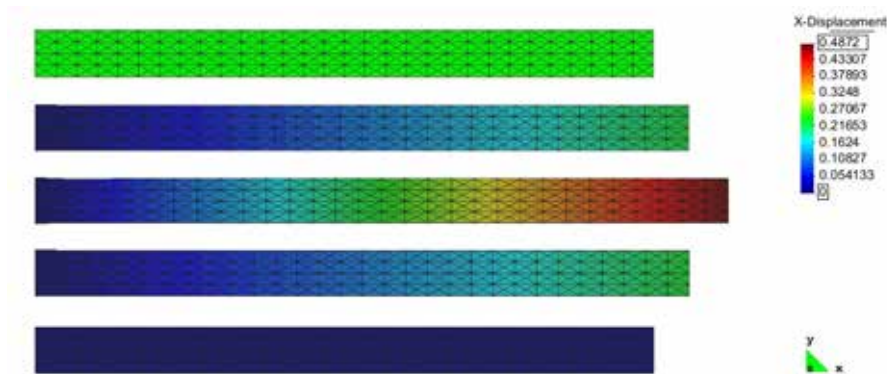
**Figure 3.6:** Clamped shallow circular arch, three curves representing the theoretical solution, the solution with the FEM and with the PFEM, for the same fine mesh of 18959 elements.



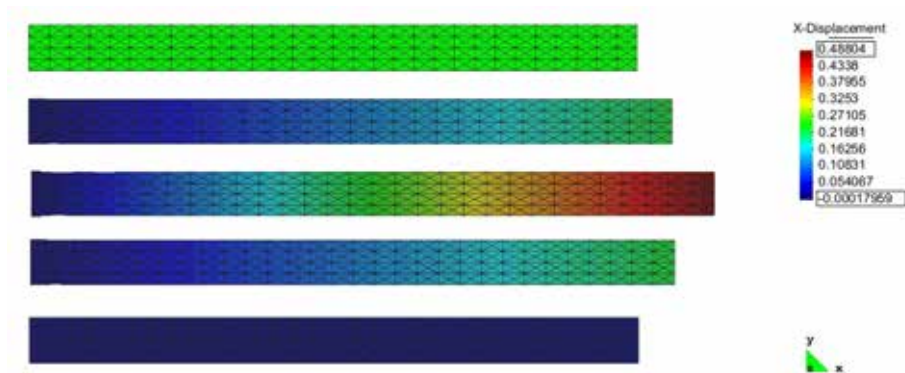
**Figure 3.7:** Elastic beam clamped in one edge and with a load in the other edge.

selected to show the stress-strain evolution in  $x$  and  $y$  directions. Figure 3.10 shows that there is a relevant difference in the  $y$  direction and a small difference in  $x$  direction. In both cases the slope of the curve is smaller using the PFEM. There is also a small difference between the loading and unloading slope. However the loss of information is not big enough to change the behavior related to the force-displacement curve. In each step the equilibrium is reached and the beam recovers elastically, almost in the ideal way.

The last results show the price that it has to be paid for using the PFEM in solid mechanics. Some accuracy is lost due to the information transfer. It is not a critical



**Figure 3.8:** Displacements in an elastic beam after loading and downloading cycle for a tensile load, computed with FEM. Curves show deformations for  $F = 0; 45 \cdot 10^8; 90 \cdot 10^8; 45 \cdot 10^8; 0$  (N).

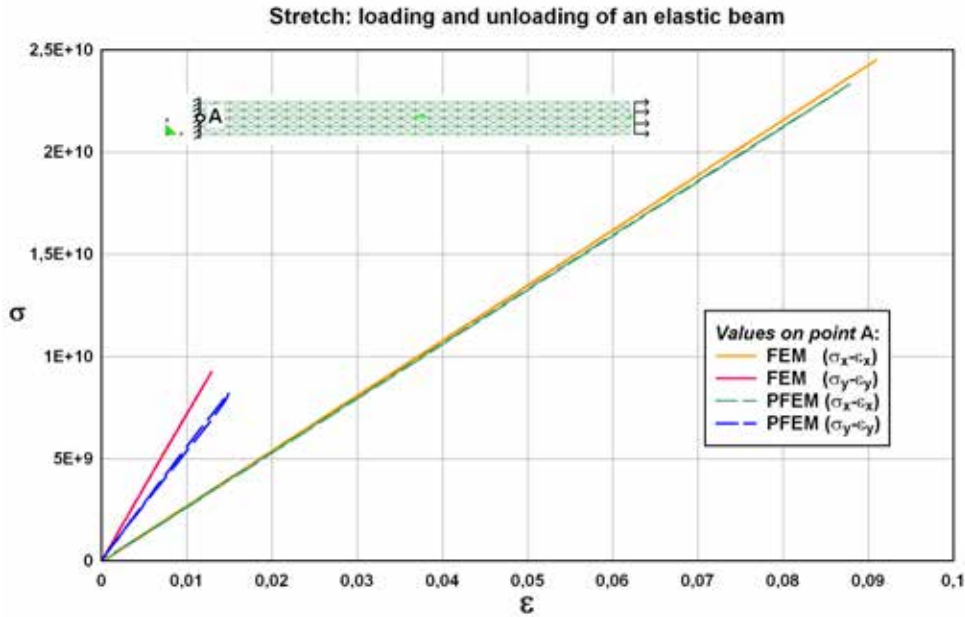


**Figure 3.9:** Displacements in an elastic beam after loading and downloading cycle for a tensile load, computed with PFEM. Curves show deformations for  $F = 0; 45 \cdot 10^8; 90 \cdot 10^8; 45 \cdot 10^8; 0$  (N).

issue but it is important to be aware of it. There are other methods for the transfer and recovery of stresses. Some of them can be found in [17], [58], [136] and [140].

### 3.5.2 Material allocation to particles

There are other variables that have a value in the integration points and are related to elements. The most common ones are these that store all information related to the materials in the domain. This information has to be transferred to nodes too. Firstly there is the need to label the material. Material properties will remain constant if the domain is not mixed with another materials. However material properties are assigned classically to elements. Secondly, the variables related to the constitutive model. They change in the same way as stresses. These variables experience an evolution in time and they have to be updated. The transfer scheme of Box 3.2 is used for updating these variables. However, in some specific cases, extra conditions are needed. See Section 3.7.3.



**Figure 3.10:** Elastic beam after loading and downloading cycle for a tensile load, computed with FEM and PFEM.

For the transfer of the tag for the material another alternative has been developed. It allows more flexibility and to extend the method to other problems like mixtures and solid-fluid interaction.

The material recognition usually is a label that associates every element with a type of material and with a constitutive behavior. Every material has some constant properties, its density  $\delta$ , the *Young* modulus  $E$ , the *Poisson* ratio  $\nu$ , etc. In the PFEM the flag for the material has to be allocated in the particles. In this case, the particles define the material of the domain.

Starting from the beginning, firstly the material from the elements must be assigned to the particles and then one of the schemes can be applied.

### 1.-Material assignment: Elements to Particles

Usually, in a finite element discretization elements define the analysis domain and materials are assigned to each element. In the PFEM these two concepts have a differentiated treatment. The first concept is the analysis domain, that corresponds to all elements and points that belong to a single body  $\mathbf{B}$ . The other concept is the material of each body  $\mathcal{M}_{\mathbf{B}}$ , which assigns some constitutive properties to each element of the body.

In order to explain the PFEM scheme. The first hypothesis is to consider that the complete domain has the same material properties. After creating the finite element mesh the material is assigned to the elements. In the PFEM the material assignment is transferred to particles. All elements surrounding a particle will have the same material and all properties will be the same. When the material assignment is recovered for the

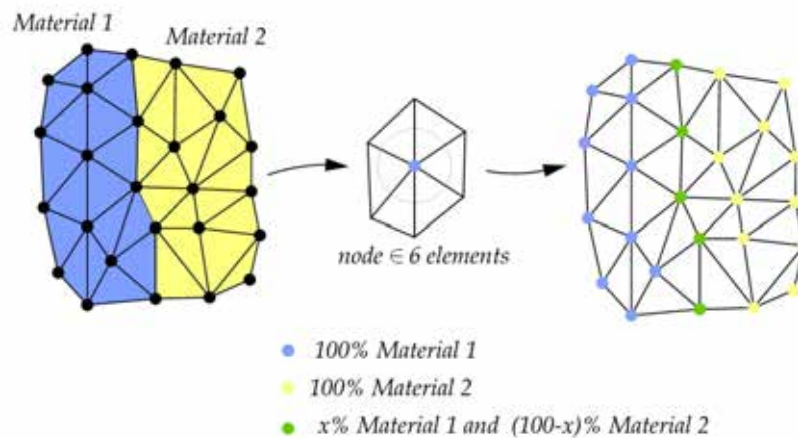
elements the process follows as in the standard FEM.

The second case is that the analysis domain has different material properties. As usual materials will be defined and assigned to elements in the preprocess. Some elements of the same body will have some property values and others a different one. This distribution will define an heterogeneity in the domain. If this heterogeneity is not due to a discontinuity in the material, it will have a region of transition in the border where the material changes. In order to reproduce this transition, the particles that share different element materials have to take this fact into account.

The first option for reproducing the transition is the following: chose the material for each particle depending on the number of neighbor elements of each material type. Then the material of a particle would be one or another depending on the largest number of elements. This option was not considered a good choice.

The second option is to define a percentage of each material depending on the volume defined by elements neighbor to a particle. This means that each particle will contain a percentage of every material around it. This procedure allows to create a transition of the properties in the surface when the material properties are recovered for elements that contain that particle.

In order to clarify this process the assignment steps are listed next. Box 3.3 shows the process used in the PFEM for assigning the material properties to particles:



**Figure 3.11:** Material assignment from elements to particles in a single body domain.  $x$  is determined by the volumes associated to the elements.

## 2.-Material assignment: Particles to Elements

As mentioned before there are several possible assignments of the material from particles to elements, depending on the type of problem to be modeled. When a new mesh is created from a cloud of points (see (4.1)), the properties of the elements have to be transferred to the elements.



### Element to Particle TRANSFER scheme

1. A domain flag is assigned for each independent body.
2. A material flag is assigned for each body  $\mathbf{B}_i$  or region of the body.
3. All bodies in the domain are meshed.
4. Automatically, the materials for each body domain are assigned to the discretizing elements
5. USUAL STORAGE FOR ELEMENT: *Number of Materials* ( $N_{\mathcal{M}}$ ), *Number of Body Domains* ( $N_{\mathbf{B}}$ ) and *Number of Constitutive Properties* ( $N_{\mathcal{C}}$ )
6. Recognition of the elements that share a particle.  
Every node has a certain number of surrounding elements, see Figure 3.11.
7. Assignment a percentage of material to each particle:  
e.g.: If a domain has two materials, *material 1* and *2*, a particle of *material 1* has 100% of *material 1* and 0% of *material 2*  
Figure 3.11 illustrates the process.

Box 3.3: Flowchart for the material assignment from elements to particles

Taking in account that the material is defined for every particle it is possible to choose one of the two distributions:

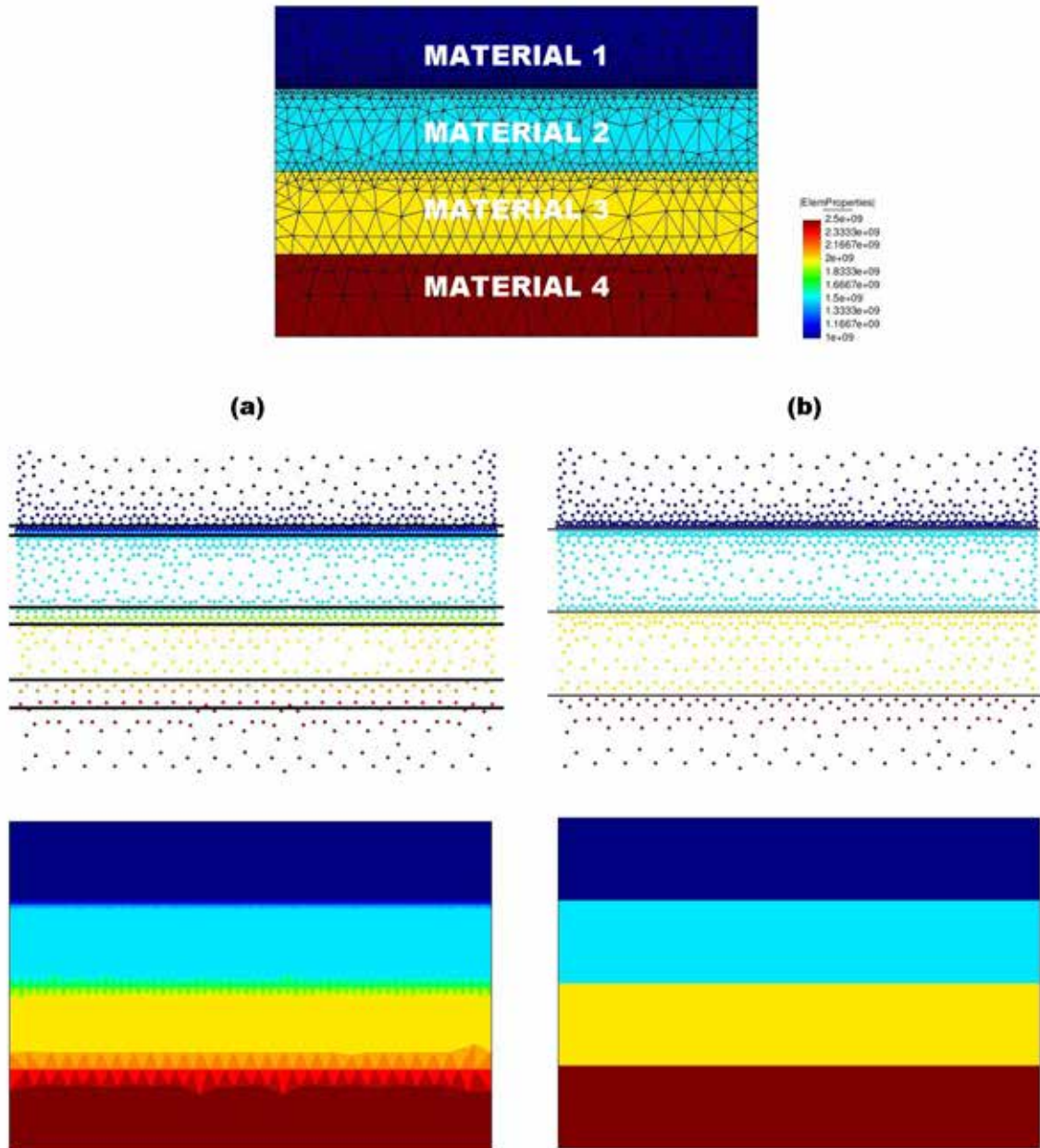
**Distribution 1:** Combining the properties of the material assigned to all particles that belong to a single element. This combination yields an isotropic value for each property in the element. Figure 3.12a.

**Distribution 2:** Assigning to an element the properties that are in common to all particles belonging to that element only. It will not mix any property of the domain. Figure 3.12b.

Box 3.4: Material assignment from particles to elements

Figure 3.12 also shows the dependence of the mesh size for the area of material transition. In many solid mechanics problems the second distribution will be the selected option.

Material assignment has also other consequences in the variables transfer process.



**Figure 3.12:** A domain defined by four different materials, (a) Assignment of materials from particles taking in account a transition in the interface (b) No transition between materials, conserving the discontinuity of material properties

When there are different constitutive laws in the same body domain, the bounds of the material must be very clear. The transfer of stresses and strains must be bounded only inside the same material. The values in the particles that share different materials do not have to be considered in the transfer.

The reason for these statements is clear when very different properties for the materials are involved. The internal variables as well as the strain and stresses are modified artificially in their transfer from particles to elements. In the case of a damage model, damage will appear in the bounds of the material transition if the particles in the material boundaries are considered.

Even so, the second distribution is used for the internal variables transfer, as well as for the mechanical variables. The particles in the material boundaries will not be considered for the assignment of the variables to the elements.

There are some especial situations where the assignment of the material to elements is undetermined. In these cases the material assignment will be ruled by the larger percentage of material assigned to the element nodes.

## 3.6 Solution algorithms

This section is a description of the solution procedures for the non-linear finite element discretized equations. Engineering problems require the consideration of nonlinear effects such as finite deformations or non-linear constitutive laws. The PFEM will use classical solution methods to cope with these different phenomena. Solution methods are distinguished between explicit and implicit. Iterative methods are applied to transient and equilibrium problems. The explanation is going to focus in time-dependent problems.

The first thing to determine is the choice of the time integration procedure. The choice of an explicit or implicit method depends upon the problem at hand. Stability and accuracy are the most relevant properties of time integration schemes. First is better to describe the general properties of these methods:

- In an explicit method the time integration of the momentum equations does not require so solve a non-linear system of equations at each time step. The key feature is that the solution of an equation system is not required when the mass matrix is approximated by a lumped mass matrix. They are easy to implement, since the solution at time  $t_{n+1}$  depends only on known variables at  $t_n$ . The main disadvantage is that explicit methods are conditionally stable, which means that if the time step exceeds a critical value, the solution will grow unboundedly. The critical time step is governed by the *Courant* criterion [32]. The central difference method is the most popular explicit method in computational mechanics and physics.
- In implicit integration schemes a non-linear system of equations has to be solved at each time step. They share many common features with the solution of the equilibrium equations since they have to be combined with linearization procedures (for example, the *Newton-Raphson* procedure). They are much more expensive. The time derivatives are approximated by quantities which also depend on the last time step  $t_n$  and on the unknown values at time  $t_{n+\alpha}$ . The main advantage of implicit schemes is that can be constructed to be unconditionally stable. Hence they can be applied with a much large time step than explicit schemes. The *Newmark* method is the most known implicit integration scheme to solve the equations of motion.

The use of large time steps in implicit methods can counteract the cost of solving a non-linear equation for each time step. However it does not work like this. The time step size depends strongly upon the physical process which is simulated. Large time steps are appropriate when the response of the structure is governed by the low frequency modes, like standard vibration problems. Small time steps will be necessary in case of shock wave problems. For small time steps explicit codes are more suitable. Therefore, the choice of the integration procedure to determine the motion of a body depends on the features of the problem to be solved.

### 3.6.1 Explicit time integration scheme

The central difference scheme is the most popular explicit method in computational mechanics and physics. In this algorithm the velocities  $\dot{\mathbf{u}}$  and the accelerations  $\ddot{\mathbf{u}}$  are approx-

imated by the central difference formulae

$$\begin{aligned}\dot{\mathbf{u}}_n &= \frac{(\mathbf{u}_{n+1} - \mathbf{u}_{n-1})}{2\Delta t} \\ \ddot{\mathbf{u}}_n &= \frac{(\mathbf{u}_{n+1} - 2\mathbf{u}_n + \mathbf{u}_{n-1})}{\Delta t^2}\end{aligned}\quad (3.12)$$

where  $(\cdot)_n$ ,  $(\cdot)_{n+1}$  and  $(\cdot)_{n-1}$  denote values at times  $t_n$ ,  $t_n + \Delta t$  and  $t_n - \Delta t$ , respectively.

Applying these approximations to (3.5), including a damping term  $\mathbf{C} \dot{\mathbf{u}}_n$ , the discrete equations of motion at time  $t_n$  become

$$\mathbf{M}[\ddot{\mathbf{u}}_{n+1} - 2\mathbf{u}_n + \mathbf{u}_{n-1}] + \frac{\Delta t}{2}\mathbf{C}[\mathbf{u}_{n+1} - \mathbf{u}_{n-1}] - (\Delta t)^2\mathbf{F}^{ext}(\mathbf{u}_n) + (\Delta t)^2\mathbf{F}^{int}(\mathbf{u}_n) = 0 \quad (3.13)$$

The last equation is rewritten as a system of equations for the unknown displacements  $\mathbf{u}_{n+1}$  :

$$\left(\mathbf{M} + \frac{\Delta t}{2}\mathbf{C}\right)\mathbf{u}_{n+1} = (\Delta t)^2(\mathbf{F}_n^{ext} - \mathbf{F}^{int}(\mathbf{u}_n)) + \frac{\Delta t}{2}\mathbf{C}[\mathbf{u}_{n-1}] + \mathbf{M}[2\mathbf{u}_n - \mathbf{u}_{n-1}] \quad (3.14)$$

In this system  $\mathbf{C}$  and  $\mathbf{M}$  do not change, hence they can be computed only once at the beginning of the simulation if the finite elements remain the same. Therefore the triangulation of the coefficient matrix  $\mathbf{M} + \frac{\Delta t}{2}\mathbf{C}$  has to be calculated only once too. If  $\mathbf{M}$  and  $\mathbf{C}$  are diagonal matrices, the factorization of  $\mathbf{M} + \frac{\Delta t}{2}\mathbf{C}$  is trivial. Only vector operations are needed to evaluate (3.14) because the nonlinearities enter via  $\mathbf{F}_n^{int}$  and  $\mathbf{F}^{ext}$  on the right-hand side of the equation.

In explicit schemes the definition of the starting values require special treatment. At time  $t_0$  values for  $\mathbf{u}_{-1}$  are needed and have to be determined from the initial conditions  $\mathbf{u}_0$  and  $\dot{\mathbf{u}}_0$ . This is made using *Taylor* expansion series at time  $t_{-1} = t_0 - \Delta t$  as follows

$$\mathbf{u}_{-1} = \mathbf{u}_0 - \Delta t \dot{\mathbf{u}}_0 + \frac{\Delta t}{2}\ddot{\mathbf{u}}_0 \quad (3.15)$$

The accelerations  $\ddot{\mathbf{u}}_0$  at time  $t_0$  are obtained by

$$\ddot{\mathbf{u}}_0 = \mathbf{M}^{-1}[-\mathbf{C} \dot{\mathbf{u}}_0 - \mathbf{F}^{ext}(\mathbf{u}_0) + \mathbf{F}_0^{int}] \quad (3.16)$$

There is a variation of the explicit scheme where the initial conditions and velocities can be applied directly when the algorithm starts. In this scheme the following approximations for the displacements and velocities are used [129]

$$\begin{aligned}
\mathbf{u}_{n+1} &= \mathbf{u}_n - \Delta t \dot{\mathbf{u}}_n + \frac{\Delta t}{2} \ddot{\mathbf{u}}_n \\
\dot{\mathbf{u}}_{n+1} &= \dot{\mathbf{u}}_n + \frac{\Delta t}{2} (\ddot{\mathbf{u}}_n + \ddot{\mathbf{u}}_{n+1})
\end{aligned} \tag{3.17}$$

replacing (3.17) in (3.5) leads to the following equation system

$$\left( \mathbf{M} + \frac{\Delta t}{2} \mathbf{C} \right) \mathbf{u}_{n+1} = \mathbf{F}_{n+1}^{ext} - \mathbf{F}^{int}(\mathbf{u}_n - \Delta t \dot{\mathbf{u}}_n + \frac{\Delta t}{2} \ddot{\mathbf{u}}_n) - \frac{\Delta t}{2} \mathbf{C} [\ddot{\mathbf{u}}_n] \tag{3.18}$$

In (3.18), the right-hand side depends only upon the vectors which have to be evaluated at time  $t_n$  and on the known loading function  $\mathbf{F}^{ext}$  evaluated at  $t_{n+1}$ .

The explicit scheme is conditionally stable. For non-linear problems an estimation of the critical time step is [13]:

$$\Delta t \leq \delta \frac{h}{c_L} \tag{3.19}$$

where  $h$  is the characteristic length of the smallest element in the mesh,  $c_L$  is the wave speed of a compression wave in a solid and  $\delta$  is a reduction factor which has to be chosen empirically for each problem ( $0.2 < \delta < 0.9$ ). In a linear elastic medium  $c_L$  can be computed by

$$c_L = \frac{3K(1-\nu)}{\rho(1+\nu)} \tag{3.20}$$

with the material *bulk modulus*  $K$ , *poisson ratio*  $\nu$  and the density  $\rho$ . For linear problems the critical time step is estimated by the *Courant* criterion [32].

Time integration with an explicit scheme in the PFEM requires some modifications in the domain treatment. For moderate speed problems a very small time step would mean a very slow increase of the kinematic variables. The readaptation of the domain and the particles would not be necessary every time step. The explicit integration should be used in the problems where the evaluation of other parameters (i.e. contact between solid domains or complex constitutive laws) will consume a small portion of time compared with the resolution of the linear system of equations associated to the problem. On the contrary an implicit integration scheme should be used for the resolution of the problem.

### 3.6.2 Implicit time integration scheme

To describe the implicit time integration procedure it is necessary to start with the discrete momentum equation (2.226). That was defined as the residual equation of the displacement-based UL formulation in (3.5):

$$0 = \mathbf{r}(\mathbf{u}_{n+1}, t_{n+1}) = \mathbf{M} \ddot{\mathbf{u}}_{n+1} - \mathbf{F}^{ext}(\mathbf{u}_{n+1}, t_{n+1}) + \mathbf{F}^{int}(\mathbf{u}_{n+1}, t_{n+1}) \tag{3.21}$$

where  $\mathbf{r}(\mathbf{u}_{n+1}, t_{n+1})$  is the residual vector. The discrete equations for the implicit update of the equations of motion are nonlinear algebraic equations in the nodal displacements  $\mathbf{u}_{n+1}$ .

The nodal acceleration in the implicit equations are approximated using the classical *Newmark*  $\beta$  equations. In the *Newmark* method, the updated displacements and velocities are expressed as follows

$$\mathbf{u}_{n+1} = \tilde{\mathbf{u}}_{n+1} + \beta \Delta t^2 \ddot{\mathbf{u}}_{n+1} \quad \text{where} \quad \tilde{\mathbf{u}}_{n+1} = \mathbf{u}^n + \beta \dot{\mathbf{u}}^n + \frac{\Delta t^2}{2}(1 - 2\beta) \ddot{\mathbf{u}}^n \quad (3.22)$$

$$\dot{\mathbf{u}}_{n+1} = \tilde{\dot{\mathbf{u}}}_{n+1} + \gamma \Delta t \ddot{\mathbf{u}}_{n+1} \quad \text{where} \quad \tilde{\dot{\mathbf{u}}}_{n+1} = \dot{\mathbf{u}}^n + (1 - \gamma) \Delta t \ddot{\mathbf{u}}^n \quad (3.23)$$

From the previous equations we obtain

$$\ddot{\mathbf{u}}_{n+1} = \frac{1}{\beta \Delta t^2} (\mathbf{u}_{n+1} - \tilde{\mathbf{u}}_{n+1}) \quad \text{when} \quad \beta > 0 \quad (3.24)$$

Solving the system using the *Newmark*  $\beta$  equations, the discrete momentum equation at time step  $n + 1$  (3.21) can be written as

$$0 = \mathbf{r} = \frac{1}{\beta \Delta t^2} \mathbf{M} (\mathbf{u}_{n+1} - \tilde{\mathbf{u}}_{n+1}) - \mathbf{F}^{ext}(\mathbf{u}_{n+1}, t_{n+1}) + \mathbf{F}^{int}(\mathbf{u}_{n+1}, t_{n+1}) \quad (3.25)$$

The linearization of (3.25) with the *Newton-Raphson* method yields an iterative procedure. It starts with the generalized *Taylor* expansion of the residual about the current value of the nodal displacements  $\mathbf{u}_{n+1}^\nu$  (2.155):

$$\mathbf{r}(\mathbf{u}_{n+1}^\nu, t_{n+1}) + \frac{\partial \mathbf{r}(\mathbf{u}_{n+1}^\nu, t_{n+1})}{\partial \mathbf{u}} \Delta \mathbf{u} + \mathcal{O}(\Delta \mathbf{u}^2) = 0 \quad (3.26)$$

where the iteration number is indicated by the superscript  $(.)^\nu$  and  $\mathbf{u}_{n+1}^\nu$  is the vector of displacements for the iteration  $\nu$  at time step  $t_{n+1} = t_n + \Delta t$ .

The linearized model of the nonlinear equations becomes

$$\mathbf{r} - \mathbf{A} \Delta \mathbf{u} = 0 \quad (3.27)$$

with the incremental displacements for the iteration  $\nu + 1$  computed as

$$\mathbf{u}^{\nu+1} = \mathbf{u}^\nu + \Delta \mathbf{u} \quad (3.28)$$

Matrix  $\mathbf{A}$  is *Jacobian* matrix of the system of equations, which can be expressed as

$$\mathbf{A} = \frac{\partial \mathbf{r}}{\partial \mathbf{u}} = \frac{\mathbf{M}}{\beta \Delta t^2} + \frac{\partial \mathbf{F}^{int}}{\partial \mathbf{u}} - \frac{\partial \mathbf{F}^{ext}}{\partial \mathbf{u}} \quad (3.29)$$

The finite element discretization of the weak form of the balance momentum equation (2.180) was shown in (2.232) and (2.224). Applying these results to obtain the *Jacobian* matrix (3.29) gives

$$\mathbf{A} = \frac{\mathbf{M}}{\beta \Delta t^2} + (\mathbf{K}^{mat} + \mathbf{K}^{geo}) - \mathbf{K}^{ext} \quad \text{for } \beta > 0 \quad (3.30)$$

where  $\mathbf{K}^{ext}$  is the matrix that results from the linearization of the external loads. Usually  $\mathbf{K}^{ext} = \mathbf{0}$  if the loads do not change with the configuration of the body. Other loads like pressure or contact forces are always normal or tangential to the surfaces. Therefore the nodal external forces change as the surface moves. These effects are accounted for the *Jacobian* matrix of the nodal external forces,  $\mathbf{K}^{ext}$  (also called the load stiffness matrix).

The finite element matrices for equation (3.30) are expressed as follows

$$\mathbf{M} = \bigcup_{e=1}^{n_e} \sum_{I=1}^{n_n} \sum_{K=1}^{n_n} \int_{\varphi(\Omega_e)} N_I \rho N_K dV \mathbf{I} \quad (3.31)$$

$$\mathbf{K}^{geo} = \bigcup_{e=1}^{n_e} \sum_{I=1}^{n_n} \sum_{K=1}^{n_n} \int_{\varphi(\Omega_e)} (\nabla_{\bar{\mathbf{x}}} N_I^T) \bar{\boldsymbol{\sigma}}_e \nabla_{\bar{\mathbf{x}}} N_K \mathbf{I} dV \quad (3.32)$$

$$\mathbf{K}^{mat} = \bigcup_{e=1}^{n_e} \sum_{I=1}^{n_n} \sum_{K=1}^{n_n} \int_{\varphi(\Omega_e)} \bar{\mathbf{B}}_{0I}^T \bar{\mathbf{D}}^M \bar{\mathbf{B}}_{0K} dV \quad (3.33)$$

where the internal and external forces are

$$\mathbf{F}^{int} = \bigcup_{e=1}^{n_e} \sum_{I=1}^{n_n} \int_{\varphi(\Omega_e)} \mathbf{B}_{0I}^T \boldsymbol{\sigma}_e dV \quad (3.34)$$

$$\mathbf{F}^{ext} = \bigcup_{e=1}^{n_e} \sum_{I=1}^{n_n} \int_{\varphi(\Omega_e)} \rho \bar{\mathbf{b}} \mathbf{N}_I dV + \bigcup_{r=1}^{n_r} \sum_{I=1}^{n_m} \int_{\varphi(\Gamma_r)} N_I \bar{\mathbf{t}} ds \quad (3.35)$$

Matrix  $\mathbf{K}^{ext}$  is computed from the gradient of  $\mathbf{F}^{ext}$ .

Several aspects must be taken in account in the implicit integration. The first one it was mentioned in the introduction. The determination of the motion of a body depends on the features of the problem to be solved. The time step has to be the one that represents correctly the period for the phenomena under analysis. When the correct time stepping is



not used, the results are completely wrong and the computed response will differ strongly from the physical solution. Usually the incorrect time step does not capture the physical response and produces an increase of the energy in the system.

The *Newmark* method has different characteristics depending on the values taken for the parameters  $\beta$  and  $\gamma$ . These values determine the order and the accuracy of the method. They have to be chosen with care, taking in account the following:

$$\begin{aligned}
 \beta = 0, \gamma = \frac{1}{2} & \quad \text{explicit central difference method} \\
 \beta = \frac{1}{4}, \gamma = \frac{1}{2} & \quad \text{undamped trapezoidal rule} \\
 \gamma > \frac{1}{2} & \quad \text{numerically damped integrator with damping proportional to } \gamma - \frac{1}{2}
 \end{aligned} \tag{3.36}$$

For stability:

$$\begin{aligned}
 & \text{Unconditionally stable for } \beta \geq \frac{\gamma}{2} \geq \frac{1}{4} \\
 & \text{Conditional stability when} \\
 & w_{\max} \Delta t = \frac{\xi \bar{\gamma} + [\bar{\gamma} + \frac{1}{4} - \beta + \xi^2 \bar{\gamma}^2]^{\frac{1}{2}}}{(\frac{\gamma}{2} - \beta)}, \quad \bar{\gamma} \equiv \gamma - \frac{1}{2} \geq 0
 \end{aligned} \tag{3.37}$$

$$\tag{3.38}$$

where  $\xi$  is the fraction of critical damping for the frequency  $w_{\max}$ .

Looking at equation (3.36), the *Newmark* integration scheme is undamped for  $\gamma = \frac{1}{2}$  and  $\beta = \frac{1}{4}$ . Even so, it has a tendency for high frequency noise to persist in the solution. This instability usually manifests itself in that the response grows, and the energy and momentum are not conserved.

### 3.6.3 Energy conservation methods

There are several aspects associated with the energy conservation of the system. The goal is to find the best energy-conserving method which maintains the energy of the system without degrading the accuracy so much. The stability of the system has to be maintained, although the solution is inaccurate due to a too large time step. Long time intervals produce instability in the classical trapezoidal rule (3.36).

Various research efforts have been invested to find a stable and energy conserving time integration scheme. A review can be seen in [67]. In essence, they can be classified in four categories. In the first case, new time integration procedures were proposed to introduce some numerical damping. An example of this type of techniques is the generalized alpha-method [26], [67]. In the second category, Lagrange multipliers are employed to

enforce momentum and energy conservation [123]. In the third case, energy conservation is enforced algorithmically; for example by using weighted stresses (e.g. [122], [68]). The fourth class of methods uses a sub-step integration to stabilize the trapezoidal rule in [11], but this increases considerably the computational effort per step.

There are several alternative for finding a method that is accurate and efficient for many applications. The selected option chosen in this work for the PFEM is the generalized alpha-method as described in [67]. It allows algorithmic damping in combination with second-order accuracy. It also has minimal numerical dissipation for lower modes and maximal numerical dissipation for the higher modes in linear structural dynamics. The method can be extended to non-linear dynamics using a combination of the internal forces at the beginning and at the end of each time step.

### Generalized alpha-method

The linear equations for a damped system are the ones expressed in (3.25) including a damping term  $\mathbf{C} \dot{\mathbf{u}}_{n+1}$ :

$$\mathbf{M} \ddot{\mathbf{u}}_{n+1} + \mathbf{C} \dot{\mathbf{u}}_{n+1} - \mathbf{F}^{ext}(\mathbf{u}_{n+1}) + \mathbf{F}^{int}(\mathbf{u}_{n+1}) = 0 \quad (3.39)$$

Subscript  $(\cdot)_{n+1}$  means that the quantity has to be evaluated at time  $t_{n+1}$ . Typically  $\mathbf{F}^{int}(\mathbf{u}_{n+1})$  can be expressed in the discrete form for a linear problem as  $\mathbf{K} \mathbf{u}_{n+1}$ .

The damping matrix  $\mathbf{C}$  is typically chosen as a combination of matrices  $\mathbf{M}$  and  $\mathbf{K}$ . In our work we have chosen the *Rayleigh* damping (e.g. [13] and [9]):

$$\mathbf{C} = a_1 \mathbf{M} + a_2 \mathbf{K} \quad (3.40)$$

where  $a_1$  and  $a_2$  are arbitrary parameters. Taking in account the physical properties it has more sense to choose a damping matrix proportional to the stiffness of the material than to the mass. In this case  $a_2 \neq 0$  introduces an increased artificial damping.

*Newmark* algorithms, used with  $\beta$  and  $\gamma$  parameters, which include numerical damping only guarantee first order accuracy. In contrast,  $\alpha$ -Methods allow algorithmic damping in combination with second-order accuracy. In these methods, the semidiscrete equation of motion (3.39) is applied to a general mid-point, instead of the end-point as in the classical *Newmark* method. The modified equations of motion of non-linear dynamics become

$$\mathbf{M} \ddot{\mathbf{u}}_{n+1-\alpha_m} + \mathbf{C} \dot{\mathbf{u}}_{n+1-\alpha_f} + \mathbf{F}_{n+1-\alpha_f}^{int} = \mathbf{F}_{n+1-\alpha_f}^{ext} \quad (3.41)$$

where subscripts denote the time discrete combinations of the accelerations, velocities and the external forces as follows

$$\begin{aligned} \ddot{\mathbf{u}}_{n+1-\alpha_m} &= (1 - \alpha_m) \ddot{\mathbf{u}}_{n+1} + \alpha_m \ddot{\mathbf{u}}_n \\ \dot{\mathbf{u}}_{n+1-\alpha_f} &= (1 - \alpha_f) \dot{\mathbf{u}}_{n+1} + \alpha_f \dot{\mathbf{u}}_n \\ \mathbf{u}_{n+1-\alpha_f} &= (1 - \alpha_f) \mathbf{u}_{n+1} + \alpha_f \mathbf{u}_n \end{aligned} \quad (3.42)$$

$$\begin{aligned}\mathbf{F}_{n+1-\alpha_f}^{ext} &= (1 - \alpha_f) \mathbf{F}_{n+1}^{ext} + \alpha_f \mathbf{F}_n^{ext} \\ \mathbf{F}_{n+1-\alpha_f}^{int} &= (1 - \alpha_f) \mathbf{F}_{n+1}^{int} + \alpha_f \mathbf{F}_n^{int}\end{aligned}\quad (3.43)$$

Classically the internal forces are developed as in (3.42), and they are evaluated as a combination at the bounds of each time step:

$$\mathbf{F}_{n+1-\alpha_f}^{int} = (1 - \alpha_f) \mathbf{F}_{n+1}^{int} + \alpha_f \mathbf{F}_n^{ext} = (1 - \alpha_f) \mathbf{F}^{int}(\mathbf{u}_{n+1}) + \alpha_f \mathbf{F}^{int}(\mathbf{u}_n) \quad (3.44)$$

As an alternative, the generalized mid-point displacement  $\mathbf{u}_{n+1-\alpha_f}$  can be applied to calculate the algorithmic internal forces as

$$\mathbf{F}_{n+1-\alpha_f}^{int} = \mathbf{F}^{int}(\mathbf{u}_{n+1-\alpha_f}) \quad (3.45)$$

Both approaches are equivalent for the linear case.

The generalized mid-point velocities and accelerations for the *Newmark* approximations in time (3.23) and (3.24) are given as functions of the end-point displacements  $\mathbf{u}_{n+1}$  as

$$\ddot{\mathbf{u}}_{n+1-\alpha_m} = \frac{1 - \alpha_m}{\beta \Delta t^2} (\mathbf{u}_{n+1} - \mathbf{u}_n) - \frac{1 - \alpha_m}{\beta \Delta t} \dot{\mathbf{u}}_n - \frac{1 - \alpha_m - 2\beta}{2\beta} \ddot{\mathbf{u}}_n \quad (3.46)$$

$$\dot{\mathbf{u}}_{n+1-\alpha_f} = \frac{(1 - \alpha_f)\gamma}{\beta \Delta t} (\mathbf{u}_{n+1} - \mathbf{u}_n) - \frac{(1 - \alpha_f)\gamma - \beta}{\beta} \dot{\mathbf{u}}_n - \frac{(1 - \alpha_f)(\gamma - 2\beta)}{2\beta} \Delta t \ddot{\mathbf{u}}_n \quad (3.47)$$

The linearization of the modified balance momentum equation (3.41) considering  $\mathbf{M}$  and  $\mathbf{C}$  as constant, gives the effective *Jacobian* matrix of the system as

$$\mathbf{A} = \frac{1 - \alpha_m}{\beta \Delta t^2} \mathbf{M} + \frac{(1 - \alpha_f)\gamma}{\beta \Delta t} \mathbf{C} + (\mathbf{K}_{n+1-\alpha_f}^{mat} + \mathbf{K}_{n+1-\alpha_f}^{geo}) - \mathbf{K}_{n+1-\alpha_f}^{ext} \quad (3.48)$$

where the tangent matrices  $\mathbf{K}_{n+1-\alpha_f}^{mat}$  and  $\mathbf{K}_{n+1-\alpha_f}^{geo}$  are the result of the linearization of the weak form (2.52) evaluated at the mid-point of the deformation state  $\mathbf{u}_{n+1-\alpha_f}$  relative to the end-point of the deformation state  $\mathbf{u}_{n+1}$ . The stiffness matrices can be also expressed as the derivative of the internal forces using the generalized mid-point displacement:

$$\mathbf{K}_{n+1-\alpha_f}(\mathbf{u}_{n+1}^\nu) = \frac{\partial \mathbf{F}^{int}(\mathbf{u}_{n+1}^\nu)}{\partial \mathbf{u}_{n+1}^\nu} \quad (3.49)$$

where  $\mathbf{u}_{n+1}^\nu$  is the vector of displacements in the iteration  $\nu$  at time step  $t+1$ .

In the case of follower loads,  $\mathbf{K}_{n+1-\alpha_f}^{ext}$  is the tangent matrix resulting from the linearization of the external forces in the same way as (3.49).

**Table 3.1:** Overview of time integration schemes [68]

Method	Algorithmic parameters	$\mathbf{F}_{n+1-\alpha_f}^{int}$
Trapezoidal Rule	$\alpha_m = \alpha_f = 0, \quad 2\beta = \gamma = 1/2$	(3.44)
Mid-Point Rule	$\alpha_m = \alpha_f = 2\beta = \gamma = 1/2$	(3.45)
Newmark Method	$\alpha_m = \alpha_f = 0$	(3.44)
Bossak- $\alpha$ Method	$\alpha_f = 0$	(3.44)
Hilber- $\alpha$ Method <sub>1</sub>	$\alpha_m = 0$	(3.44)
Hilber- $\alpha$ Method <sub>2</sub>	$\alpha_m = 0$	(3.45)
Generalized- $\alpha$ Method <sub>1</sub>	$\dots$	(3.44)
Generalized- $\alpha$ Method <sub>2</sub>	$\dots$	(3.45)

The choice of the free algorithmic parameters define different classical integration schemes developed as integration methods. They are summarized in Table 3.1.

A stability analysis shows that the generalized  $\alpha$  - *method* is unconditional stable by setting the following parameters:

$$\beta = \frac{1}{4}(1 + \alpha_m + \alpha_f)^2, \quad \gamma = \frac{1}{2} + \alpha_m + \alpha_f \quad \text{and} \quad 0 < \alpha_m < \frac{1}{3} \quad 0 < \alpha_f < \frac{1}{3} \quad (3.50)$$

For the PFEM is important to have an energy conserving integration method that can model the ideal behavior of solid dynamics. At the same time, efficiency is also needed for many applications. The *Bossak- $\alpha$*  method [130] is the one of the simplest to implement. It has only one free algorithmic parameter  $\alpha_m$  on the accelerations in order to control the numerical damping imposed to the system. For  $\alpha_m = 0$  there is no damping and the operator is the trapezoidal rule, while for  $\alpha_m = \frac{1}{3}$  a significant damping is introduced. This operator is used primarily because of the slight numerical damping it offers. It is also a good help when an automatic time stepping scheme is applied. Each change in the time step introduces some slight noise in the solution. Hence a small numerical damping seems a good choice to quickly remove this high frequency noise without having any significant effect on the meaningful, lower frequency, response.

For most problems this method is efficient enough for the dissipation of the high frequencies that persist in the solution.

#### 3.6.4 Flowchart for implicit time integration in the PFEM

Figure 3.13 shows the flowchart for implicit time integration for the PFEM. The flowchart is for a general implicit method. In fact the generalized alpha-method is used but the scheme is defined for the standard *Newton* method with the *Newmark*  $\beta$ -equations.

### SOLUTION METHOD (NEWTON'S METHOD / NEWMARK $\beta$ -EQUATIONS)

Flowchart IMPLICIT SOLUTION in the PFEM: INPUT : INITIAL OR PREVIOUS STEP CONDITIONS

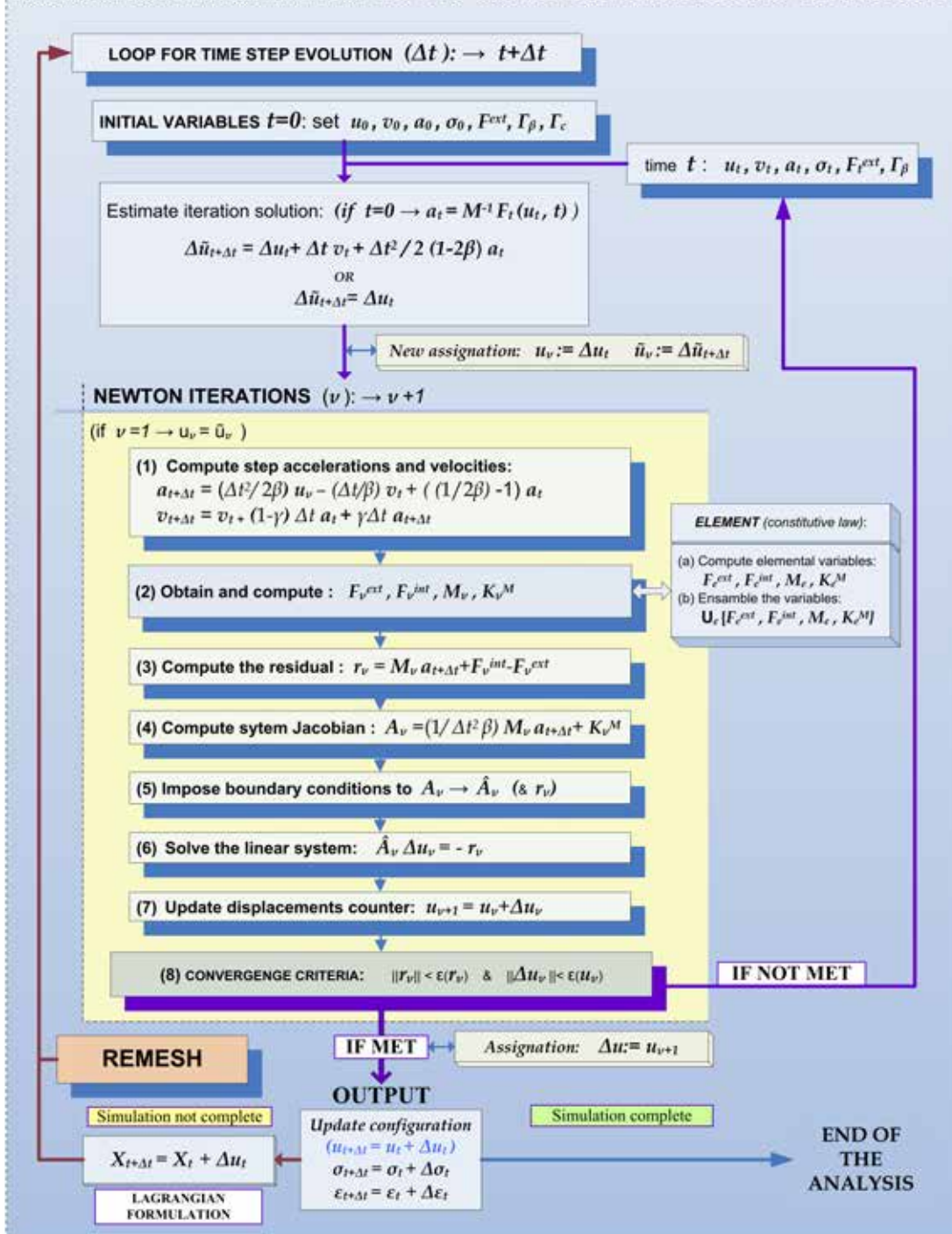


Figure 3.13: Solution scheme for the PFEM

### 3.7 Constitutive laws: Numerical implementation

The general theory for linear and non-linear constitutive laws was introduced in Section 2.4. Some particular aspects have to be clarified regarding the numerical implementation. The numerical definition of the constitutive parameters, the numerical scheme for the constitutive resolution and some topics concerning with the PFEM will be introduced in the next sections.

#### 3.7.1 Damage model in the PFEM

This section explains the numerical implementation of the constitutive equations used in the PFEM for solid mechanics. The continuum formulation for these equations was explained in Section 2.4 .

The implemented model is a damage model particularized for each material through the definition of a specific damage surface (2.112) for each case. After the definition of  $\zeta = f(\bar{\tau})$  the determination of the initial threshold value  $\zeta_0 = f(\bar{\tau}_0) \equiv c^{\max}$  is required. The results from a uniaxial tension test are needed to obtain the maximum tensile strength of the material  $f'_t = \bar{\tau}_t^{\max}$ . This measure will give the first threshold value for the models described in (2.145), (2.148) and (2.150). As follows from equation (2.152):

$$c^{\max} = f(\bar{\tau}_0) = \bar{\tau}^{\max} = f'_t \quad (3.51)$$

For the *Simo and Ju* model equation (2.142) gives:

$$c^{\max} = f(\bar{\tau}_0) = \frac{\bar{\tau}_t^{\max}}{\sqrt{\mathfrak{E}}} \cong \frac{f'_t}{\sqrt{E}} \quad (3.52)$$

where  $E$  is the *Young* modulus of the material in the initial state.

Parameter  $A$ , defined by the fracture energy of the material, included in the softening models (2.122) and (2.123) is determined for all cases by (2.151):

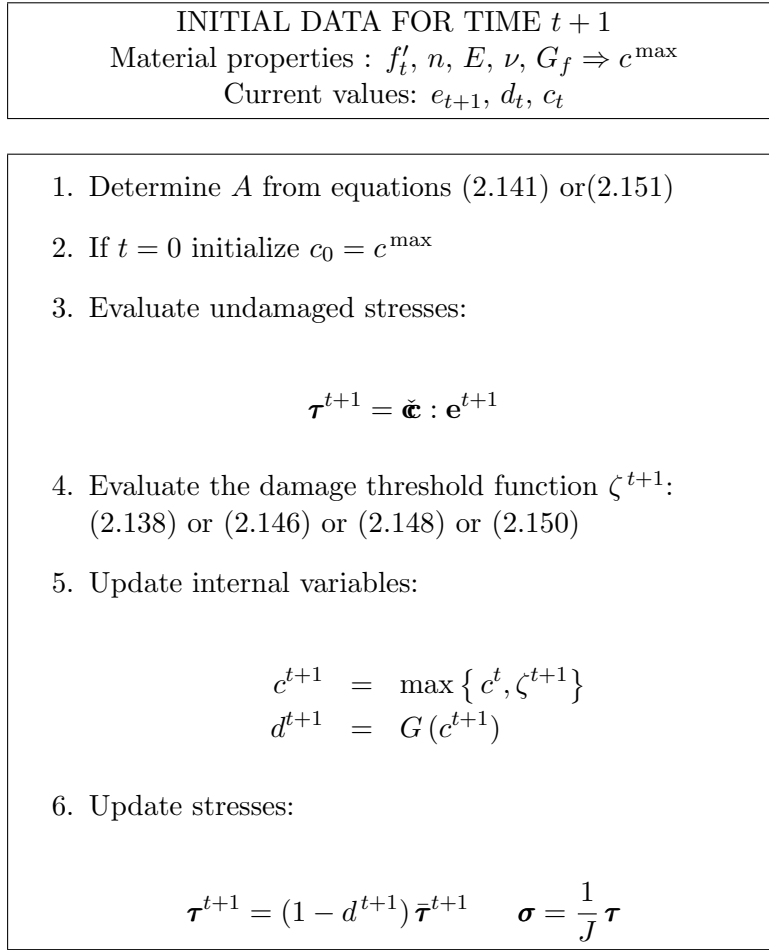
$$A = \left( \frac{G_f E}{l_e (\zeta_0)^2} - \frac{1}{2} \right)^{-1} \geq 0 \quad (3.53)$$

For the *Simo and Ju* model equation (2.141) gives:

$$A = \left( \frac{G_f}{l_e (\zeta_0)^2} - \frac{1}{2} \right)^{-1} \geq 0 \quad (3.54)$$

The process to obtain the parameter  $A$  can be found in [98].

The flowchart of the numerical implementation of the model is presented in Box 3.5.



Box 3.5: Flowchart for the numerical implementation of the isotropic damage model

The flowchart of Box 3.5 is inserted inside the solution method, (see chapter 3.6). In each iteration a solution for the stress tensor is given for every integration point of the finite element mesh.

### 3.7.2 Perturbation method to obtain the tangent constitutive tensor

To complete the numerical implementation of a non-linear analysis, the tangent constitutive tensor is needed. It provides the relation between the increment of stresses and the increment of strains. The convergence of the method depends on the accuracy of this tensor. A better approximation to the tangent tensor will yield a higher convergence rate.

Sometimes the analytical expression for this tensor is very difficult to obtain and there are some cases where it is impossible. The expression of the tangent in (2.133) or (2.153) usually is only valid for infinitesimal deformations, giving a bad convergence rate in other cases.

In order to solve these two negative aspects a numerical procedure is used to obtain the tangent constitutive tensor for large displacements and for any damage surface. The procedure is based on performing a derivation numerically. This is called a Perturbation Method.

The tangent constitutive tensor is defined as follows:

$$\dot{\boldsymbol{\tau}} = \check{\boldsymbol{\mathfrak{C}}} : \dot{\boldsymbol{\epsilon}} \quad (3.55)$$

The perturbation method consist in defining  $n$  small variations, or perturbations, of the strain vector coefficients  $\delta e_k$  to obtain  $n$  stress vectors  $\delta\boldsymbol{\tau}$  that configure the columns of the tangent matrix.

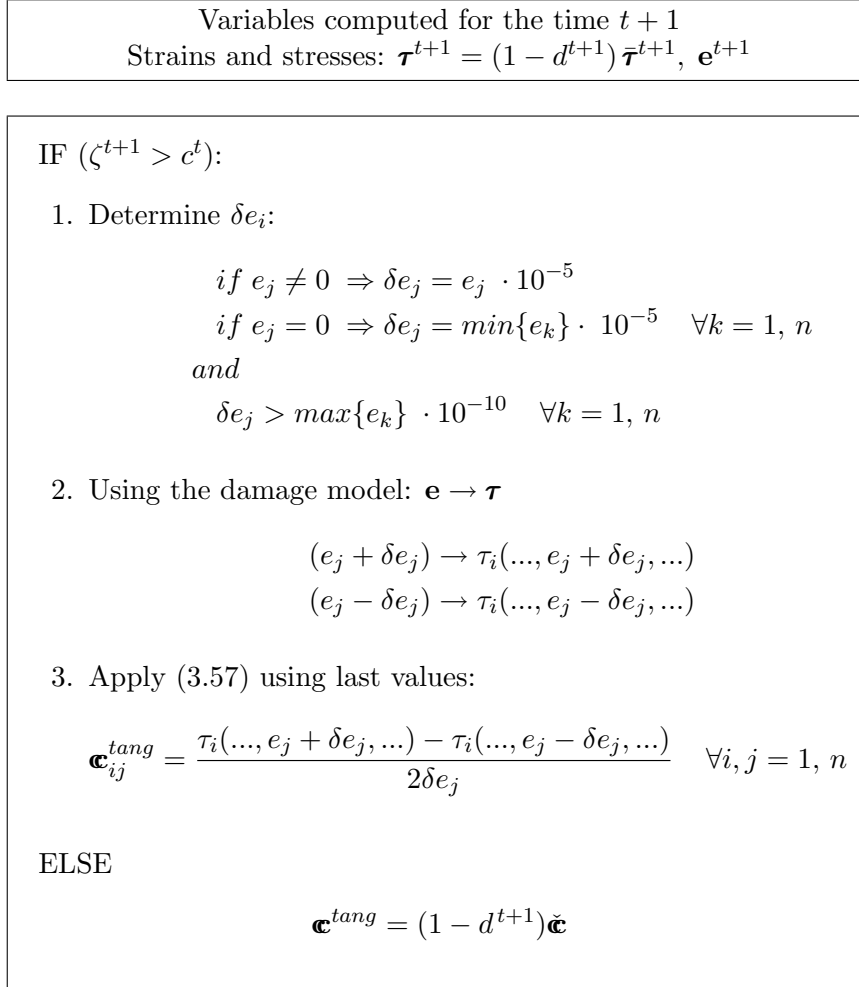
$$\check{\boldsymbol{\mathfrak{C}}}_{ij} = \frac{\dot{\tau}_i}{\dot{\epsilon}_j} \equiv \frac{\delta\tau_i}{\delta e_j} \quad (3.56)$$

Each variation of the coefficient  $\delta e_j$  has as a result a variation of all the coefficients in  $\delta\boldsymbol{\tau}$ . Every  $\delta\tau_i$  is going to be a coefficient of the column  $i$  of the tangent matrix  $\boldsymbol{\mathfrak{C}}^{tang}$ . It can be written as:

$$\begin{aligned} \boldsymbol{\mathfrak{C}}_{i1}^{tang} &= \frac{\tau_i(e_1 + \delta e_1, e_2, e_3, \dots, e_n) - \tau_i(e_1 - \delta e_1, e_2, e_3, \dots, e_n)}{2\delta e_1} \\ \boldsymbol{\mathfrak{C}}_{i2}^{tang} &= \frac{\tau_i(e_1, e_2 + \delta e_2, e_3, \dots, e_n) - \tau_i(e_1, e_2 - \delta e_2, e_3, \dots, e_n)}{2\delta e_2} \\ &\cdot \\ &\cdot \\ &\cdot \\ \boldsymbol{\mathfrak{C}}_{in}^{tang} &= \frac{\tau_i(e_1, e_2, e_3, \dots, e_n + \delta e_n) - \tau_i(e_1, e_2, e_3, \dots, e_n - \delta e_n)}{2\delta e_n} \end{aligned} \quad (3.57)$$

When the damage surface is reached or exceeded the tangent constitutive tensor will be computed using the Perturbation Method described in (3.57). The flowchart for the numerical implementation can be seen in Box 3.6. There are other schemes for the perturbation that use only an increment  $\delta e_i$  and the previous computed stresses of the step. Therefore are cheaper than the presented one, see [78], but they offer a worse approximation in some cases.





Box 3.6: Flowchart for the numerical implementation of the perturbed constitutive tensor for a damage model

Using this numerical procedure an accurate approximation of the tangent constitutive tensor for any constitutive law and any yield surface is provided. Furthermore this ensures a satisfactory convergence for the main numerical process.

It must be taken in account that the approximation of a derivation numerically fails when the derived function presents a kink. Therefore the perturbation will fail when the evolution of the constitutive law encounters with abrupt changes in the slope. In these cases the convergence will be affected negatively.

### 3.7.3 Updating and transferring constitutive internal variables

The damage model takes into account the energy of the element to define the evolution of the damage threshold. The internal variables and the damage influence on stresses are

consistent if the element remains the same during the analysis.

In the PFEM this is not possible, as the particles are the core of the domain. The smoothing produced in the variable transferal (see Section 3.5) dangers the constitutive behavior. Therefore in order not to experience an artificial smoothing of the internal variables, a special transfer scheme must be followed. This is structured in two steps and presented in Box 3.7.

**1. A selective transfer of internal variables from *Elements to Particles*:**

The transfer *Element to Particle* considers only the damage volume associated to the particle.

**2. A selective transfer of internal variables from *Particles to Elements*:**

No information from the damaged particles is transferred to an *Element* which contains a non-damaged *Particle*.

(We consider an *Element* to be the integration points of the finite element, and a *Particle* the nodes of the finite element mesh)

Box 3.7: Rules for updating the constitutive internal variables in the PFEM.

Using this transfer scheme for the variable update the smoothing of the constitutive behavior is minimized. An extra consideration has to be added to the scheme: the largest damage value has to be conserved in order to satisfy the *Kuhn-Tucker* conditions for the damage evolution (see (2.115)).

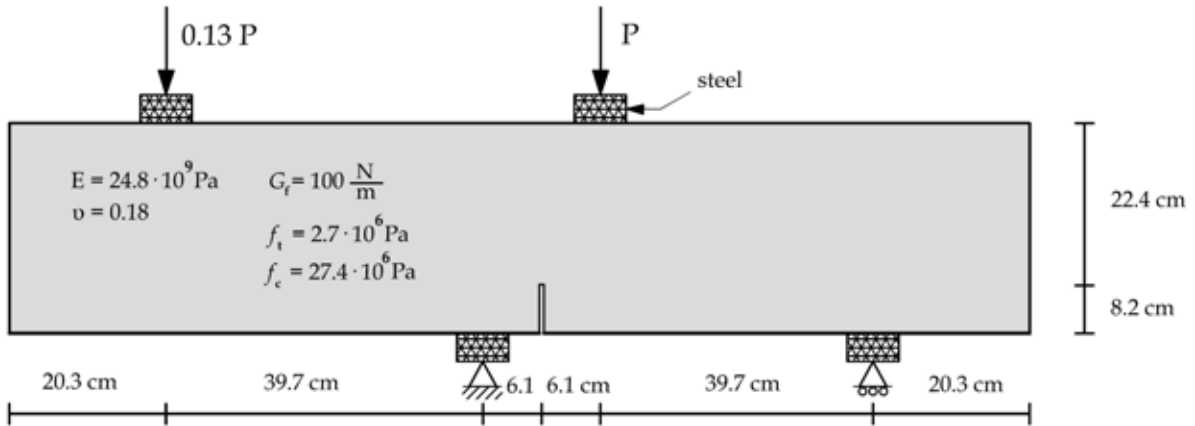
With above considerations the constitutive evolution of the variables is consistent with the method. This does not create artificial damage propagation and gives certain consistency to the constitutive law.

### 3.7.4 Damage, comparison between the FEM and the PFEM

In order to test the behavior of the constitutive models in the PFEM a practical example is analyzed. The objective is to compare the obtained response for the FEM and the PFEM and determine the error for the material characterization due to the transfer of the constitutive variables. Examples in 2D and 3D are studied in order to perform the comparison and check the influence of the dimension in the numerical computations.

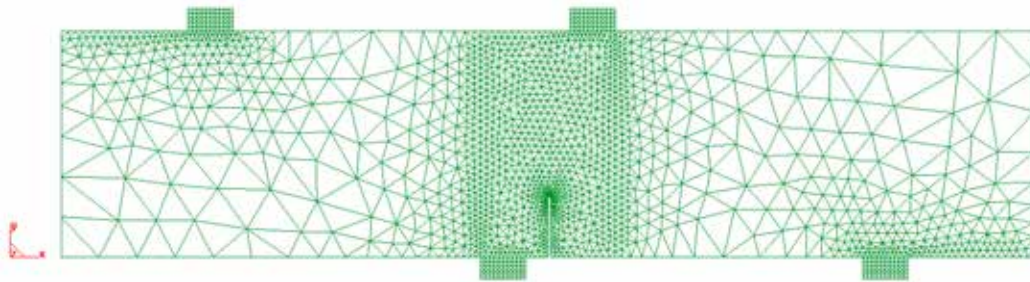
#### Damage example in 2D: notched shear beam

The selected case is a classical problem of a notched shear beam (see [111], [8], [82]). The supports and the loading conditions are non-symmetric with respect to the notch. The geometry and the problem parameters are shown in Figure 3.22.



**Figure 3.14:** Fracture test model of a notched unreinforced beam.

The finite element discretization of the beam is presented in Figure 3.15.

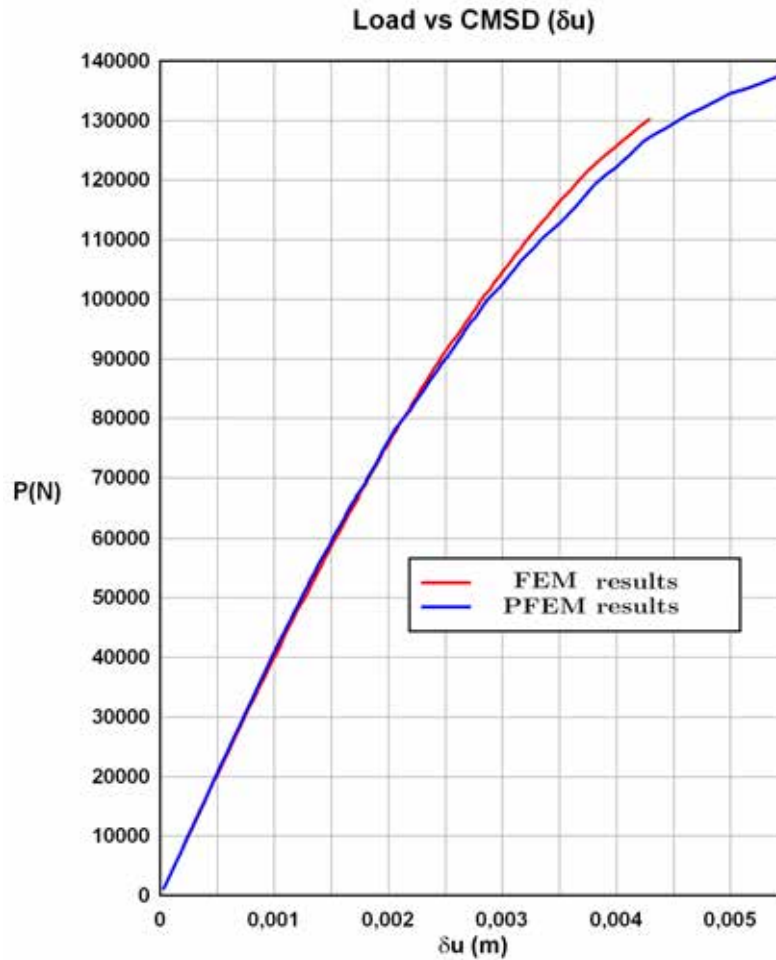


**Figure 3.15:** Finite element idealization for the notched unreinforced beam.

The load is applied on steel supports in order to distribute it on the beam. This also avoid localized stresses than can induce damage in these zones. For the damage threshold two different flow functions are used: *Simo-Ju* and *Mohr-Coulomb*. The *Simo-Ju* model is analyzed with *exponential softening* and with *linear softening* (see equation (2.122) and (2.123)). The *Mohr-Coulomb* model is analyzed using *linear softening*. The description of these flow functions is presented in Section 2.4.

For the response of the beam, the load is compared with the Crack Mouth Sliding Displacement (CMSD), which coincides with the relative vertical displacement on the lower points of the notch  $\partial u$ . Figure 3.16 shows the non-linear response of the beam in terms of load-CMSD for the *Simo-Ju* model with *exponential softening*.

The experiments give a range of results between  $P = 125 - 140 \text{ kN}$  for the maximum load. The curve has to snap back after the load peak. This behavior can only be captured by a numerical analysis with control of the displacements. In this example the purpose is to compare the FEM and the PFEM hence the comparison of the first state of the physical

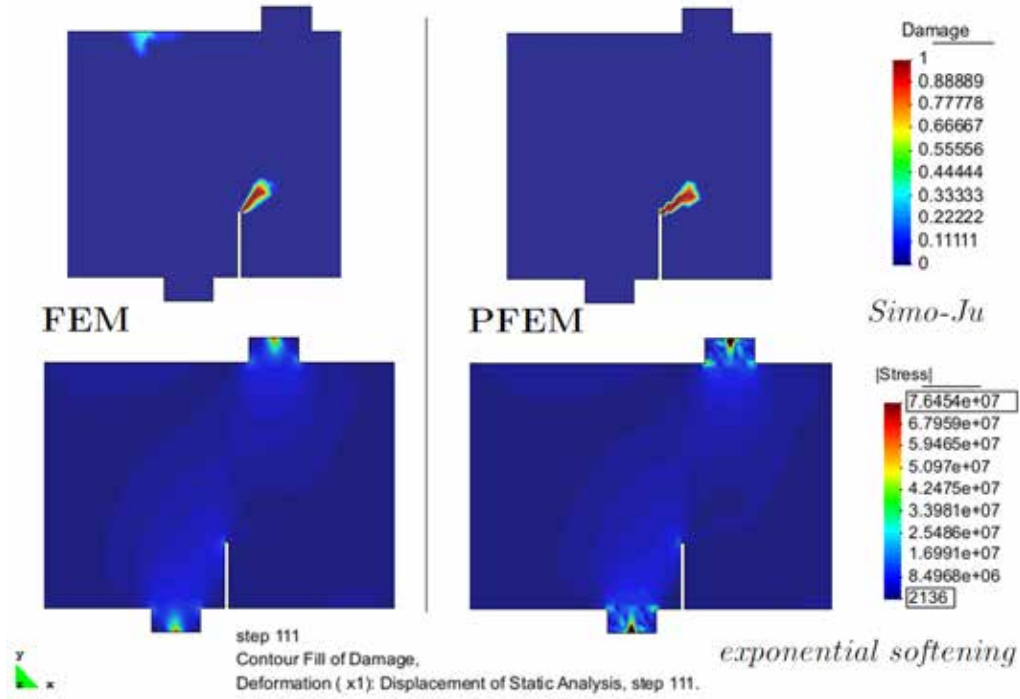


**Figure 3.16:** Load vs CMSD response for fracture problem using *Simo-Ju* model with *exponential softening*.

response gives a more objective view. The response curve for the PFEM is always below the FEM curve. This is a consequence of the unavoidable smoothing that the stresses and internal variables experience due to mesh transfer. At the same time this smoothing produces a relaxation and allows a larger load and a later response peak. This happens in this model but it will differ for other models.

The crack patterns are presented in Figure 3.17 with the corresponding stress field for the FEM and the PFEM.

Figures 3.16 and 3.17 show small differences in the results computed with the FEM and with the PFEM. The crack pattern has similar fracture direction and the response peak is within the experimental range in both cases. If the softening law is changed the relative response is very similar. Figure 3.18 and Figure 3.20 show the non-linear response of the beam in terms of load-CMSD and the crack patterns for the *Simo-Ju* model with *linear softening*.



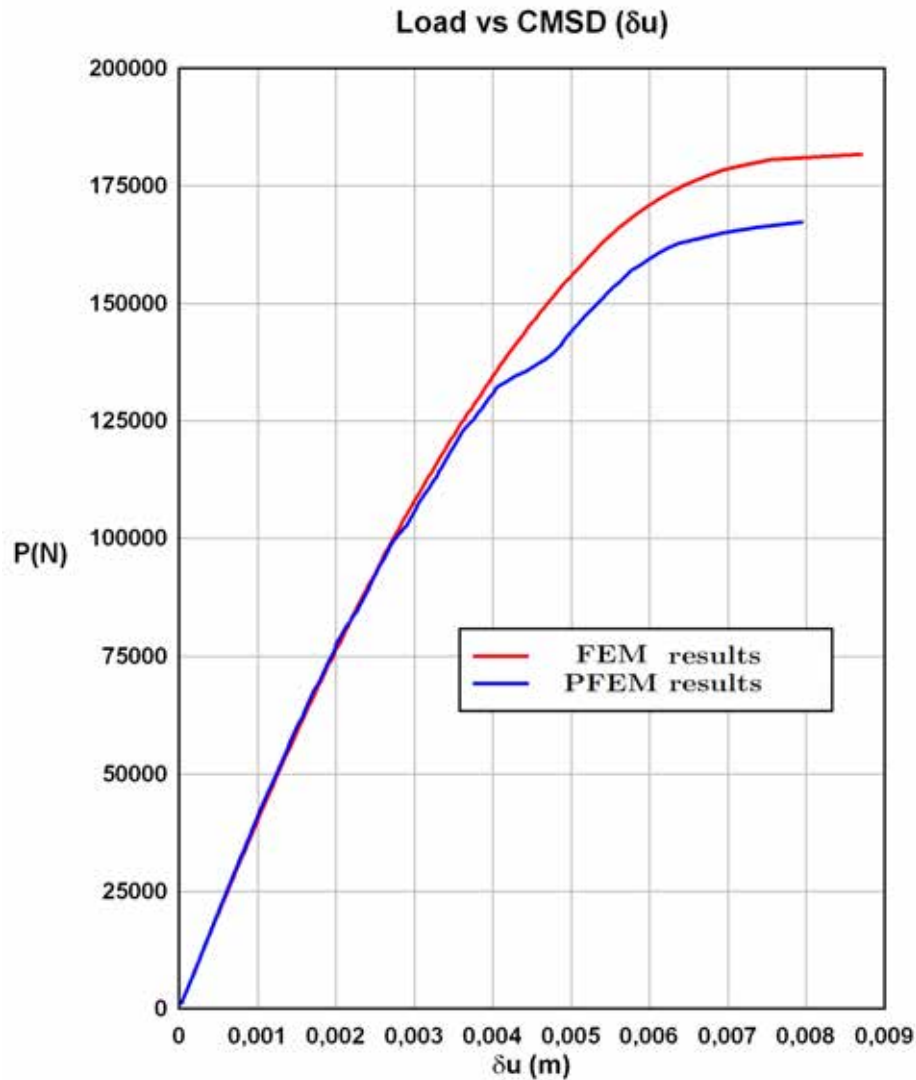
**Figure 3.17:** Crack pattern and stresses at-peak using *Simo-Ju* model with *exponential softening*. Analysis with the FEM on the right and with the PFEM on the left.

Using *linear softening* the peak load is larger and the crack pattern has a small increase in the slope compared with *exponential softening*. The same behavior is observed for the FEM and the PFEM. The bend that can be observed for the PFEM curve represents the premature damage on the beam in the steel supports. This is produced by the concentration of stresses in this area due the variable transfer.

*Mohr-Coulomb* model with *linear softening* has also been tested in order to get the response with a a more suitable flow function for a geomaterial. For this case the used material changes a little bit. The used parameters are defined in Figure 3.22 considering the friction angle of the material as  $\theta = 0.32 \text{ rad}$  and  $f_t = \sigma_t^{max} = 2.7 \cdot 10^6$ . This friction angle is relatively small and it does not gives the same tension-compression relation. It can be seen using next equivalence:

$$R_{Mohr} = \left| \frac{f_c}{f_t} \right| = \tan^2 \left( \frac{\pi}{4} + \frac{\theta}{2} \right) \quad (3.58)$$

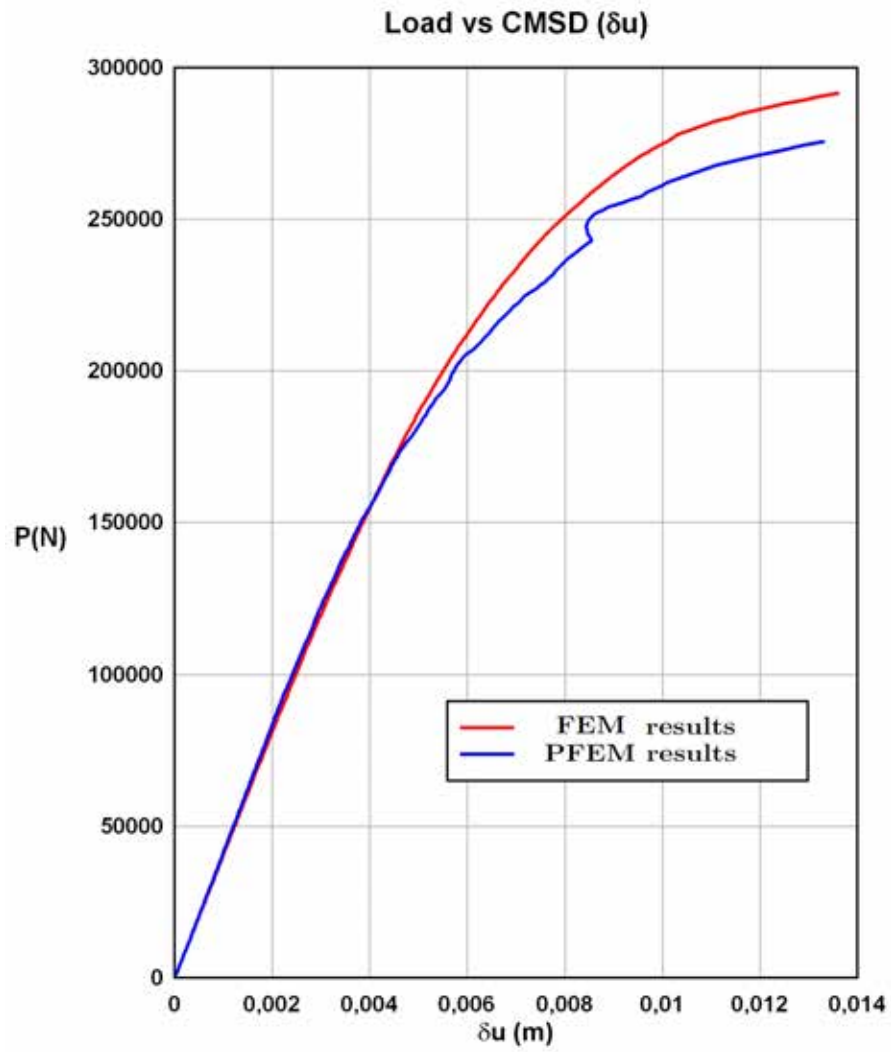
where  $R_{Mohr}$  is tension-compression ratio defined for the *Mohr-Coulomb* criterion and  $\theta$  is the friction angle. Last equation expresses the limitation of the flow function to characterize all types of materials. Each flow function is designed to model a specific material type. In order to have a material equivalence and obtain a constitutive behavior similar to *Simo-Ju* model, a modification of the *Mohr-Coulomb* criterion is needed. It can be found in [98].



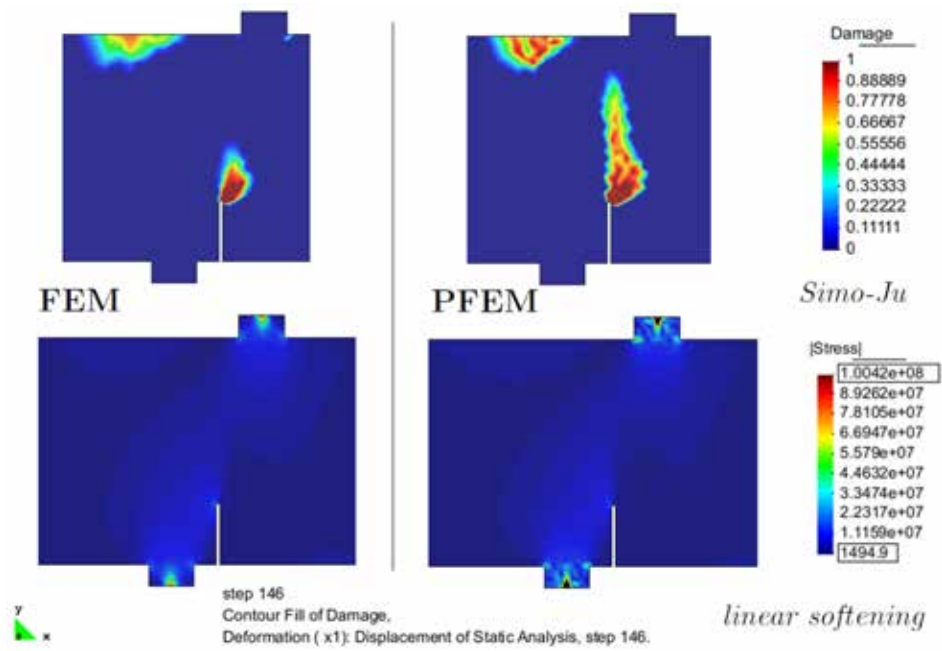
**Figure 3.18:** Load vs CMSD response for fracture problem using *Simo-Ju* model with *linear softening*.

Figures 3.19 and 3.21 show the non-linear response of the beam in terms of load-CMSD and the crack patterns for the *Mohr-Coulomb* model with *linear softening*. The strange bend that can be observed for the PFEM curve represents the premature damage on the beam in the steel supports. This is caused by the same phenomenon again, the concentration of stresses due the variable transfer in the base of the steel supports.

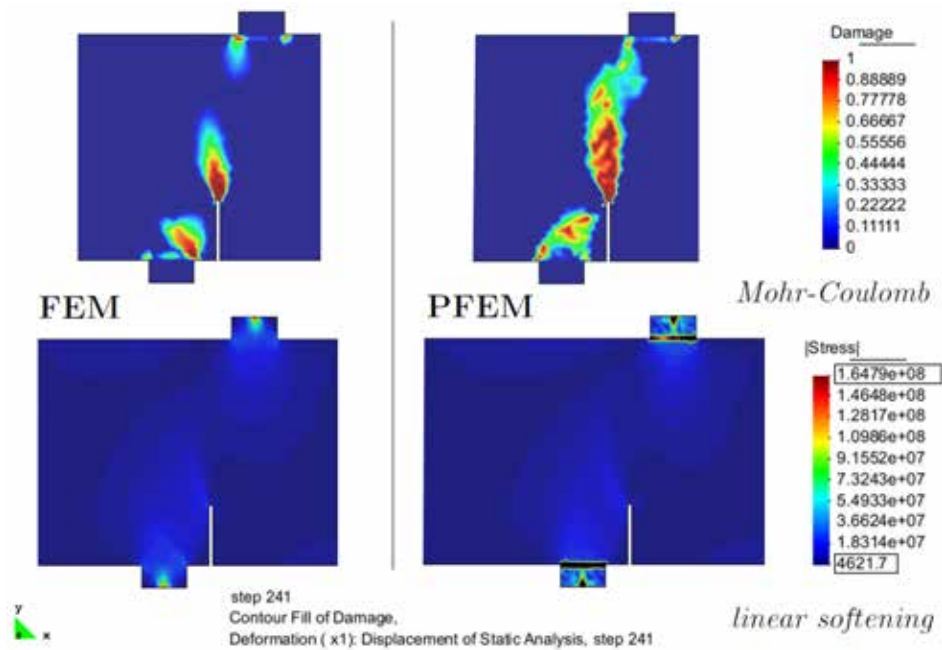
The results obtained using the standard FEM or the PFEM are very similar. In both cases getting a good result depends on the complexity of the constitutive law and also on a proper discretization of the domain.



**Figure 3.19:** Load vs CMSD response for fracture problem using *Mohr-Coulomb* model with *linear softening*.



**Figure 3.20:** Crack pattern and stresses at-peak using *Simo-Ju* model with *linear softening*. Analysis with FEM on the right and with PFEM on the left.

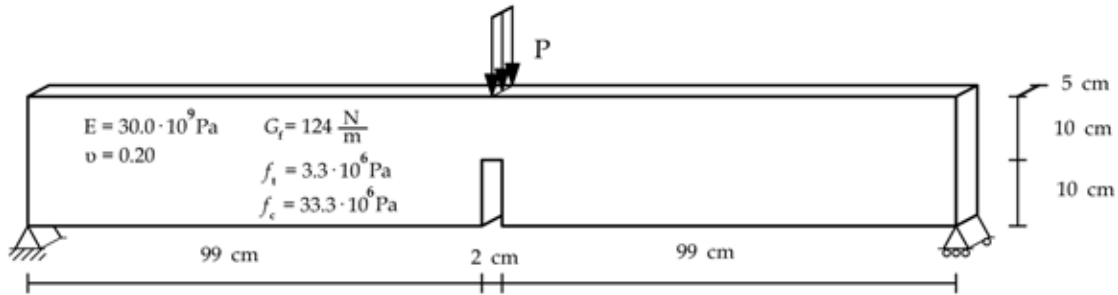


**Figure 3.21:** Crack pattern and stresses at-peak using *Mohr-Coulomb* model with *linear softening*. Analysis with FEM on the right and with PFEM on the left.



### Damage example in 3D: middle notched shear beam

The case analyzed is also a classical problem of a notched beam (see [111] and [101]). Now the notch is in the middle of the beam. The supports and the loading conditions are symmetric with respect to the notch. The geometry and the problem parameters are shown in Figure 3.22.



**Figure 3.22:** Fracture test model of a 3D notched unreinforced beam.

The finite element discretization of the beam is presented in Figure 3.23. The initial mesh has 74076 elements and 18062 nodes.

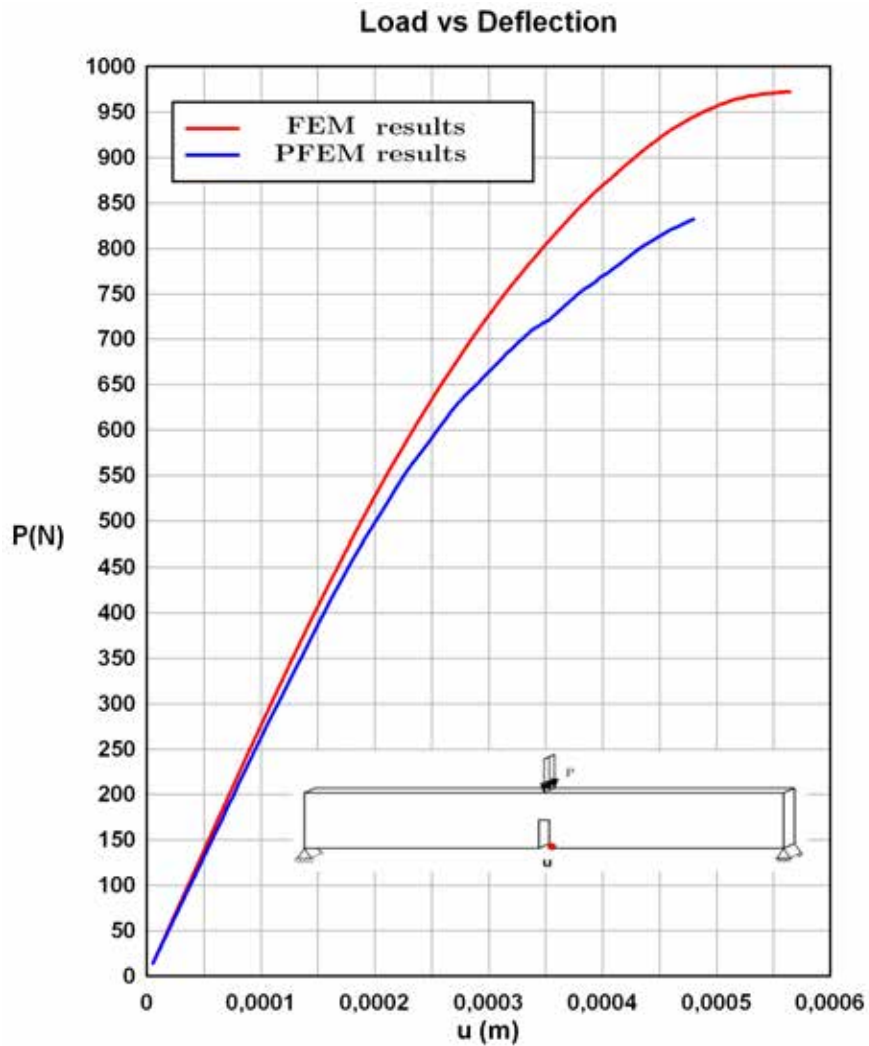


**Figure 3.23:** Finite element idealization for a 3D notched unreinforced beam.

The load is applied on the middle top of the beam distributed along the thickness. One of the supports has only a vertical constraint on movements and the other is constrained vertically and horizontally. For the damage threshold a *Simo-Ju* function is used. The *Simo-Ju* model is analyzed with *exponential softening* (see equation (2.122)).

The experiments and previous finite element analysis in 2D give a range of results between  $P = 750 - 850 \text{ N}$  for the maximum load supported by the beam. After the load peak the propagation of the crack is fast and the load has to be reduced in order to get a load-displacement curve. This behavior can only be captured by a numerical analysis with displacement control. The first state of the physical response, from  $P = 0$  until the load peak gives a more objective view for the comparison of the standard FEM and the PFEM.

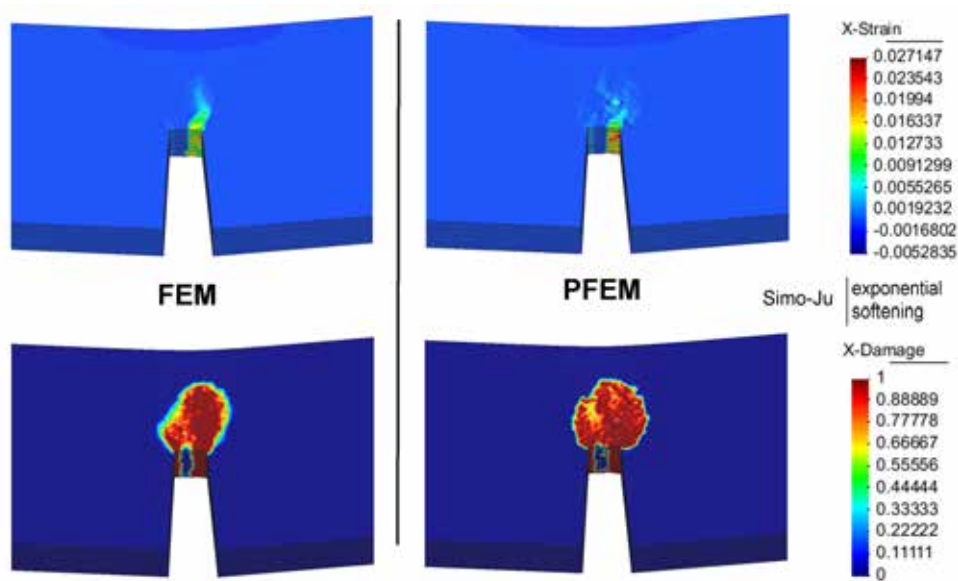
In this example the load is compared with the deflection in the middle of the beam. The displacement of the middle right lower edge of the notch is taken as a reference. Figure 3.24 depicts the non-linear response of the beam in terms of load-deflection.



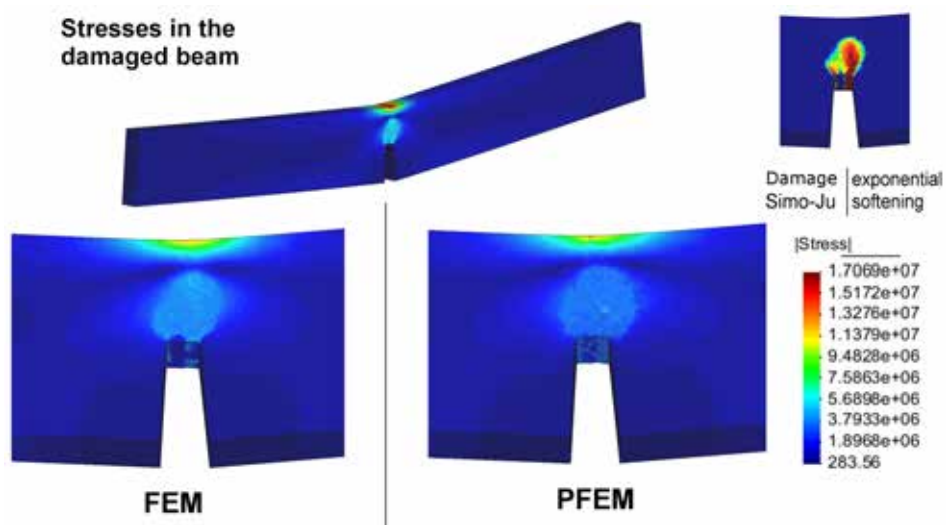
**Figure 3.24:** Load vs deflection response for fracture problem using *Simo-Ju* model with *exponential softening* in a 3D model.

The response curve for the PFEM is always below the FEM response curve. The same that occurred in the 2D case. This is a consequence of the unavoidable smoothing that the stresses and internal variables experience due to the mesh transfer.

Figures 3.26 and 3.25 compare the results obtained with FEM and PFEM. The crack pattern has the same fracture direction and the response peak is inside of the experimental range in both cases. Computed stresses and strains have similar distributions in both cases also.



**Figure 3.25:** Crack pattern and strains at-peak using *Simo-Ju* model with *exponential softening*. Analysis of a 3D model with FEM on the right and with PFEM on the left. Comparison at the same time step.



**Figure 3.26:** Stresses in the notch area, at-peak using *Simo-Ju* model with *exponential softening*. Analysis of a 3D model with FEM on the right and with PFEM on the left. Comparison at the same time step.

*Seek simplicity, and distrust it.*

Alfred North Whitehead (1861-1947)

# 4

## Meshing process in the PFEM

### 4.1 Introduction

In the PFEM the domain is defined by a cloud of particles instead of a group of sub-domains or elements as is common in the FEM. From these particles a union of sub-domains is created and the total domain is defined. This vision of the continuum corresponds to one of the standard methodologies developed in computational geometry. The methodology used to perform the meshing process is the *Delaunay Tessellation* (DT) [37]. This method starts from the introduction of a point distribution in the domain and then performs iteratively a partition in order to create elements with optimal shape.

In the PFEM, the meshing of the domain is essential for the computation with the FEM. The analysis is focused in the movement of the particles, but the filling of the continua is required for the boundary recognition and contact detection. By means of the remeshing process the boundary of the different body entities is identified. The particles carry forward enough information to allow the analysis in the new mesh at each time step.

When a result is computed an update of the position is performed. Particles have a new relative position in space. The *Delaunay Tessellation* is used to create a mesh from a cloud of particles in space [21]. This mesh is created in the new analysis domain at each time step when large deformations occur, or when is required due to other circumstances. See Figure 3.1.

### 4.2 Remeshing techniques

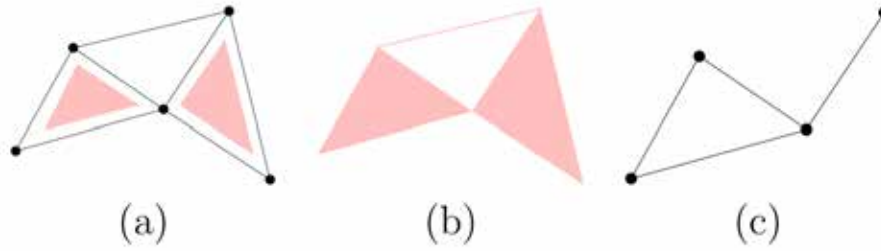
The *Delaunay tessellation* of a vertex set, introduced by Delaunay in 1934 [37], is a useful geometric procedure to divide a continuum domain in sub-domains (elements). The language of combinatorial topology will be used to describe the DT. Understanding of some basic notions is necessary, such as convex hull, simplex or simplicial complex.

#### 4.2.1 Some combinatorial topology

This is an introduction section with simplified explanations of some basic notions of combinatorial topology. It aims to be a quick reference for the understanding of the techniques explained in the next sections

##### Convex hull

The *convex hull* of a set of points  $V \subset \mathbb{R}^n$  is denoted as  $convV$  and defines the smallest convex set that contains  $V$ .



**Figure 4.1:** (a) A two-dimensional simplicial complex  $K$  consists of 2 triangles (which are shaded), 7 edges, and 5 vertices (b) The underlying space (c) A subcomplex, which is a 1-dimensional simplicial complex, consists of 4 edges, and 4 vertices. (TetGen Manual, [4])

### Simplex

A *simplex*  $\varsigma$  is the convex hull of a related independent set of points  $S$ . The dimension of  $\varsigma$  is  $k$ , and is established by  $k = n_p - 1$ , where  $n_p$  is the number of points that contains  $S$ . Consequently in  $\mathbb{R}^3$  the maximum number of related independent points of a independent set  $S$  is 4. It is possible to have non-empty *simplex*  $\varsigma$  of dimensions 0, 1, 2 and 3. These dimensions define geometrical entities corresponding to *vertices*, *edges*, *triangles* and *tetrahedra*, respectively. For any subset  $T \subset S$ , the *simplex*  $\varrho = \text{conv}T$  is a face of the *simplex*  $\varsigma$  and is denoted as  $\varrho \leq \varsigma$ .  $\varrho$  would be a proper face of  $\varsigma$  if  $T$  is a proper subset of  $S$ .

### simplicial complex

A *simplicial complex*  $K$  is a finite set of simplices that fulfill the next:

1. Any face of a simplex of the *simplicial complex*  $K$  is also included in  $K$ .
2. The intersection of any two simplices of the simplicial complex  $K$  is a face of both simplices.

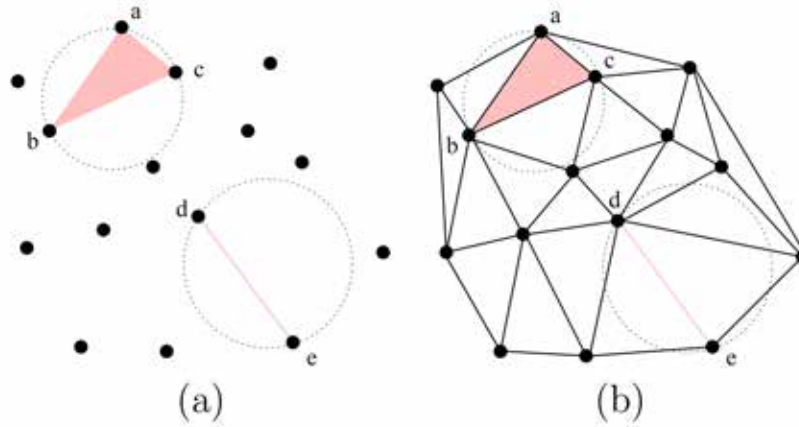
The second condition defines the empty set of points,  $S$  with  $n_p = 0$ . That happens when two simplices are disjoint; the unique possible simplex is (-1)-dimensional and is a face of any simplex. Figure 4.1a illustrates an example of a two-dimensional *simplicial complex*.

### Underlying space

The *underlying space* of a set of simplices  $L$ , denoted as  $|L|$ , is the union of interiors,  $\bigcup_{\varsigma \in L} \text{interior}\{\varsigma\}$  (see an underlying space in Figure 4.1b).  $|L|$  is a topologically closed set, iff  $L$  is a simplicial complex.

### Subcomplex

A *subcomplex* of  $K$  is a subset of simplices of  $K$ , hence it is also a simplicial complex. Figure 4.1c shows an example of a subcomplex.



**Figure 4.2:** The Delaunay criterion and Delaunay triangulation in two dimensions (a) Both the 2-simplex  $abc$  and the 1-simplex  $de$  are Delaunay (b) The corresponding Delaunay triangulation of the point set shown in (a) (TetGen Manual, [4])

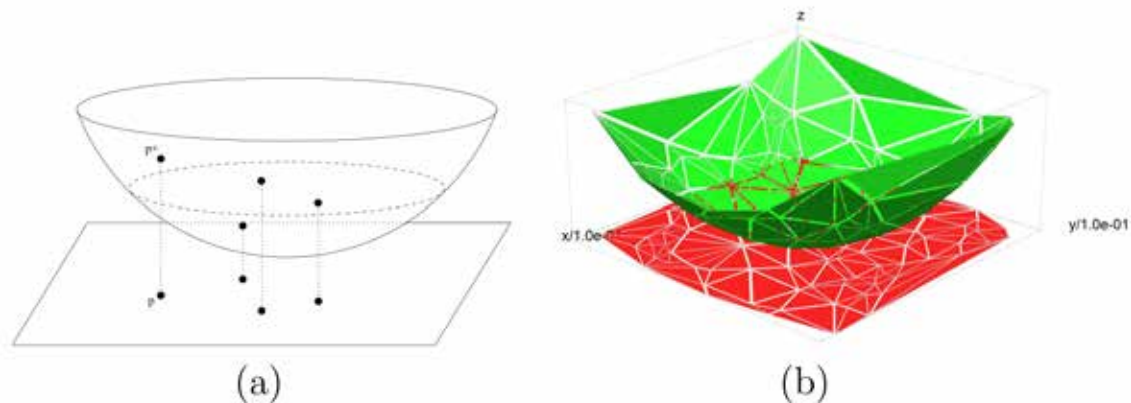
#### 4.2.2 Delaunay tessellation and convex hull

Let  $V$  be a set of  $n$  vertices or points in the  $d$ -dimensional Euclidean space  $\mathbb{E}^d$ . The points of  $V$  are called vertices or nodes.  $\zeta$  is defined as a  $k$ -simplex ( $0 \leq k \leq n$ ) formed by vertices of  $V$ . The *circumsphere* of  $\zeta$ , denoted by  $c_\zeta$ , is a sphere that passes through all vertices of  $\zeta$ . If  $k = d$ ,  $\zeta$  has a unique circumsphere, otherwise, there are infinite circumspheres that can contain the vertices of  $\zeta$ . The *simplex*  $\zeta$  is Delaunay when exists a circumsphere with no vertex of  $V$  lying inside of it. Figure 4.2a shows Delaunay simplices in a set of two-dimensional vertices.

The *Delaunay tessellation*  $\mathcal{D}$  of  $V$  is a simplicial complex consisting of Delaunay simplices, the set of all simplices of  $\mathcal{D}$  covers the convex hull of  $V$ . Therefore a two-dimensional Delaunay tessellation, called *Delaunay triangulation* is the unique triangulation of  $V$  where there are no vertices of  $V$  inside the circumsphere of any triangle. This is illustrated in Figure 4.2b. In three dimensions, it is also called *Delaunay tetrahedralization*.

A Delaunay tessellation in the Euclidean space  $\mathbb{E}^d$  corresponds to a convex hull in  $\mathbb{E}^{d+1}$ . For a point  $p = (p_1, p_2, \dots, p_d) \in \mathbb{E}^d$ , it is possible to define a "lift point"  $p^+ = (p_1, p_2, \dots, p_d, p_{d+1}) \in \mathbb{E}^{d+1}$ , where  $p_{d+1} = \sum_{i=1}^d p_i^2$ . This property can be translated to a set of vertices  $V \subset \mathbb{E}^d$ .  $V^+ = \{p^+ | p \in V\}$  would be defined as the set of "lifted points". The origin of these vertices will be in the set  $V \subset \mathbb{E}^d$  and will be lifted on a paraboloid in  $\mathbb{E}^{d+1}$  (Figure 4.3a). Then, the convex hull  $con(V^+)$  is a  $(d + 1)$ -dimensional convex polytope. Therefore a Delaunay tessellation of  $V$  can be produced by projecting  $con(V^+)$  into  $d$  dimensions. Figure 4.3 illustrates the relationship when  $d = 2$ .

In three or higher dimensional combinatorial structures convex hull algorithms related to convex polytopes are used to generate Delaunay tessellations [73], [27] and [10]. Recent research shows that they are effective in removing slivers for quality mesh generation [23], [24].

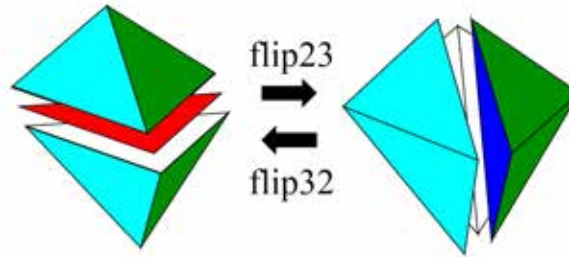


**Figure 4.3:** The relation between Delaunay triangulation in  $\mathbb{E}^d$  and convex hull in  $\mathbb{E}^{d+1}$  (here  $d = 2$ ) (a) Some 2D points and their corresponding 3D lift points (b) The Delaunay triangulation of a set of 2D points and the lower convex hull of its 3D lifted points. (TetGen Manual, [4])

The Delaunay tessellation has many optimal properties. Among all triangulations of a set of points in  $\mathbb{R}^2$ , it maximizes the minimum angle of the elements and also minimizes the maximum circumradii. It is combined with optimal time complexity algorithms (divide-and-conquer and plane-sweep) for a fast mesh generation. A discussion of some optimal properties of the Delaunay triangulation in three or higher dimensions can be found in [105].

There is an alternative for the construction of Delaunay tessellations. They can be constructed by "flips". A *flip* is an operation to transform a set of non-locally Delaunay simplices into another set of simplices which are locally Delaunay. Figure 4.4 illustrates two types of flips in three dimensions. The basic idea of flip-based algorithms is relatively simple: start from an arbitrary triangulation; create flip simplices that are not locally Delaunay until all simplices are locally Delaunay. The result is a Delaunay triangulation. Lawson [72] first used a flip algorithm to construct two-dimensional Delaunay triangulations. However, in a dimension higher than two, such simple scheme may not terminate, see the discussion in [59]. Nevertheless, it has been proved that Delaunay tetrahedralizations can be constructed by incremental insertion of vertices and flips, see [59].

In higher dimensions tessellations get more complex. An example is the duality problem in the Delaunay criterion. It happens when different element distributions satisfy all criteria. When this duality occurs in a Delaunay tetrahedralization, flat tetrahedra can appear. These elements are called *slivers*, and have some problems in the evaluation of the shape functions of finite elements. A huge values of the gradient of the shape functions are observed due to the flat shape of the tetrahedron. In order to overcome these drawbacks one of the solutions is to generalize the Delaunay tessellation, allowing to include other polyhedrons in the tetrahedralization. This generalization is called *Extended Delaunay Tessellation* (EDT), [55], [56]. The drawback of the EDT is the need of suitable shape functions for the new polyhedra. An extended explanation of the EDT can be found in [36] and [77].



**Figure 4.4:** Two flip operations in three dimensions. A 2-to-3 flip replaces a non-locally Delaunay face (the red face on the left figure) into three locally Delaunay faces (the blue and white faces on the right figure); while a 3-to-2 flip does the inverse. (TetGen Manual, [4])

### 4.2.3 Voronoï diagrams and Delaunay tessellations

There are some important properties used in computational geometry for the design of efficient algorithms. The motivation of the first property comes from the problem of the nearest-point search. It exists a special diagram for a set of nodes that solves this problem; it is called *Voronoï* diagram, see [124].

The *Voronoï* diagram is the dual of the Delaunay tessellation, both defined in the same vertex set. For any vertex  $p \in V$ , the Voronoï cell of  $p$  is the set of spatial points that are closer to  $p$  than any other vertex of  $V$ . In other words, it is the set  $\{\forall x \in \mathbb{E}^n \mid |x - p| \leq |x - b|, b \in V\}$ , where  $|\cdot|$  stands for the Euclidean distance. The Voronoï diagram of  $V$  subdivides the space  $\mathbb{E}^n$  into Voronoï cells (some of them may be unbounded). Delaunay triangulation and Voronoï diagram are geometrically dual. That means, for example in two dimensions, that Voronoï polygons correspond to Delaunay vertices, Voronoï edges correspond to Delaunay edges, and Voronoï vertices correspond to Delaunay triangles (illustrated in Figure 4.5a).

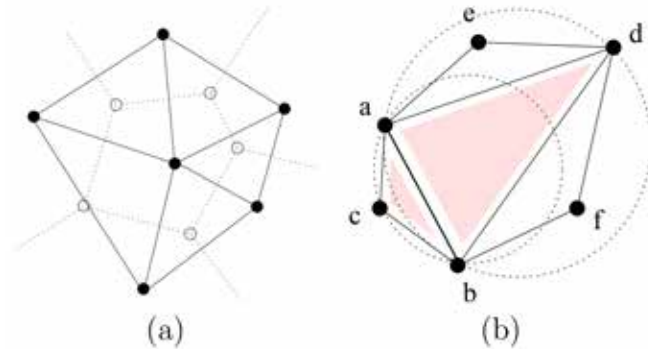
Another useful property is the *localization* property of a Delaunay tessellation. Let  $\mathcal{T}$  be an arbitrary tessellation of  $V$ , and  $\varsigma$  be a simplex of  $\mathcal{T}$ . If  $\mathcal{K}$  is a subcomplex of  $\mathcal{T}$  formed by simplices contained in  $\varsigma$ .  $\varsigma$  is defined as *locally Delaunay* if exists a circumsphere  $c_i \in \varsigma$  that encloses no vertices from the vertex set of the subcomplex  $\mathcal{K}$  in its interior. Figure 4.5ab illustrates the property in two dimensions. Evidently, if every simplex of  $\mathcal{T}$  is locally Delaunay, then  $\mathcal{T}$  is a Delaunay tessellation.

## 4.3 Boundary recognition

In solid mechanics, the geometry of the domain is determined at the onset of the analysis. By means of meshing the spatial domain, a group of nodes and elements will discretize that geometry. Once a mesh is generated the essential boundary conditions are imposed and labeled in nodes. The nodes define the initial cloud of particles that will contain the characteristics of the continuum under study.

The discrete partition with particles defines the continuum for the rest of the time-





**Figure 4.5:** Properties of Delaunay triangulations. (a) The relation between Delaunay triangulation and Voronoi diagram. (b) Locally Delaunay property. Edge  $ab$  is locally Delaunay. Here only  $c$  and  $d$  affect the property because there are triangles  $abc$  and  $abd$  sharing edge  $ab$ .  $e$  and  $f$  are excluded from the definition of the property. (TetGen Manual, [4])

dependent problem. The condition imposed over each particle is known at each time step. The solution of the PDE on the system will transport the particles, which can experiment large displacements and deformations. When this happens the cloud of points in the space is redistributed and, hence, defines a different shape of the domain geometry.

The problem now is how to define the new shape on the space using only a cloud of points. With the meshing method above explained (the *Delaunay tessellation*) the convex hull of a given cloud of points can define a finite element mesh. The convex hull can be formed for multiple independent domains, with cavities or holes and with a very specific geometry. To recover the original shapes of each independent body a new definition of shape in the space, starting from a cloud of points, has to be formulated.

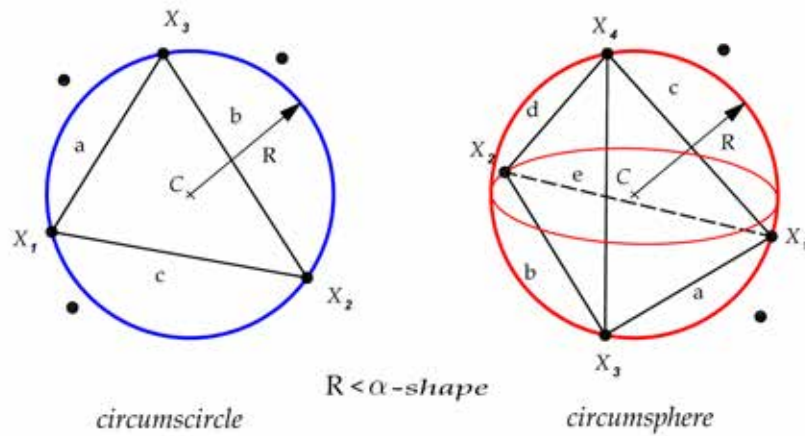
A mathematical rigorous definition of shape was introduced by *Edelsbrunner et al.* [41]. They proposed a natural generalization of convex hulls, called  $\alpha$ -hulls, and their combinatorial variant,  $\alpha$ -shapes. In two dimensions  $\alpha$ -shapes are related to the Delaunay circumcircles obtained from a Delaunay triangulation. In three dimensions they are related to Delaunay circumspheres. The Delaunay tessellation was explained in Section 4.2.2.

### 4.3.1 Alpha shape concept

Alpha shapes provide a mathematical basis for defining the shape of an unstructured set of points. Conceptually,  $\alpha$ -shapes are a generalization of the convex hull of a point set. They constitute a whole family of shapes, ranging from the original point set up to the convex hull. Each shape is parameterized by a real number  $\alpha$  that takes a value between 0 and  $\infty$ . The  $\alpha$  parameter represents the radius of a circle (2D) or a sphere (3D) in the Euclidean space  $\mathbb{E}^d$ , (Figure 4.6).

The  $\alpha$ -shape of a point set  $V \in \mathbb{E}^d$  is a polytope that is neither necessarily convex nor necessarily connected.

For  $\alpha = \infty$ , the  $\alpha$ -shape is identical to the convex hull  $S$ . However, as  $\alpha$  decreases, the



**Figure 4.6:** The alpha-shape parameter is compared with the radius of the circumcircle and the circumsphere formed by a set of points in two-dimensions or three-dimensions respectively, see Appendix A

shape shrinks by gradually developing cavities. These cavities may join to form tunnels or even closed holes (voids). For  $\alpha = 0$ , the  $\alpha$ -shape is only the points of  $V$ . When a sphere, of several of such spheres with radius  $\alpha$  can occupy its space without enclosing any point of  $V$ , no polytope can be formed connecting the vertices of  $V$ .

Intuitively,  $\alpha$  determines the size of the Delaunay *Simplicial Complexes* defined in the point set  $V \in \mathbb{E}^d$ . When  $\alpha$  becomes small enough, the Delaunay simplices of the simplicial complex become disjoint and the polytope disappears.

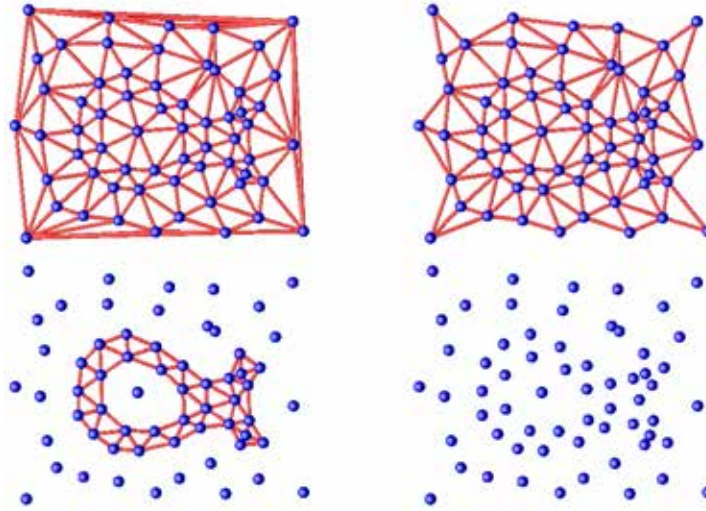
Figure 4.7 illustrates the complete family of alpha shapes for a set of points in two dimensions.

More rigorous definitions for alpha shapes can be found in [41].

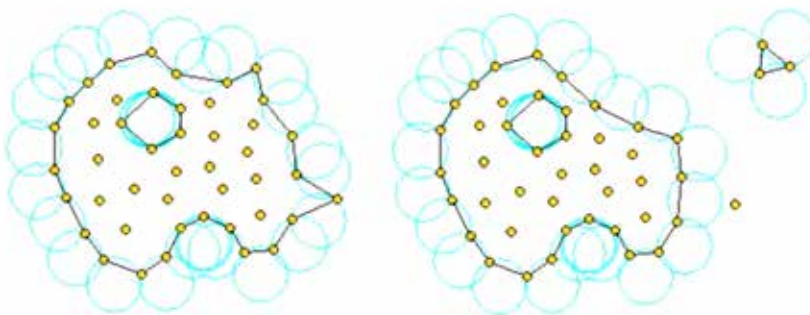
The alpha shape concept can be applied directly as a criteria in the Delaunay tessellation for obtaining the desired shape for the cloud of particles defining the convex hull. It means that first the set of considered particles are selected by the alpha concept and these determine the boundaries of the convex hull with polytopes. Then the tessellation is performed inside these polytopes. Figure 4.8 shows the boundary recognition of a set of particles previous to the tessellation.

The  $\alpha$ -shape method is therefore a way of finding the boundaries of a uniformly distributed set of particles. When the particles have non regular distributions, a recognition of the boundary with only one global uniform  $\alpha$  parameter will fail. The description of the shape of the domain for a non-anisotropic particle distribution is a drawback for  $\alpha$ -shapes. This penalizes the possibility of mesh refinement for getting more accurate results.

This drawback is solved using non-uniform  $\alpha$ -shapes for all set of particles. The  $\alpha$  parameter can be adapted to the particle distribution and modified consistently with the refinement of the mesh. This strategy will be adopted for the meshing process in the PFEM.



**Figure 4.7:** The complete family of alpha shapes of a set of points in two-dimensions.(Del Pin, [36])



**Figure 4.8:** Boundary recognition of a set of particles using the  $\alpha$ -shape concept ([84])

## 4.4 Remeshing strategy for the PFEM

The remeshing of the domain is essential in the PFEM. By means of the remeshing the boundary of the different body entities are identified. The particles carry forward enough information to allow the analysis in the new mesh at each time step.

### 4.4.1 Assigning non-uniform $\alpha$ -shapes

The PFEM uses non-uniform  $\alpha$ -shapes for characterizing the domain distributions. It means that the  $\alpha$  parameter is adapted to the particle distribution. Each particle has a characteristic parameter  $\alpha$ , which is defined by the initial size of the mesh and computed using the average distance between neighbor particles as

$$\alpha_j = \sum_{i=1}^{n_b} \frac{|\mathbf{x}_i - \mathbf{x}_j|}{n_b} \quad (4.1)$$

where  $n_b$  is the number of neighbor particles  $i$ , which are defined as the particles contained in the same simplices as particle  $j$ .  $\mathbf{x}_k$  is the spatial position of a particle.

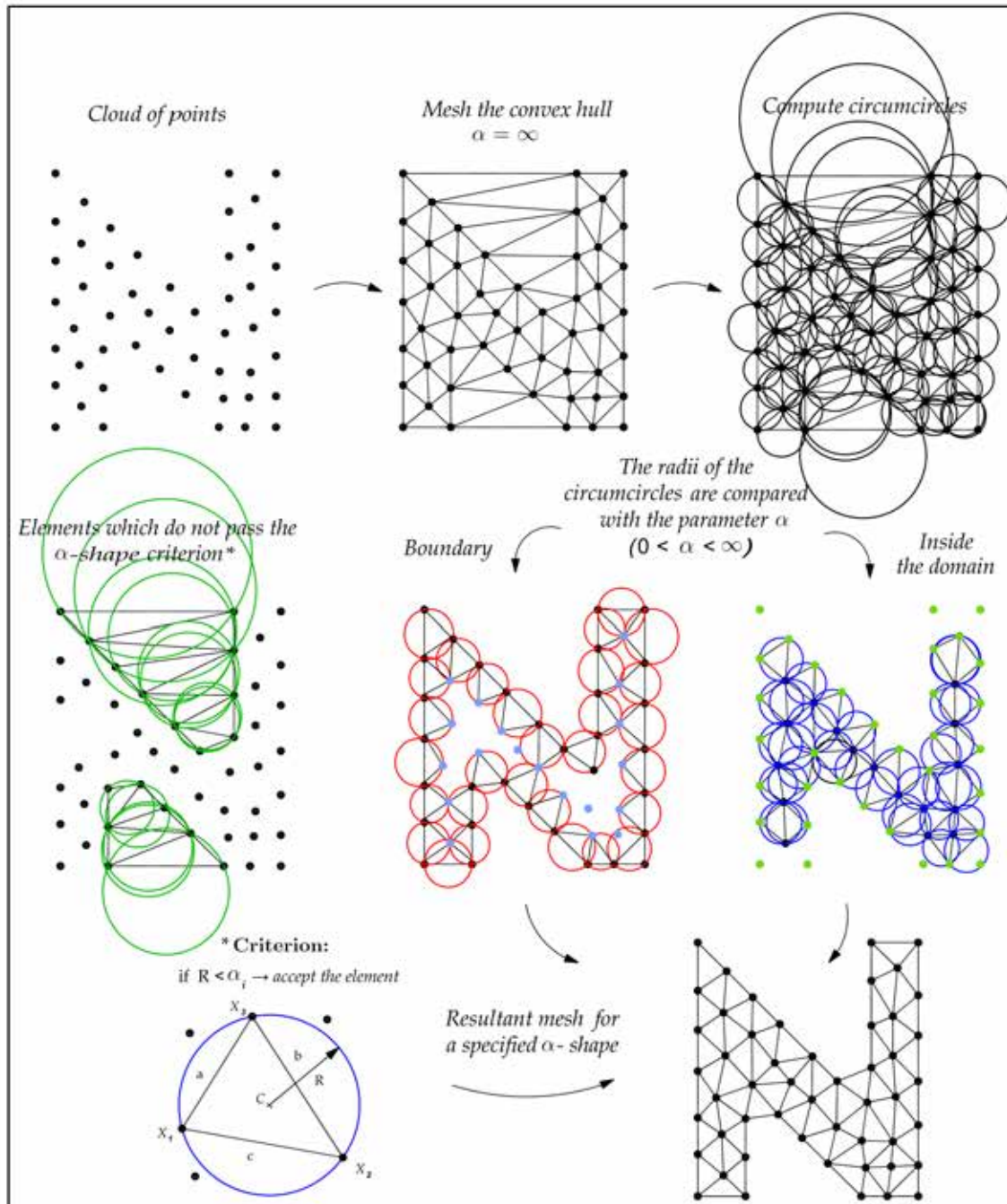
In case of mesh refinement, a  $\alpha_j$  parameter is assigned to each inserted particle  $i$ . This  $\alpha_j$  corresponds to the particle distribution that defines the desired local mesh size, computed using (4.1).

Every  $\alpha_j$  is compared with the radius of the Delaunay circumcircles or circumspheres that include the particle. This provides a criterion to accept or throw away new generated elements in the Delaunay tessellation. This process is shown in Figure 4.9.

The approximation described can be presented as a new strategy for the generation of a mesh and the recognition of the boundary. One effective procedure consists in building a Delaunay triangulation taking in account the whole convex hull. Once the mesh is created, including all the domain, the elements in the mesh are redefined and filtered with the  $\alpha$  parameter. The filter value is defined for each element by the value of  $\alpha_j$  of its nodes (particles). Figure 4.9 illustrates this strategy.

Using this scheme the elements that are finally accepted yield the domain shape while the rest are just not considered any further.

If a particle does not belong to any element, the particle is not considered as a part of the domain. There are several options for the treatment of isolated particles. The first one is let them move freely, because they can be rejoined to the body in future time steps. The second one is remove the particle from the analysis. This is recommended when particles do not contribute anymore to the mechanical properties of the domain.

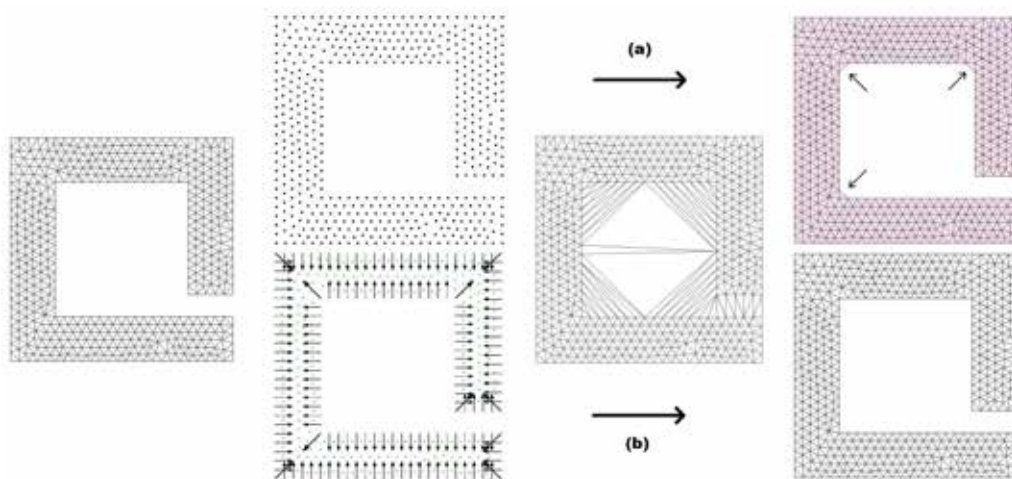


**Figure 4.9:** Flowchart for meshing and boundary recognition in the PFEM. From a cloud of points a mesh is defined and filtered by means of the  $\alpha$ -shape criterion.

#### 4.4.2 Refining $\alpha$ -shapes

From the definition in Section 4.3.1,  $\alpha$ -shapes are a generalization of the convex hull of a point set. It means that all convex volumes defined by a set of points will be meshed. Intrinsically any hole inside the  $\alpha$ -hull will be formed as a function of the  $\alpha$  parameter values.

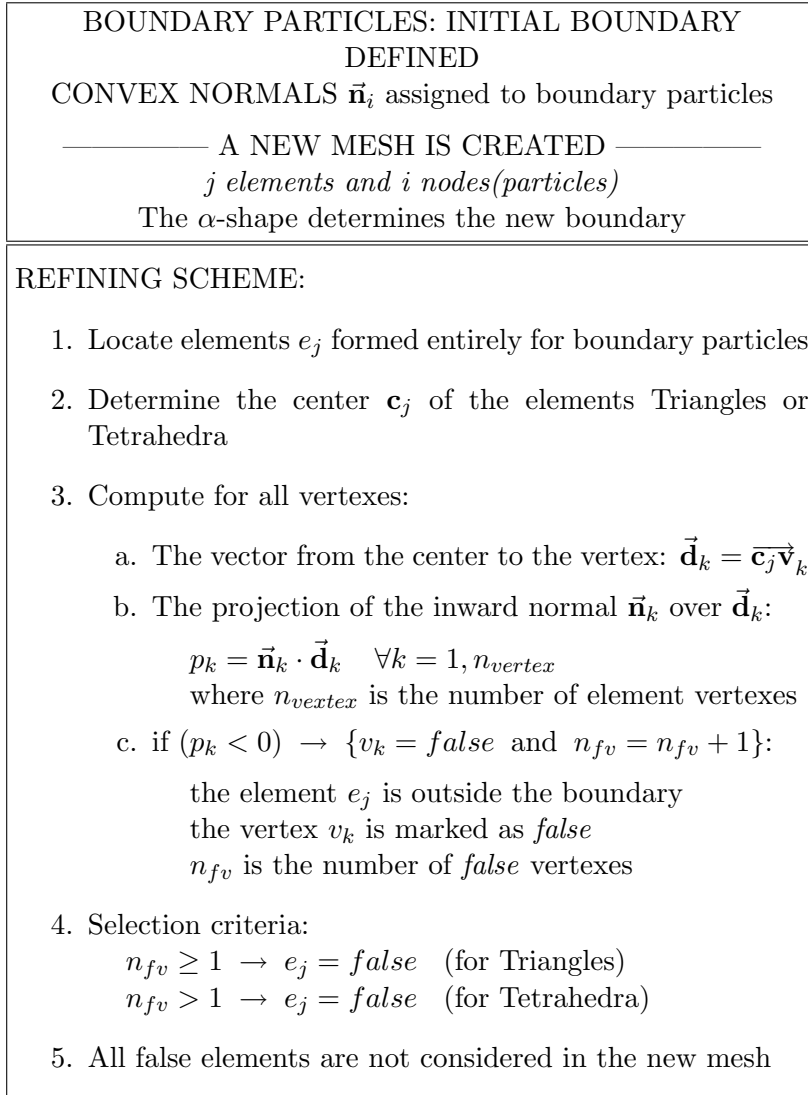
Starting from a given geometry, a distribution of points is defined to describe the body domain. This cloud of points is remeshed again using non-constrained uniform (or non-uniform)  $\alpha$ -shapes. When empty convex edges or empty holes are inside the domain, the  $\alpha$ -shapes do not reproduce the initial geometry anymore (Figure 4.10). Some elements in the edges pass the alpha-shape filtering.



**Figure 4.10:** Starting from an initial two-dimensional geometry (left figure), a cloud of points is defined and a new mesh is created. (a) Resultant mesh applying only the  $\alpha$ -shape concept (b) Result after using the normals to the boundaries and refining the  $\alpha$ -shape

To overcome this small deficiency in the boundary detection an additional algorithm can be applied after mesh generation. The algorithm polish the  $\alpha$ -shape and the original geometry is recovered.

The algorithm is based in the convex nature of the elements formed in the edges or in the holes, which must be polished. The criteria to consider or not the elements is based on the boundary normals to the element nodes. The normals at the boundaries are defined pointing to the interior of the body domain (inward normals). Firstly, the element formed entirely by boundary particles is identified. When the inward normals to the boundary surface point toward the exterior of the element, that element is tagged outside the geometry and is not considered. Box 4.1 summarizes the flowchart for the *refining scheme*.



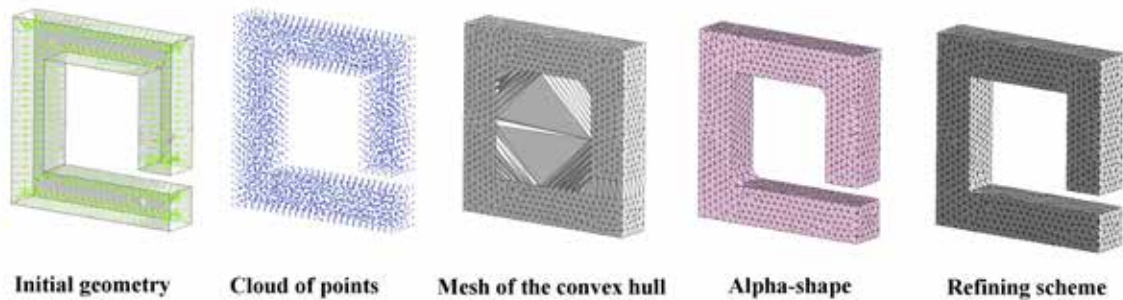
Box 4.1: Flowchart for the boundary refinement. The refining scheme is complementary to the  $\alpha$ -shape process.

The remeshing flowchart is presented in Figure 4.11. The difference between using only  $\alpha$ -shapes for the boundary recognition and complementing it with the refining scheme can be seen in Figures 4.10 and 4.11.

#### 4.4.3 Data structures and algorithms

In the remeshing process for the PFEM some particular data structures and geometrical algorithms are needed. This section will not explain on detail the algorithms but will give some references where they can be found.

In the developed code the Delaunay tessellation has not been programmed, free-ware



**Figure 4.11:** The boundary normals are defined in the initial three-dimensional geometry (on the left), a cloud of points determine the domain and a new mesh is made. First the  $\alpha$ -shape concept is applied and complemented with the refining scheme.

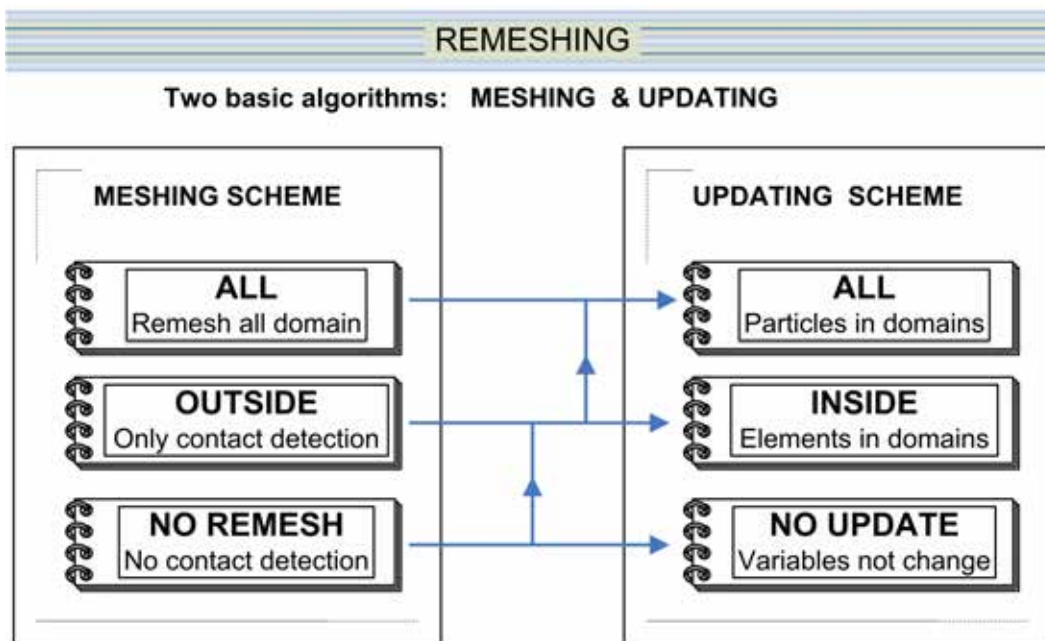
codes have been used as libraries. The two dimensional mesher is called Triangle [5], and the three dimensional one TetGen [4]. The cloud of particles that define the domain are passed to the mesher and the new connectivities are obtained as output.

The required data structures are vectors and containers for the particle properties. The design of these containers is a choice of the programmer. Some basic rules and effective algorithms can be found in [74]. It is important to have a rapid access and manipulation of the information. There are some needed data structures for static data as: points surrounding points, elements surrounding points, elements surrounding elements and external faces. It is also important how to assign the normal vector for every node or for every surface element.

After meshing the information of the particles has to be interpolated to the added particles. The numbering has to be updated and the container information relocated.

For meshing some different schemes are possible. Choosing one or another will depend on the problem requirements. Figure 4.12 shows the main possible strategies for remeshing. The first important thing is to distinguish two steps in the meshing process: the creation of a new mesh (**MESHING**) and the update of the variables in the new mesh (**UPDATING**). If a new mesh is created the variables of the problem must be updated and renumbered. Nevertheless if the mesh remains the same or some parts of the domain are not remeshed, the variables can be preserved on elements.





**Figure 4.12:** Remeshing schemes. Arrows show the possible updating schemes after remeshing. Last scheme NO REMESH - NO UPDATE coincides with the classical FEM.

### Remeshing scheme

In the remeshing scheme of Figure 4.13 the number of particles can change. Two possible reasons for a change in the number of particles are: a mesh refinement for the complete domain or for part of it, or a duplication of particles to create fracture lines inside the bodies. These two possibilities increase the number of particles. The information will have to be interpolated carefully for each case. Other possibility is the lose of particles. Particles will be taken away when they are not going to contribute anymore to the mechanical process. In this case it has to be ensured that the information contained in these particles is not needed and does not influence the rest of the domain. In the PFEM scheme implemented in this work only the removal of particles is applied. Other options are considered for further applications.

Once the mesh is created the updating of the problem variables must be performed. Note that elements are not conserved as a new mesh is created at each time step. Hence there are new integration points not coinciding with the integration points of the prior mesh. The possibilities for the updating are presented in Figure 4.14.

## REMESHING

Flowchart for the remeshing in the PFEM: **INPUT : VARIABLES FROM A CONVERGED STEP**

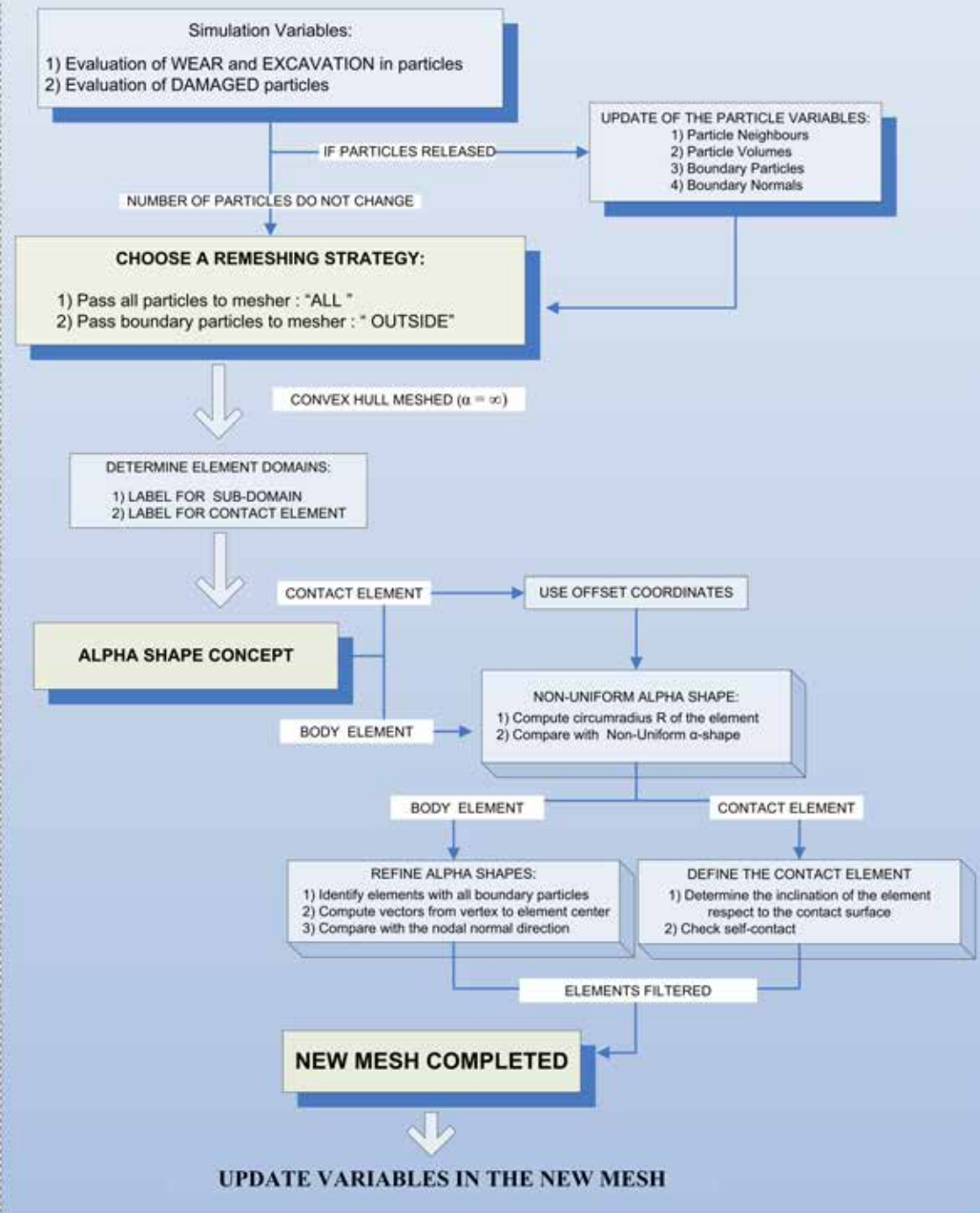
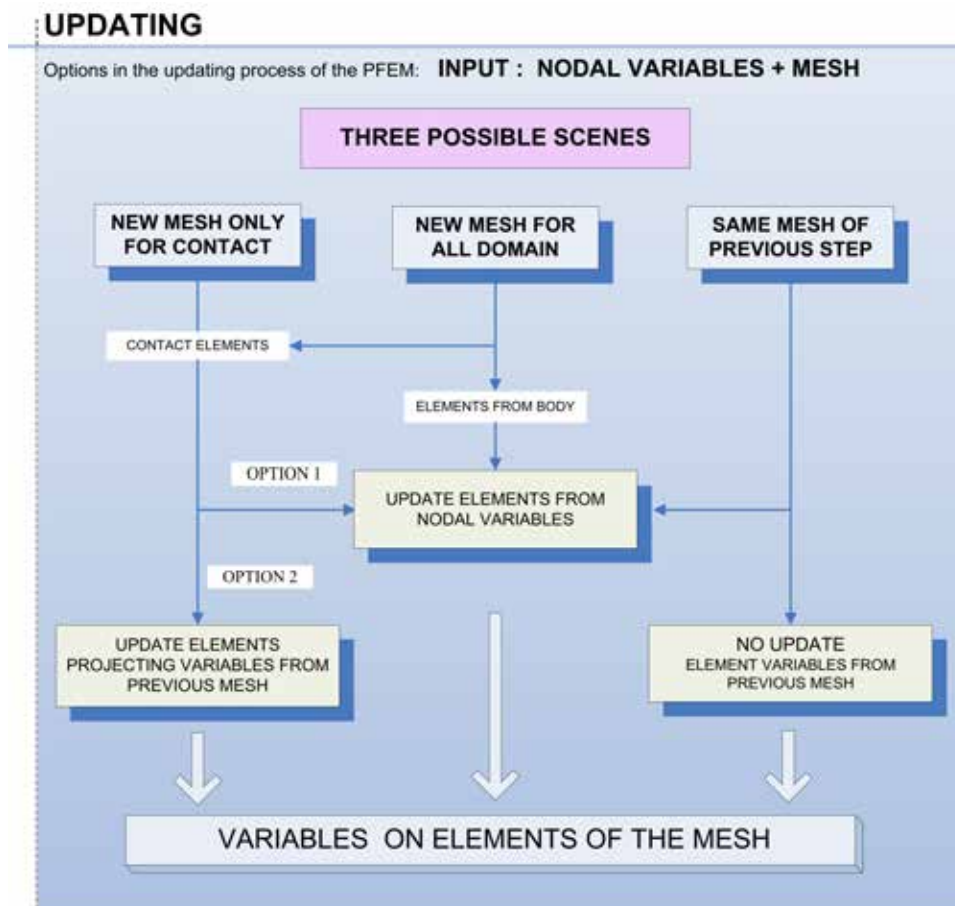


Figure 4.13: Remeshing scheme for the PFEM

## Updating Scheme

There are two important things to comment in Figure 4.14. The first one refers to the case when a mesh does not change. Here the method is analogous to the classical FEM. This is good because you can activate or deactivate the remeshing process if not needed and have a scheme based on particles or in elements according to one's wish. In this situation it is not necessary to transfer information to the particles. When the particle based scheme is required the nodal information will be requested. It is preferable to keep the same mesh as much as possible to reduce the number of mesh-particle transfers.

The other important thing to comment is the special option for updating contact elements. Because contact is a problematic and sensitive issue, the option to project the stresses from a mesh to another is considered. That is because the contact mesh is formed by a single layer of elements and a smoothing of the historical variables affects it particularly.



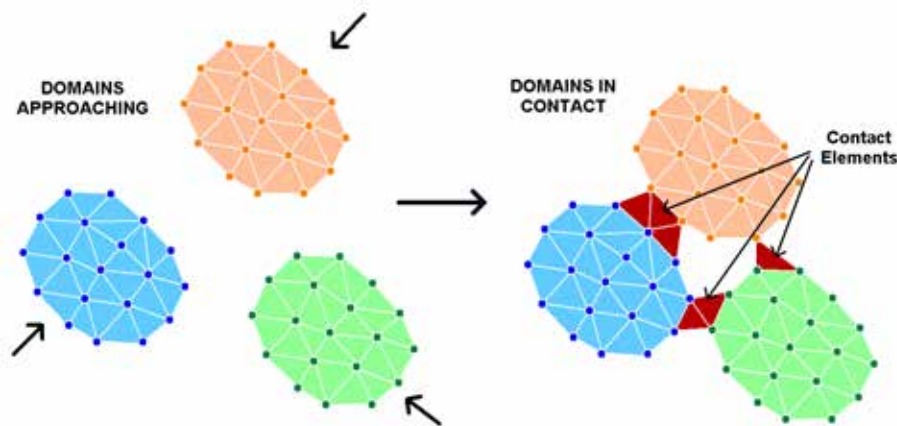
**Figure 4.14:** Updating scheme for the PFEM after the remesh of the domain

## 4.5 Geometric contact detection

The PFEM is designed for the treatment of large displacements and large deformations of continuum domains as well as for multi-body systems. The most important physical phenomena that has to be represented is the interaction between multiple bodies.

The first difficulty is to detect when two sub-domains (bodies) enter in contact. In last sections the remeshing and boundary recognition has been explained. The system is composed by a cloud of particles that will describe a body with a boundary and an interior domain. The contact identification is naturally done by the alpha shape concept.

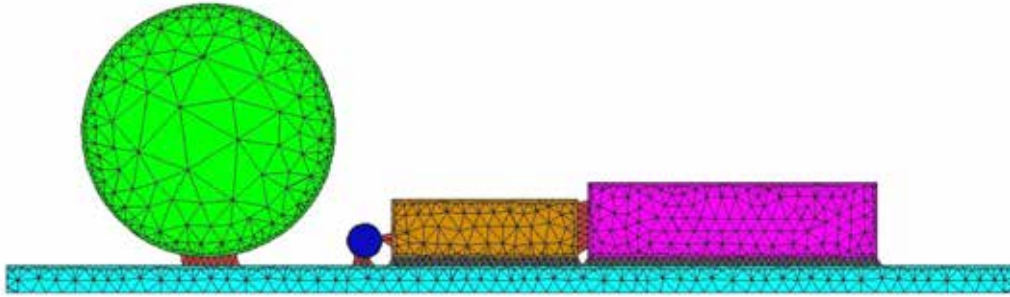
In order to define different entities inside the cloud of points, each particle has a flag for body identification. The alpha-shape method does not distinguish the different bodies or sub-domains and hence the complete particle domain is remeshed. When the particles of two sub-domains get closer the alpha-shapes joint these two sub-domains and create a larger domain. The detected boundary will contain particles of different bodies. An interface of elements between the two sub-domains is created. These elements are called *contact elements*. The process is displayed in Figure 4.15.



**Figure 4.15:** Using the alpha-shapes the detection of contact between sub-domains is done automatically. This generates contact elements when the sub-domains are close enough.

The way to detect contact in the PFEM does not add extra cost in the analysis. The interface elements that define geometrical contact are used to define the mechanical contact. This will be explained in Chapter 5.

This method for detecting the geometrical interaction between bodies is computationally cheap but has some drawbacks. Looking at Figure 4.16 one can realize the most obvious one. There is a layer containing an element between the sub-domains in contact. This disadvantage is solved by extending the strategy of the boundary detection to a new meshing scheme.



**Figure 4.16:** Different sub-domains in contact. An interface mesh is created by the *Delaunay* tessellation using  $\alpha$ -shapes.

#### 4.5.1 Settling the boundaries

Contact is detected when an interface layer of elements is created joining two different sub-domains. When the purpose of the remeshing is only the detection of contact, a simplified strategy can be adopted in order to find this interface.

Particles within a sub-domain do not take part in the contact layer of elements. The contact mesh can be created with a Delaunay tessellation of the boundary particles, without considering the interior ones. Hence the outward mesh, generated by the convex-hull formed by the particles of the boundary, defines the contact mesh.

The process requires some extra considerations. The first one is the previous identification of the boundary particles for every sub-domain. The process requires a search of the boundary faces of the elements. Effective algorithms can be found in [74]. The second consideration is that if a new mesh is not needed for the sub-domains, the interior mesh can be kept the same. This is advantageous, because in these cases the elemental information of the sub-domains can remain the same and no transfer is needed. Remember that any transfer implies a loss of information and a smoothing of the variables; see Section 3.5.

When sub-domains need to be remeshed, two independent meshing processes are needed. One process remeshes the interior of the sub-domain and the other remeshes the outside domain using the particles on the boundary. That would be an ineffective way of doing the detection if the particles on the boundary are the same in the two cases, as the second tessellation would not be necessary. However the independent meshing of the particles on the boundary is a good way to have a better performance of the layer of elements between contacting sub-domains.

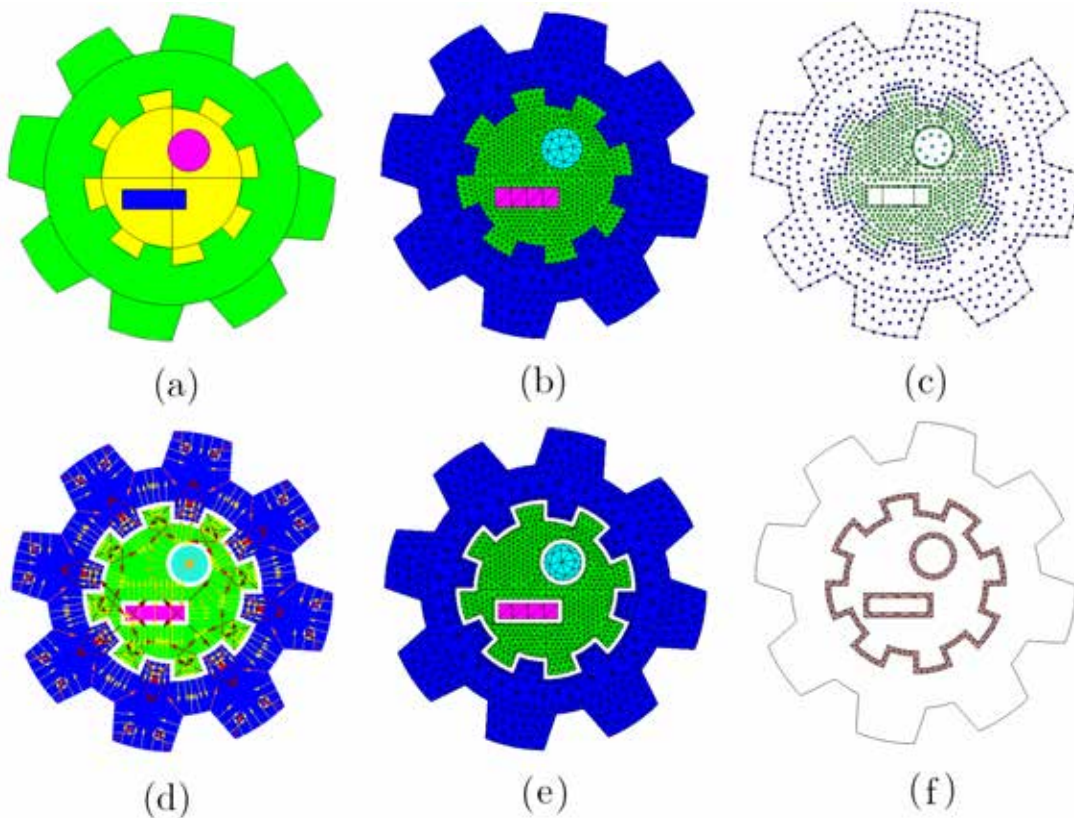
#### 4.5.2 Contracting boundaries

The boundaries of each sub-domain will be interconnected with a layer of elements (contact elements) when the respective boundary particles are close enough. Let us consider now that a geometric offset contracting the boundary particles is done. The interface mesh will not be detected at the same space position because the sub-domains will not be joined for the interface mesh after being filtered by the alpha shape technique.

In order to obtain the real geometrical contact, a contraction of the settled boundaries can be used before remeshing the outside of the sub-domains. An offset of the correct distance yields the real contact position without introducing any artificial layer of elements between them. The contact elements will exist and occupy some space in the contracted geometry but not in the real one. This strategy is adopted in our work in order to get the best geometrical accuracy in the contact detection. Figure 4.18 shows the process of offsetting the sub-domains and detecting the contact.

### Offset algorithm

The contraction of the boundary of a geometry is carried out using the discretization of the surface. The points discretising the surface mesh of the body and the normals of the surface have to be defined. The offset of these points will describe the contracted or expanded surface of the body. See Figure 4.17.



**Figure 4.17:** Offset algorithm: (a) Original geometry, (b) Domain discretization, (c) Identification of the contour particles, (d) Inward normals defined on contour particles, (e) Contracted body mesh, (d) Contact interface mesh.

To implement the offset scheme, the algorithm described in Box 4.2 is used for each particle of the boundary.

1. Set the associated surface elements for each particle of the surface
2. Compute the inwards normal unit vector  $\mathbf{n}_i$  for each surface element associated to a particle
3. Simplify coinciding normal unit vectors, and average nearly coinciding vectors for all surface elements associated to the particle:

$$c_i = \begin{cases} 0 & \text{if } \langle \mathbf{n}_i | \mathbf{n}_j \rangle \geq 0.995 \\ 0 & \text{if } 0.95 \geq \langle \mathbf{n}_i | \mathbf{n}_j \rangle \leq 0.995 \\ 1 & \text{if } \langle \mathbf{n}_i | \mathbf{n}_j \rangle \leq 0.95 \end{cases} \Rightarrow \bar{\mathbf{n}}$$

for  $i = 1, \dots, n_s \quad j = i + 1, \dots, n_s$

(4.2)

where  $n_s$  is the number of neighbor surface faces.  $\bar{\mathbf{n}}$  is a linear average of all vectors that satisfy the condition.  $c_i$  is the variable that describes the vector contribution

4. Compute the unit offset vector for each particle  $\mathbf{v}_j^{set}$ :

$$\mathbf{v}_j = \sum_{i=1}^{n_s} \mathbf{v}_i c_i + \bar{\mathbf{n}} \quad \mathbf{v}_j^{set} = \frac{\langle \mathbf{v}_j^{set} | \mathbf{v}_j^{set} \rangle}{\| \langle \mathbf{v}_j^{set} | \mathbf{v}_j^{set} \rangle \|}$$
(4.3)

5. Compute the relative offset parameterized modulus  $a^{set}$  as the average cosine director with respect the neighbor normals:

$$a_j^{set} = \frac{\langle \mathbf{v}_j^{set} | \mathbf{v}_i c_i \rangle}{n_s}$$
(4.4)

The resultant parameterized offset vector is

$$\hat{\mathbf{v}}_j^{set} = \frac{\mathbf{v}_j^{set}}{a_j^{set}}$$
(4.5)

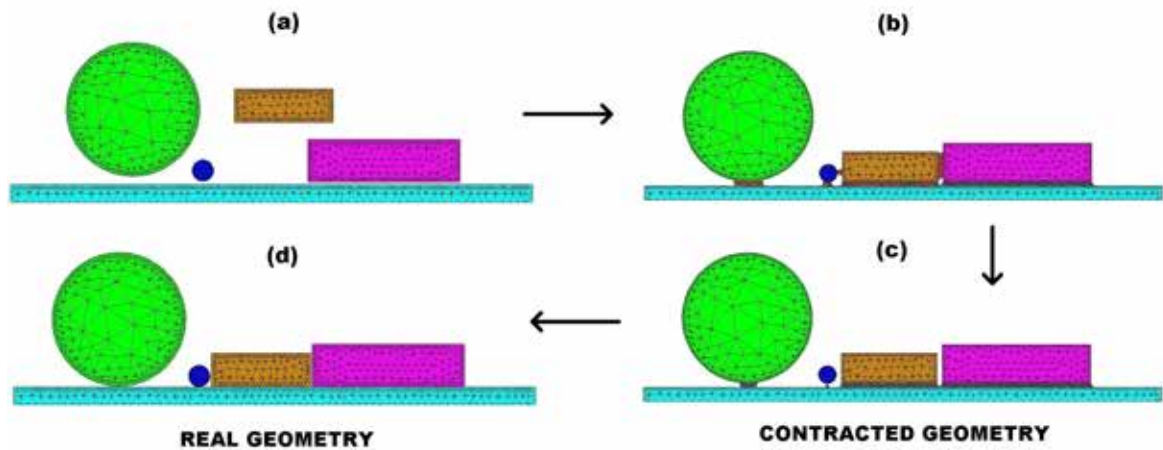
Vector  $\hat{\mathbf{v}}_j^{set}$  gives the contracted location of the new point for a unit distance.

6. Apply the correct offset distance  $d_c$  to each point of the boundary surface

$$\mathbf{x}_j = \mathbf{x}_j + \hat{\mathbf{v}}_j^{set} \cdot d_c$$
(4.6)

Box 4.2: The offset algorithm to obtain the contracted or expanded surface

To set the correct distance for the contraction  $d_c$ , the size of the elements of the interface has to be calibrated. In order to obtain a suitable value for  $d_c$  the contraction is not applied until the first contact occurs. After that, the minimum normal distance between the contacting surfaces defines the thickness of the layer of elements. Half of this thickness is applied for the contraction of the sub-domains. An expression for  $d_c$  will be deduced in the next chapter (see equation (5.47)). This gives the exact geometrical contact when the contracted surfaces of the sub-domains interact again. See Figure 4.18.



**Figure 4.18:** Contact detection and offset algorithm: (a) Original state and geometry, (b) After first detection of active contact elements (black elements) the offset of the boundary is applied, (c) Contracted geometry, (d) Exact geometrical contact.

The drawback of the contraction scheme is that is delimited by the size of the discretization of the body. The size of the contraction  $d_c$  has to be smaller than the length of the surface elements. If  $d_c$  is larger, the contracted surface will not maintain the proportionality with the original geometry. Edges are the geometrical entities that are more influenced by the size limitation.

Unstructured sizes of elements in the surface will not be a problem due to the non-uniform assignation of the alpha shapes (see Section 4.4.1). As a consequence of this the contact layer will be composed of different element sizes. However this has not a negative influence in the computation of the mechanical behavior of the contact interaction. The only aspect to be taken into account is that a better discretization leads to better results, as is usual in finite element analysis.





*All truths are easy to understand  
once they are discovered;  
the point is to discover them.*

Galileo Galilei (1564-1642)

# 5

## Contact mechanics

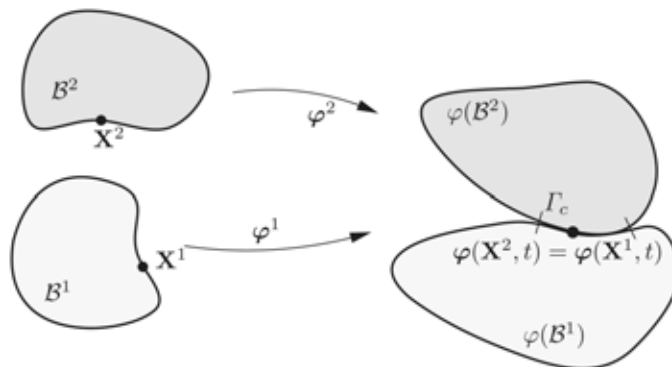
### 5.1 Introduction

In this chapter the basis of contact kinematics and kinetics is explained. As an state of the art a small explanation of the main statements is presented. The extension of this explanation can be found in the specialized literature (specially in Wriggers [132] and in [135], [13] and [134]).

### 5.2 Contact kinematics

Here the formulation for all contact relations in finite deformations is presented. The problem appears when two or more bodies  $\mathcal{B}^\alpha$  approach each other during a finite deformation process and come into contact on parts of their boundaries denoted by  $\Gamma_c$ . In Figure (5.1) we can observe that two points,  $\mathbf{X}^1$  and  $\mathbf{X}^2$ , in the initial configuration of the bodies which are distinct can occupy the same position in the current configuration,  $\varphi(\mathbf{X}^2) = \varphi(\mathbf{X}^1)$ , within the deformation process. Hence this implies that contact conditions have to be formulated with respect to the current configuration.

In general, two steps have to be performed to detect whether or not contact takes place. These steps are the global search for contact and the set-up of the local kinematic relations for the interaction. The contact search has been introduced in Section 4.5 and will be discussed extendedly in Section 5.5. Next sections are going to define the kinematic



**Figure 5.1:** Finite deformation contact [132])

relations due to normal and tangential contact.

In the continuum-based formulation of contact, kinematic relations are needed to formulate the contact constraints.

There are two classical approximations for the contact kinematics, the non-penetration condition [34] and the allowance of small penetration [15].

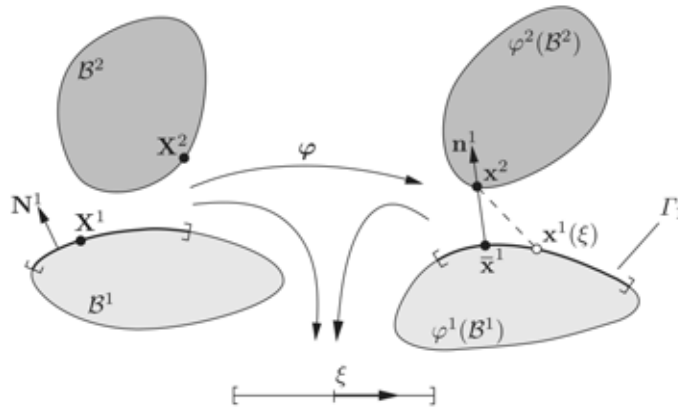
In both approaches the distance between the bodies in contact tries to be minimized in order not to have any penetration. The non-penetration condition is very important for the definition of the tangential velocity in the contact interface, which is needed to formulate frictional problems. There are several references for that in [121], [15], [71], or [107].

The theory of contact starts considering two elastic bodies  $\mathcal{B}^\alpha$ ,  $\alpha = 1, 2$ , each occupying the bounded domain  $\Omega^\alpha \subset R^3$ , see Figure (5.1). The boundary surface  $\Gamma^\alpha$  of a body  $\mathcal{B}^\alpha$  is split in three different parts. One part of the boundary surface  $\Gamma^\alpha$  corresponds to the area of prescribed surface loads  $\Gamma_\sigma^\alpha$ , another part of the boundary surface has the prescribed displacements  $\Gamma_u^\alpha$ , and there is a third part  $\Gamma_c^\alpha$  where the two bodies  $\mathcal{B}^1$  and  $\mathcal{B}^2$  come into contact. In the contact area, the constraint equations must be formulated. They are composed of an approach for defining the normal contact, as well as kinematic relations for the tangential contact.

### 5.2.1 Normal Contact

Looking at Figure (5.2) and assuming that the two bodies come into contact, the non-penetration condition is formulated. It is based in the normal direction and is given by (5.1):

$$(\mathbf{x}^2 - \mathbf{x}^1) \cdot \mathbf{n}^1 \geq 0 \quad (5.1)$$



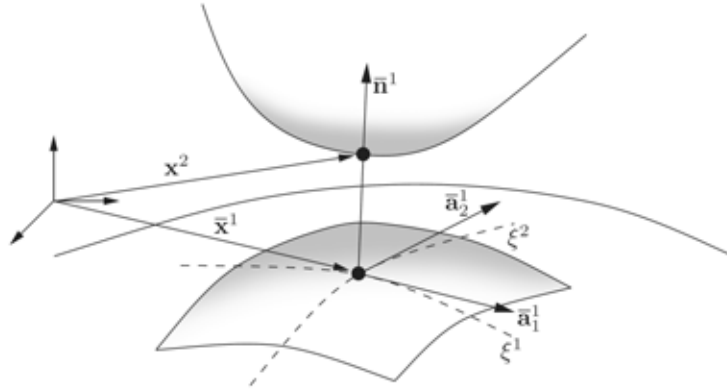
**Figure 5.2:** Deformed configuration  $\mathcal{B}^\alpha$  and minimum distance between bodies [132]

where  $\mathbf{x}^\alpha$  are the coordinates of the current configuration  $\varphi(\mathcal{B}^\alpha)$  of the body, see (2.2).

The normal vector  $\mathbf{n}^1$  is associated with body  $\mathcal{B}^1$ . To find the point where the normal has to be computed the criteria of the minimum distance is applied.

This criteria assumes that the contact boundary describes, at least locally, a convex region. Hence, every point  $\mathbf{x}^2$  on  $\Gamma^2$  can be related with a point  $\bar{\mathbf{x}}^1 = \mathbf{x}^1(\bar{\boldsymbol{\xi}})$  on  $\Gamma^1$ . This correspondence is done via the minimum distance between bodies. Looking at Figure (5.3) the minimum distance can be defined as:

$$\hat{d}^1(\xi^1, \xi^2) = \|\mathbf{x}^2 - \bar{\mathbf{x}}^1\| = \min \{ \|\mathbf{x}^2 - \mathbf{x}^1(\boldsymbol{\xi})\| \mid \mathbf{x}^1 \subseteq \Gamma^1 \} \quad (5.2)$$



**Figure 5.3:** Minimum distance problem and coordinate system [132]

Point  $\bar{\mathbf{x}}^1$  is computed from the necessary condition for the minimum distance. This is the reason why in equation (5.2)  $\boldsymbol{\xi} = (\xi^1, \xi^2)$  represents the parametrization of the boundary  $\Gamma^1$  using convective coordinates. They are used for an accessible treatment of the variables in the formulation of the minimum distance condition. A description of the convective coordinates can be found in [15], [133] or [71].

The minimum distance will be used later to define the gap or penetration between the two bodies, when a small penetration is allowed.

In Figures (5.2) and (5.2) the normal vector  $\mathbf{n}^1$  is depicted. It is defined at the point  $(\bar{\xi}^1, \bar{\xi}^2)$  defining the minimum distance. There is also vector  $\mathbf{a}_\alpha^1$ , which is one of the tangent vectors of the plane  $\alpha$ . Thus the condition  $-\mathbf{n}^1(\bar{\xi}^1, \bar{\xi}^2) \cdot \mathbf{a}_\alpha^1(\bar{\xi}^1, \bar{\xi}^2) = 0$  means that the current point  $\mathbf{x}^1(\bar{\xi}^1, \bar{\xi}^2)$  is the orthogonal projection of a given point  $\mathbf{x}^2$  onto the current surface  $\varphi_t^1(\Gamma_c^1)$ . Note: here and in the following, the bar over a quantity denotes that is computed at the minimum distance point.

The definition of the outward unit normal on the current surface  $\varphi_t^1(\Gamma_c^1)$  at the master point  $\mathbf{x}^1$  is expressed as:

$$\bar{\mathbf{n}}^1 = \frac{\bar{\mathbf{a}}_1^1 \times \bar{\mathbf{n}}_2^1}{\|\bar{\mathbf{a}}_1^1 \times \bar{\mathbf{n}}_2^1\|} \quad (5.3)$$

where  $\bar{\mathbf{a}}_\alpha^1$  are tangent vectors at  $\mathbf{x}^1(\bar{\xi}^1, \bar{\xi}^2)$ . The normal can also be defined as:

$$\bar{\mathbf{n}}^1 = \frac{\mathbf{x}^2 - \mathbf{x}^1(\bar{\xi}^1, \bar{\xi}^2)}{\|\mathbf{x}^2 - \mathbf{x}^1(\bar{\xi}^1, \bar{\xi}^2)\|} \quad (5.4)$$

with the only inconvenience that is not defined for  $\|\mathbf{x}^2 - \mathbf{x}^1(\bar{\xi}^1, \bar{\xi}^2)\| = 0$ .

After the definition of the projection  $\bar{\mathbf{x}}^1$  and the normal direction  $\bar{\mathbf{n}}^1$ , the classical approximations for the contact kinematics are defined. The inequality constraint for the *non-penetration condition* yields

$$g_N = (\mathbf{x}^2 - \bar{\mathbf{x}}^1) \cdot \bar{\mathbf{n}}^1 \geq 0 \quad (5.5)$$

and the *penetration function* is defined as

$$\bar{g}_N = \begin{cases} (\mathbf{x}^2 - \bar{\mathbf{x}}^1) \cdot \bar{\mathbf{n}}^1 & \rightarrow \text{if } (\mathbf{x}^2 - \bar{\mathbf{x}}^1) \cdot \bar{\mathbf{n}}^1 < 0 \\ 0 & \rightarrow \text{otherwise} \end{cases} \quad (5.6)$$

The magnitude of penetration of one body into the other has to be used in conjunction with the contact method. The penalty method makes use of function  $\bar{g}_N$ ; see section 5.4.

For geometrically linear kinematics, it is advantageous to write the inequality constraint as

$$g_N = (\mathbf{x}^2 - \bar{\mathbf{u}}^1) \cdot \bar{\mathbf{u}}^1 + g_X \geq 0 \quad (5.7)$$

where  $g_X = (\mathbf{X}^2 - \bar{\mathbf{X}}^1) \cdot \bar{\mathbf{n}}^1$  is the initial gap between two bodies. In this case the penetration function can be written as

$$\bar{g}_N = \begin{cases} (\mathbf{u}^2 - \bar{\mathbf{u}}^1) \cdot \bar{\mathbf{n}}^1 + g_X & \rightarrow \text{if } (\mathbf{u}^2 - \bar{\mathbf{u}}^1) \cdot \bar{\mathbf{n}}^1 + g_X < 0 \\ 0 & \rightarrow \text{otherwise} \end{cases} \quad (5.8)$$

There are some problems related with the determination of the minimum distance when the function is locally non-differentiable. The definition of the surface normal using finite elements is also a handicap as here exists a discontinuity in the normal field. The discussion of these problems can be found in [132].

### 5.2.2 Tangential Contact

Two cases are typically distinguished in the tangential direction of the contact interface. The first one is the so-called *stick state* in which a point which is in contact is not allowed to move in a tangential direction. The second case is the *sliding state*, which means that a point moves in a tangential direction at the contact interface. This movement can only be described by the relative deformation in the interface.

#### Stick Condition

When one point sticks to another body it does not move in a tangential direction, and hence the computed values for the convective coordinates of that point  $(\bar{\xi}^1, \bar{\xi}^2)$  do not change during the motion. The mathematical condition for stick can be formulated as:

$$\mathbf{g}_T = g_{T\alpha} \bar{\mathbf{a}}^\alpha = 0 \quad \text{where} \quad g_{T\alpha} = (\mathbf{x}^2 - \bar{\mathbf{x}}^1) \cdot \bar{\mathbf{a}}_\alpha^1 \quad (5.9)$$

Here  $\mathbf{g}_T$  denotes the relative displacement in a tangential direction, which has to be zero.

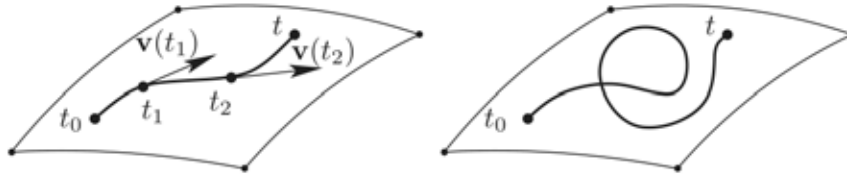
In the stick case it is not necessary to distinguish between normal and tangential directions. Thus, when the normal gap is closed,  $g_N = (\mathbf{x}^2 - \bar{\mathbf{x}}^1) \cdot \bar{\mathbf{n}}^1 = 0$  the conditions  $g_N = 0$  and  $\mathbf{g}_T = 0$  are imposed with a more simple condition.

$$\mathbf{x}^2 - \bar{\mathbf{x}}^1 = 0 \quad (5.10)$$

### Slip Condition

When the surfaces of the bodies describe a relative deformation, the change of the point  $\mathbf{x}^2$  related to its projection  $\bar{\mathbf{x}}^1$  defines the tangential slip (see Figure (5.2)). This means that the solution point, which has been obtained via the minimal distance problem will move on the body surface.

Usually the path of a point  $\mathbf{x}^2$  on the master surface is not known a priori. It could even cross its own line, as shown in Figure (5.4). The only known variable is the relative velocity vector. The velocity  $\mathbf{v}$  of point  $\mathbf{x}^2$  relative to the contact surface is known at times  $t_1$  and  $t_2$ .



**Figure 5.4:** Path of the point  $\mathbf{x}^2$  relative to the contact surface [132]

Therefore in a frictional sliding situation the integration of the relative velocities will be the way to find the path of  $\mathbf{x}^2$  on the master surface. After stating a tangential relative velocity of a point  $\mathbf{x}^2$  on the contact surface, defined in terms of the body  $\mathcal{B}^1$ , the path of the point on  $\Gamma^1$  is computed from

$$d\mathbf{g}_T = \bar{\mathbf{a}}_\alpha^1 d\xi^\alpha = \bar{\mathbf{x}}_{,\alpha}^1 d\xi^\alpha \quad (5.11)$$

where  $\bar{\mathbf{a}}^\alpha$  are the tangent vectors evaluated at the projection point  $\xi^\alpha$ . The length of the frictional path can be computed by

$$\mathbf{g}_T = \int_{t_0}^t \|\dot{\xi}^\alpha \bar{\mathbf{x}}_{,\alpha}^1\| dt \quad (5.12)$$

where  $t$  is the time used to parameterize the path of point  $\mathbf{x}^2$ . The equalities  $d\mathbf{g}_T = \|d\mathbf{g}_T\|$  and  $\dot{\xi}^\alpha dt = d\xi^\alpha$  has been considered to define (5.12). See Figure (5.3).

### 5.2.3 Normal gap for the PFEM

The functions for the kinematics of contact are modified slightly for a correct treatment in the PFEM. Contact is activated before the surfaces of the bodies  $\mathcal{B}^1$  and  $\mathcal{B}^2$  encounter. That requires a different definition for the gap functions. An initial predefined gap  $\hat{g}_N$  will determine a distance for the subsequent contact. This distance will be a function of the  $\alpha$ -shapes of the finite elements associated to the contact interface (see Section 4.4.1). The definition of  $\hat{g}_N(\alpha_i)$  is presented in Section 5.5.2 (equation (5.47)).

The inequality constraint for the non-penetration condition in that case is

$$g_N = (\mathbf{x}^2 - \bar{\mathbf{x}}^1) \cdot \bar{\mathbf{n}}^1 - \hat{g}_N \geq 0 \quad (5.13)$$

where  $\hat{g}_N$  is the prior gap between the two bodies. In this case, the penetration function defines if the contact is active or not; i.e.

$$\bar{g}_N = \begin{cases} true & \rightarrow \text{if } (\mathbf{x}^2 - \bar{\mathbf{x}}^1) \cdot \bar{\mathbf{n}}^1 - \hat{g}_N < 0 \\ false & \rightarrow \text{otherwise} \end{cases} \quad (5.14)$$

when  $\bar{g}_N = true$  then  $g_N = (\mathbf{x}^2 - \bar{\mathbf{x}}^1) \cdot \bar{\mathbf{n}}^1 - \hat{g}_N$ , that is necessary for notation consistency for the imposition of the correct contact constraints.

For the tangential contact, the definition of the tangential gap does not change (see equation (5.12)).

## 5.3 Contact boundary value problem

### 5.3.1 Constraint formulation

The most general consideration is that normal contact is only a constraint for non-penetration. There are other possibilities that consist in introducing constitutive equations which describe the approach for treating the contacting surfaces.

For the case of frictionless contact, it suffices to impose normal contact with a non-penetration condition or a constitutive model. However, when friction is considered, the tangential stress has to be computed from the kinematics and constitutive equations.

#### Normal contact constraint

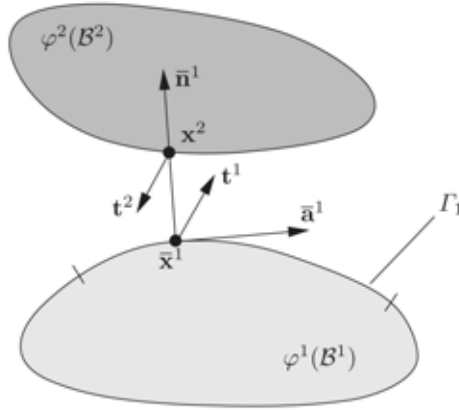
Imposing normal contact as an unilateral constraint consists essentially in enforcing the geometrical constraints in the correct way. Numerous researchers have used this strategy, e.g. see [60] or [63]. In this case, it is not necessary to specify constitutive relations in the contact interface. The contact pressure is obtained as a reaction in the contact surface. Contact takes place when  $g_N$  is equal to zero.

When that occurs, the associated normal component of the stress vector in the contact interface must be non-zero. The stress vector is expressed as

$$\mathbf{t}^1 = \boldsymbol{\sigma}^1 \bar{\mathbf{n}}^1 = p_N^1 \bar{\mathbf{n}}^1 + \mathbf{t}_T^{1\beta} \mathbf{a}_\beta^1 \quad (5.15)$$

where  $p_N^1$  is the normal component of the stress vector. Following the principle of action-reaction, the stress vector acts in the contact point of both surfaces and  $\mathbf{t}^1 = -\mathbf{t}^2$ ;

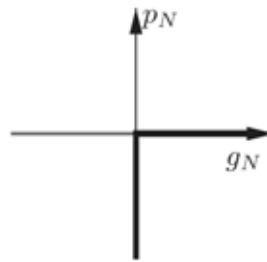
see Figure 5.5. The tangential stress  $\mathbf{t}_T^{1\beta}$  is zero for frictionless contact and in absence of adhesive stresses the normal component of the stress vector must be compressive, i.e.  $p_N = p_N^1 = p_N^2 < 0$ .



**Figure 5.5:** Stresses in the contact interface [132]

The above statements define the *Kuhn Tucker* conditions (5.16) for contact. They describe a non-smooth contact law for the normal contact pressure (see Figure 5.6); i.e.

$$\boxed{g_N \geq 0, \quad p_N \leq 0, \quad g_N p_N = 0} \quad (5.16)$$



**Figure 5.6:** Contact force versus normal gap [132]

Taking into account that  $\mathbf{t} = \boldsymbol{\sigma} \mathbf{n}$  the normal and tangential components of the contact stress vector can be written as

$$\mathbf{t} = \mathbb{P} \mathbf{t} + (\mathbb{I} - \mathbb{P}) \mathbf{t} \quad (5.17)$$

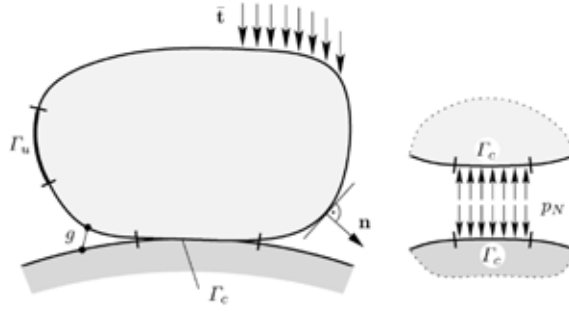
with  $\mathbb{P} = \bar{\mathbf{n}}^1 \otimes \bar{\mathbf{n}}^1$ , the projection tensor and the fourth order unit tensor  $\mathbf{t} = \mathbb{I} \mathbf{t}$ .



### 5.3.2 Frictionless contact in finite deformations

The constraint equations introduce additional terms in the formulation of the boundary value problem. Due to the contact constraints a variational inequality will appear instead of the standard variational equations of classical solid mechanics; see Section 2.3.2.

For the formulation of the boundary value problem a solid with a rigid surface is considered. The motion of the body is constrained in one side by the rigid surface  $\Gamma_u$ ; see Figure 5.7.



**Figure 5.7:** Unilateral contact of a solid [132]

Firstly the local momentum equation for each body in contact  $\mathcal{B}^\gamma$  with  $(\gamma = 1, 2)$ , is formulated as

$$\operatorname{div} \boldsymbol{\sigma}^\gamma + \rho^\gamma \bar{\mathbf{b}}^\gamma = 0 \quad (5.18)$$

The deformation field is defined as

$$\boldsymbol{\varphi}^\gamma = \boldsymbol{\varphi}^\gamma(\mathbf{u}) \quad (5.19)$$

and the boundary conditions for the deformation and the stress field as

$$\boldsymbol{\varphi}^\gamma = \bar{\boldsymbol{\varphi}}^\gamma \quad \text{on } \Gamma_\varphi^\gamma \quad (5.20)$$

$$\mathbf{t}^\gamma = \bar{\mathbf{t}}^\gamma \quad \text{on } \Gamma_\sigma^\gamma \quad (5.21)$$

where  $\bar{\boldsymbol{\varphi}}^\gamma$  and  $\bar{\mathbf{t}}^\gamma$  are prescribed quantities. When the bodies come into contact, the kinematic contact conditions (5.5) become the *Khun-Tucker* conditions (5.16). The new boundary conditions for contact are represented as:

$$g_N \geq 0, \quad p_N \leq 0, \quad g_N p_N = 0 \quad \text{on } \Gamma_c \quad (5.22)$$

The weak form of the local field equations for contact problems leads to a variational inequality due to the fact that the constraint condition (5.22) is represented by an inequality [40]. For the general case we can write

$$\sum_{\gamma=1}^2 \int_{\varphi(\mathcal{B}^\gamma)} \boldsymbol{\sigma}^\gamma \cdot \nabla^S (\boldsymbol{\eta}^\gamma - \boldsymbol{\varphi}^\gamma) dv \geq \sum_{\gamma=1}^2 \left( \int_{\varphi(\mathcal{B}^\gamma)} \rho^\gamma (\bar{\mathbf{b}}^\gamma \cdot (\boldsymbol{\eta}^\gamma - \boldsymbol{\varphi}^\gamma)) dv - \int_{\varphi(\partial\mathcal{B}_\sigma^\gamma)} \bar{\mathbf{t}} \cdot (\boldsymbol{\eta}^\gamma - \boldsymbol{\varphi}^\gamma) d\Gamma \right) \quad (5.23)$$

The integration is performed with respect to the domain  $\varphi(\mathcal{B}^\gamma)$  occupied by the body  $\mathcal{B}^\gamma$  in the current configuration. The deformation  $(\boldsymbol{\varphi}^1, \boldsymbol{\varphi}^2) \in \mathbf{H}$  is fulfilled for all  $(\boldsymbol{\eta}^1, \boldsymbol{\eta}^2) \in \mathbf{H}$  with

$$\mathbf{H} = \{(\boldsymbol{\eta}^1, \boldsymbol{\eta}^2) \in \mathbf{V} \mid [\boldsymbol{\eta}^2 - \hat{\boldsymbol{\eta}}^1(\bar{\boldsymbol{\xi}}^1, \bar{\boldsymbol{\xi}}^2)] \cdot \bar{\mathbf{n}}^1 \geq 0\} \quad (5.24)$$

where  $\mathbf{V}$  represents the space of test functions. Due to the inequality constraint on the deformation field, the contact problem is nonlinear even for linear elasticity. In consequence, special algorithms have to be designed.

The weak form of the contact boundary problem is presented by the variational inequality (5.23). This is different from the usual situation in solid mechanics, where the solution has to fulfil variational equations.

The variational inequality also characterizes the solution of the called *Signorini's* problem [117]. This problem defines the boundary value problem that includes the inequalities related to frictionless contact expressed in term of displacements for linear elasticity.

The variational inequality (5.23) is also stated for frictionless contact. The problem is even more complex when friction is present. In this case, the inequality constraints in the normal direction are involved as well as a special constitutive behavior in the tangent direction at the contact interface. This phenomenon is governed by sudden changes of the solution states, such that the solution jumps from a state of stick to another of sliding. These jumps are a handicap for the numerical scheme.

### 5.3.3 Frictional contact

The frictional response is important in most technical applications. Taking into account that we aim at modeling wear and excavation processes, friction is a phenomena that we must account for. The science dealing with the interfacial behavior related to the frictional response is called tribology [104]. Some tribology topics about friction are introduced in Section 6.2.

Friction can be defined as the resistance to motion arising from solids at the real contact surface area. Friction effects arise from the tangential forces transmitted across the contact interface when solid surfaces are pressed together by a normal force.

Friction has been investigated for a long time, starting with Leonardo da Vinci, Amon-ton [6], Coulomb [31] and Morin [79]. However many frictional phenomena are not completely understood up to-date. Scientists and engineers have typically assumed that friction is due to the interlocking of mechanical protuberances or asperities on the surfaces of

contacting materials. In this way is possible to explain why friction force is proportional to the load, and independent of the contact area. This is called the *roughness hypothesis* and is accepted if no adhesion forces exist. In general, friction depends on several parameters; the most relevant ones are the normal pressure, the relative tangential velocity, the surface roughness and the temperature.

In Section 5.2.2 the kinematics of the tangential contact were split in the *stick state* and the *sliding state*. The *stick state* is treated as a constraint, as no tangential relative displacement occurs in the contact zone. The *sliding state* is associated with a relative tangential movement in the contact interface. Frictional contact is modelled by appropriate constitutive laws. The most frequently used is the classical Coulomb law, presented in this section. However, other frictional laws which take into account local, micromechanical phenomena within the contact interface are available [128]. An overview of frictional contact models with relation to numerical modeling may be found in [90].

### Stick as a constraint

Stick is equivalent to the case in which the relative tangential velocity is zero. Hence, the stick condition can be obtained from (5.9) as

$$\dot{\mathbf{g}}_T = 0 \Leftrightarrow \mathbf{g}_T = 0 \quad (5.25)$$

This condition is formulated in the current configuration. In general it imposes a nonlinear constraint equation on the motion at the contact interface. Many possibilities exist to enforce the constraint condition (5.25). They depend upon the method used and will be discussed in Section 5.4.

### Coulomb law

*Coulomb law* describes the relative tangential kinetics of solids in contact. It is expressed as

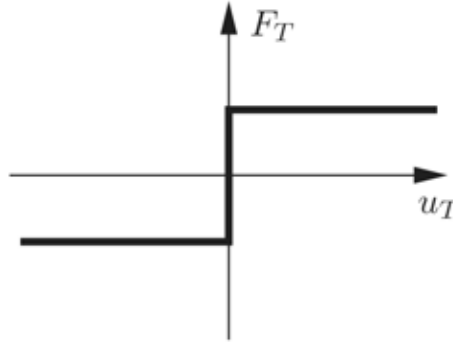
$$\mathbf{t}_T = -\mu |p_N| \frac{\dot{\mathbf{g}}_T}{\|\dot{\mathbf{g}}_T\|} \quad \text{if } \|\dot{\mathbf{t}}_T\| > \mu |p_N| \quad (5.26)$$

where  $\mu$  is the sliding friction coefficient. This coefficient is constant in the classical Coulomb law and it depends upon the material pairing of the solids in contact. Table (5.1) gives some values for different materials. The friction coefficient depends upon other different parameters such as the surface roughness, the relative sliding velocity  $\dot{\mathbf{g}}_T$  between the contacting bodies, the contact normal pressure  $p_N$  and the temperature  $\theta$ . If such effects are considered, a variant of Coulomb law with a variable coefficient  $\mu = \mu(\dot{\mathbf{g}}_T, p_N, \theta)$  is obtained.

Simplifying the dependencies of the friction coefficient and incorporating only the relative sliding velocity  $\dot{\mathbf{g}}_T$  in the expression, the friction coefficient is given by

$$\mu(\dot{\mathbf{g}}_T) = \mu_D + (\mu_S - \mu_D)e^{-c\|\dot{\mathbf{g}}_T\|} \quad (5.27)$$

where  $\mu_D$  and  $\mu_S$  are the dynamic and static friction coefficients. Figure 5.9 shows that the friction coefficient takes the static value  $\mu_S$  for zero sliding velocity. For large



**Figure 5.8:** *Coulomb* friction law (Zavarise et al., [134])

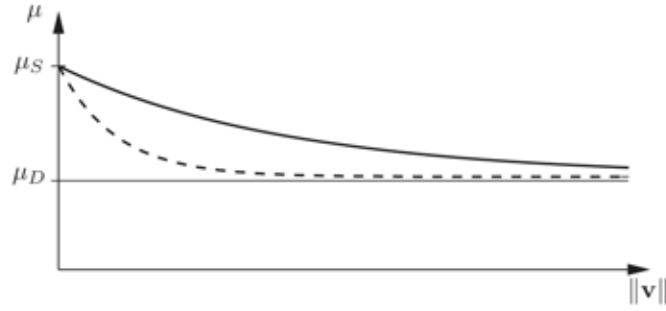
**Table 5.1:** Friction coefficient for different material pairings [134]

Material pairing	Friction coefficient $\mu$
concrete-concrete	0.5-1.0
concrete-sand	0.35-0.6
concrete-steel	0.2-0.4
metal-wood	0.3-0.65
rubber-steel	0.15-0.65
steel-steel	0.2-0.8
steel-teflon	0.04-0.06
steel-ice	0.015-0.03
wood-steel	0.5-1.2
wood-wood	0.4-1.0

velocities  $\mu$  converges to the dynamic friction coefficient  $\mu_D$ . The constitutive parameter  $c$  describes how fast the static coefficient approaches the dynamic one.

If the sliding velocity is neither too large nor too small, then the friction force is proportional to the normal force, leading to a constant friction coefficient  $\mu$ . In this case is not necessary to apply (5.27).

Figure 5.8 shows the non-differentiability of *Coulomb* law at the onset of sliding. This non-differentiability is a problem for the numerical analysis of the tangential constitutive equation. In order to avoid this problem a regularization of the stick-slip behavior is performed.



**Figure 5.9:** Velocity-dependent *Coulomb* friction law [134]

### Regularization of Coulomb law

To formulate the tangential constitutive equations in the contact interface a regularization of the stick-slip behavior of *Coulomb* law is performed. This is based on a functional form which yields a smooth transition from stick to slip, as depicted in Figure 5.10. Associated models were introduced in [91] and can be found also in [106].

The explicit forms are given for two-dimensional problems by

$$\mathbf{t}_T = -\mu \varphi^i(\dot{\mathbf{g}}_T) |p_N| \quad (5.28)$$

where function  $\varphi^i$  describes the smooth transition from sticking to sliding and is defined by either

1. a square root regularization

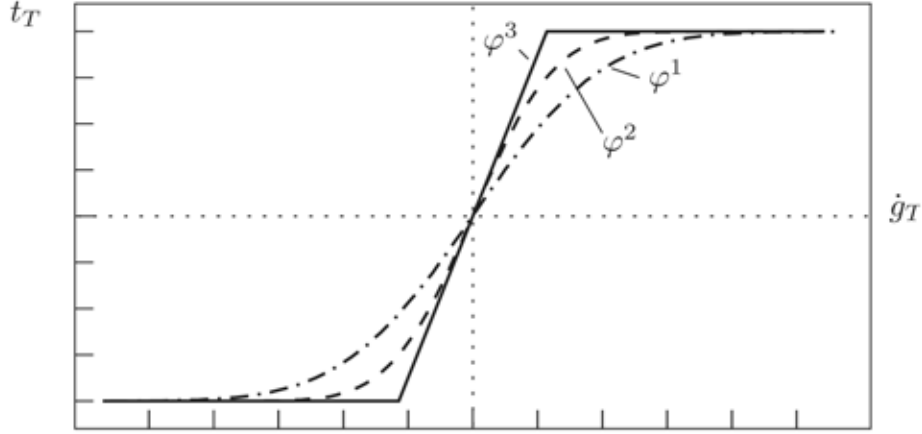
$$\varphi^1 = \frac{\|\dot{\mathbf{g}}_T\|}{\sqrt{\|\dot{\mathbf{g}}_T\|^2 + \varepsilon^2}} \quad (5.29)$$

2. a hyperbolic tangent regularization

$$\varphi^2 = \tanh \frac{\|\dot{\mathbf{g}}_T\|}{\varepsilon} \quad (5.30)$$

3. or a piecewise polynomial regularization

$$\varphi^3 = \begin{cases} -1 & \text{if } \dot{\mathbf{g}}_T < -\varepsilon \\ \frac{\dot{\mathbf{g}}_T}{2\varepsilon} & \text{if } -\varepsilon \leq \dot{\mathbf{g}}_T \leq \varepsilon \\ 1 & \text{if } \dot{\mathbf{g}}_T > \varepsilon \end{cases} \quad (5.31)$$



**Figure 5.10:** Regularization of *Coulomb* friction law [134]

The scalar parameter  $\varepsilon$  denotes the regularization variable. Note that the limit case  $\varepsilon \rightarrow 0$  yields the classical *Coulomb* law.

The presented models lead to numerical algorithms which are simpler and robust. These regularized laws physically have the drawback that they describe the stick-slip motion only in an approximate way. It is important to take in account that for a too large value of the parameter  $\varepsilon$ , such models might not be able to predict real stick-slip motions. The regularization introduced by the functions  $\varphi^i$  is depicted in Figure 5.10.

#### 5.4 Treatment of the contact constraints

There are several alternatives to incorporate the contact constraints into the variational formulation for frictionless and for frictional contact. The methods are designed to fulfil the constraint equations on the normal direction in the contact interface. For the tangential part, constitutive relations are used when stick-slip motion occurs.

The most simple representation assumes that the contact interface is known. This is often the case and is called *active set* strategy. It means that the active set of constraints are known within an incremental solution step. This assumption is employed to solve the inequality (5.23). Once the contact interface is known, the weak form can be written as an equality. Hence, equation (5.23) can be written as

$$\sum_{\gamma=1}^2 \left\{ \int_{\varphi(B^\gamma)} \boldsymbol{\sigma}^\gamma \cdot \nabla^S \boldsymbol{\eta}^\gamma \, dv - \int_{\varphi(B^\gamma)} \rho^\gamma \bar{\mathbf{b}}^\gamma \cdot \boldsymbol{\eta}^\gamma \, dv - \int_{\varphi(\partial B_\sigma^\gamma)} \bar{\mathbf{t}} \cdot \boldsymbol{\eta}^\gamma \, d\Gamma \right\} + C_c = 0 \quad (5.32)$$

where  $C_c$  are contact contributions associated with the active constraint set.  $\boldsymbol{\eta}^\gamma \in V$  is the test function or virtual displacement, which is zero at the boundary  $\Gamma_\sigma^\gamma$ , where the

deformations are prescribed. As explained before, the formulation assumes that contact is active at the surface  $\Gamma_c \equiv \partial\mathcal{B}_c$ .

A large variety of formulations exist for the constraint term  $C_c$ . The most known popular ones are listed next.

1. The LAGRANGE MULTIPLIER method.
2. The PENALTY method.
3. The method of direct elimination (NODE to NODE).
4. The Perturbed Lagrange formulation (which combines (1) and (2) in a mixed form).
5. The Augmented Lagrange method.
6. The Nitsche method (which enforces geometrical constraints in a weak sense).
7. The Barrier method.
8. The Cross-Constraint method (which combines the penalty and barrier methods).

All these formulations and their associated algorithms can be found in [132] and [134]. For the penalty techniques applied to solve contact problems see the text books [14] and [75]. There are other solution schemes, based on mathematical programming, which are applicable to frictional contact problems, (see [28] and [64]).

Most standard finite element codes which are able to handle contact situations use either the *Penalty* or the *Lagrange Multiplier* method. An overview and their mathematical framework can be seen in [63].

Each of the methods has its own advantages and disadvantages. In contact mechanics, the algorithmic treatment is one of the major problems after the formulation of the contact inequality. This is associated with the numerical analysis of the contact constraints. Some difficulties must be overcome and the most remarkable are the non-differentiability of the normal contact and the treatment of the friction terms. The existence of a large variety of formulations also reflects that there are many different algorithms which had been developed in order to solve contact problems.

The *Lagrange Multiplier* method and the *Penalty* method are presented in the next sections. After that, the formulation developed for the PFEM will be explained.

#### 5.4.1 Lagrange multiplier method

A classical approach is using Lagrange multipliers to add constraints to the weak form. The contact contribution can be formulated in terms of the contact potential as:

$$\Pi_c^{LM} = \int_{\varphi(\partial\mathcal{B}_c)} (\lambda_N g_N + \boldsymbol{\lambda}_T \cdot \mathbf{g}_T) d\Gamma \quad (5.33)$$

where  $\lambda_N$  and  $\boldsymbol{\lambda}_T$  are the Lagrange multipliers.  $g_N$  and  $\mathbf{g}_T$  are the normal and tangential gap functions. The variation of  $\Pi_c^{LM}$  leads to the constraint formulation (5.32):

$$C_c^{LM} = \int_{\boldsymbol{\varphi}(\partial\mathcal{B}_c)} (\lambda_N \delta g_N + \boldsymbol{\lambda}_T \cdot \delta \mathbf{g}_T) d\Gamma + \int_{\boldsymbol{\varphi}(\partial\mathcal{B}_c)} (\delta \lambda_N g_N + \delta \boldsymbol{\lambda}_T \cdot \mathbf{g}_T) d\Gamma \quad (5.34)$$

Note that the Lagrange multiplier  $\lambda_N$  can be identified as the contact pressure  $p_N$ .

In the case of pure stick, the relative tangential slip  $\mathbf{g}_T$  is zero, which yields a constraint equation from which  $\boldsymbol{\lambda}_T$  is identified as the reaction.

In the case of sliding, a tangential stress vector  $\mathbf{t}_T$  is determined from the constitutive law for frictional slip (see Section 5.3.3) and it will be used instead of  $\boldsymbol{\lambda}_T$ , leading to

$$C_c^{LM \text{ slip}} = \int_{\boldsymbol{\varphi}(\partial\mathcal{B}_c)} (\lambda_N \delta g_N + \mathbf{t}_T \cdot \delta \mathbf{g}_T) d\Gamma + \int_{\boldsymbol{\varphi}(\partial\mathcal{B}_c)} (\delta \lambda_N g_N) d\Gamma \quad (5.35)$$

The variation of the tangential gap  $\mathbf{g}_T$  and the normal gap  $g_N$  can be found in Section 5.2.

#### 5.4.2 Penalty method

The penalty term due to the constraint condition (5.6) is formulated in terms of a contact potential as:

$$\Pi_c^P = \frac{1}{2} \int_{\boldsymbol{\varphi}(\partial\mathcal{B}_c)} (\epsilon_N (\bar{g}_N)^2 + \epsilon_T \mathbf{g}_T \cdot \mathbf{g}_T) d\Gamma, \quad \text{with } \epsilon_N, \epsilon_T > 0 \quad (5.36)$$

where  $\epsilon_N$  and  $\epsilon_T$  are the penalty parameters. The variation of (5.36) yields

$$C_c^P = \frac{1}{2} \int_{\boldsymbol{\varphi}(\partial\mathcal{B}_c)} (\epsilon_N \bar{g}_N \delta \bar{g}_N + \epsilon_T \mathbf{g}_T \cdot \delta \mathbf{g}_T) d\Gamma, \quad \text{with } \epsilon_N, \epsilon_T > 0 \quad (5.37)$$

As in the Lagrange multiplier method, the slip condition leads to

$$C_c^{P \text{ slip}} = \frac{1}{2} \int_{\boldsymbol{\varphi}(\partial\mathcal{B}_c)} (\epsilon_N \bar{g}_N \delta \bar{g}_N + \mathbf{t}_T \cdot \delta \mathbf{g}_T) d\Gamma, \quad \text{with } \epsilon_N, \epsilon_T > 0 \quad (5.38)$$

It can be shown that the *Lagrange Multiplier* method is recovered from this formulation for  $\epsilon_N \rightarrow \infty$  and  $\epsilon_T \rightarrow \infty$ . However, large numbers for  $\epsilon_N$  and  $\epsilon_T$  turn the numerical problem to be ill-conditioned.

#### 5.4.3 Continuum constraint method

For the PFEM a new formulation for the contact contribution has been developed. It is based in the interface mesh created in the contact detection (Section 4.5) and it has been named as the *Continuum Constraint Method*. The contact constraint for the present formulation is founded on the potential of a fictitious continuum domain that works between the contacting bodies  $\mathcal{B}^c$ . When the constraint is active on  $\partial\mathcal{B}^c$ , the contact term in the functional (5.32) is related to the continuum and is similar to the internal weak term.



Some particular considerations for defining the contact gap are applied to fulfill the contact kinematics (see Section 5.2.3). Considering equation (5.17) the contact constraint can be written as:

$$C_c^{CC} = \int_{\varphi(\mathcal{B}^c)} (\mathbb{P} \boldsymbol{\sigma}_c + \mathbb{T} \boldsymbol{\sigma}_c) \cdot \nabla^S \boldsymbol{\eta}_c \, dv \quad (5.39)$$

with the projection tensor  $\mathbb{P} = \bar{\mathbf{n}}^1 \otimes \bar{\mathbf{n}}^1$  and  $\bar{\mathbf{n}}^1$  is the normal to the contact surface at the minimum distance point. Tensor  $\mathbb{T}$  can be written for the stick case as  $\mathbb{T} = (\mathbb{I} - \mathbb{P})$ .

The integration is performed with respect to the continuum domain  $\varphi(\mathcal{B}_\sigma^c)$  instead of the contact surface  $\partial\mathcal{B}_c \equiv \Gamma_c$ . Note that  $\varphi(\mathcal{B}_\sigma^c)$  is the sub-domain occupied by the interface contact media  $\mathcal{B}_\sigma^c$  in the current configuration. For simplicity  $\mathcal{B}^c \equiv \mathcal{B}_\sigma^c$ . The test functions  $\boldsymbol{\eta}_c^\gamma$  of equation (5.39) are the virtual valued functions for the contact interface  $\mathcal{B}_\sigma^c$  equivalent to the ones of equation (5.32); i.e.  $\boldsymbol{\eta}_c \equiv \boldsymbol{\eta}_c^\gamma$ .

The deformation experienced at the interface is fulfilled for all  $(\boldsymbol{\eta}_c^1, \boldsymbol{\eta}_c^2) \in \mathbf{H}$  for  $\gamma = 1, 2$ , such that

$$\mathbf{H} = \{(\boldsymbol{\eta}_c^1, \boldsymbol{\eta}_c^2) \in \mathbf{V} \mid [\boldsymbol{\eta}_c^2 - \hat{\boldsymbol{\eta}}_c^1(\bar{\boldsymbol{\xi}}^1, \bar{\boldsymbol{\xi}}^2)] \cdot \bar{\mathbf{n}}^1 \geq 0\} \quad (5.40)$$

where  $\mathbf{V}$  is the space of test functions; see Section 5.3.2. This is the same to say that the test functions must satisfy the non-penetration condition (5.1) established in the kinematic conditions for the normal contact.

Tensor  $\boldsymbol{\sigma}_c$  in equation (5.39) is the *Cauchy* stress tensor for the contact domain and can be written as:

$$\boldsymbol{\sigma}_c = \mathbb{P} \boldsymbol{\sigma}_c + (\mathbb{I} - \mathbb{P}) \boldsymbol{\sigma}_c = \mathbb{P} \boldsymbol{\sigma}_c + \mathbb{T} \boldsymbol{\sigma}_c \quad \text{with} \quad \mathbb{T} = \mathbb{I} - \mathbb{P} \quad (5.41)$$

Using the last definition of  $\boldsymbol{\sigma}_c$ , the expression (5.39) becomes

$$C_c^{CC} = \int_{\varphi(\mathcal{B}^c)} \boldsymbol{\sigma}_c \cdot \nabla^S \boldsymbol{\eta}_c \, dv \quad (5.42)$$

There is an analogy between equations (5.42) and (2.210) that represents the virtual internal work for a continuum domain.

The gap function for this formulation is defined in (5.14); see Section 5.2.3. The Kuhn-Tucker conditions (5.16) are fulfilled using these new definitions. The constitutive law used to compute  $\boldsymbol{\sigma}_c$  is similar to a penalty law that depends on  $\delta\bar{g}_N$  and  $\delta\mathbf{g}_T$ . The limit for this penalization is extended in the context of material incompressibility. For this case, the mixed formulations for the continuum are comparable to the classical formulations for contact [138], [88].

The penalty constraint affects the normal and tangential directions. For the slip condition the constraint in the tangential direction changes. When this happens, frictional laws are applied to define the tangential behavior. The slip condition leads to the equation.

$$C_c^{CC} = \int_{\varphi(\mathcal{B}^c)} (\mathbb{P} \boldsymbol{\sigma}_c + \boldsymbol{\kappa}_c) \cdot \nabla^S \boldsymbol{\eta}_c \quad (5.43)$$

where  $\boldsymbol{\kappa}_c$  is denoted the tangential stress tensor for contact. Using a classical frictional law, e.g. *Coulomb* law,  $\boldsymbol{\kappa}_c$  can be written as:

$$\boldsymbol{\kappa}_c = -\mu \mathbb{T} \boldsymbol{\sigma}_c \quad (5.44)$$

where  $\mu$  is the sliding friction coefficient of (5.27) and  $\mathbb{T} = \bar{\mathbf{t}}^1 \otimes \bar{\mathbf{n}}^1$ , being  $\bar{\mathbf{n}}^1$  and  $\bar{\mathbf{t}}^1$  the normal and the tangent vectors defined on the contact surface respectively. In this case (5.43) yields

$$C_c^{CC} = \int_{\boldsymbol{\varphi}(B^c)} (\mathbb{P} \boldsymbol{\sigma}_c - \mu \mathbb{T} \boldsymbol{\sigma}_c) \cdot \nabla^S \boldsymbol{\eta}_c \quad (5.45)$$

## 5.5 Contact search

The geometrical detection of contact is a complex problem in computational contact mechanics. It has been previously introduced in Section 4.5 but it will be extended in this chapter. The geometric search is especially complex when the contact of more than two bodies has to be considered. That can happen when the problem is such that the solids can break, and hence several discrete elements are originated from the initial set up during the solution process. This is common in dynamic and impact analysis.

Static contact problems with large deformations also need fast and reliable search algorithms, as contact areas are not known a priori and can change much within a load step. The search for an active set of contact constraints is not trivial in this case, since a surface point of a body may contact any portion of the surface of another body. Thus the search for the correct contact detection eventually needs considerable effort, depending on the problem. This is especially true when self-contact is possible.

Classically, the contact search task is split into two phases: the spatial search for objects (finite element subdomains) which might possibly come into contact, and the determination of pairs of objects (finite element subdomains) which actually intersect and hence are in contact.

Usually in the first phase the finite elements lying on the surface of the solid are ordered by a sorting algorithm. After that, a hierarchical structure is set up to find out which bodies, part of the bodies, surfaces or parts of the surfaces are able to come into contact. This a global search within a given time step or displacement increment. Search algorithms combine complex structures of cell decompositions, binary tree searches or more advanced spatial methods. Some general contact search algorithms can be found in [132].

The complicated phases in the spatial search are really simplified in the PFEM as the spatial search is intrinsically done at the mesh generation level. Contact is detected when the domains are so close as to generate contact elements between the interacting subdomains. That was introduced in Section 4.5 and can be seen in Figure 4.15. An interface mesh is created anticipating the spatial contact. Hence, a mesh for the domain and another one for the interface are generated. That permits to anticipate the collision of the different subdomains. In order to include self-contact in the spatial search new tools have to be added to the contact search algorithm.

### 5.5.1 Self-contact in the PFEM

When a point can come into contact with a part of the surface of its own body, the search of the correct contact location becomes more complex. In the PFEM the interface mesh is created between different subdomains. When self-contact has to be included other particle variables have to be evaluated.

The algorithm for detecting self-contact elements are founded in the evaluation of the surface normals. Let us consider that every particle that belongs to the surface of a subdomain has an associated inward normal vector which points towards the inner side. After applying the *Alpha-shape* criteria, a self-contact element will fulfill the conditions of Box 5.1.

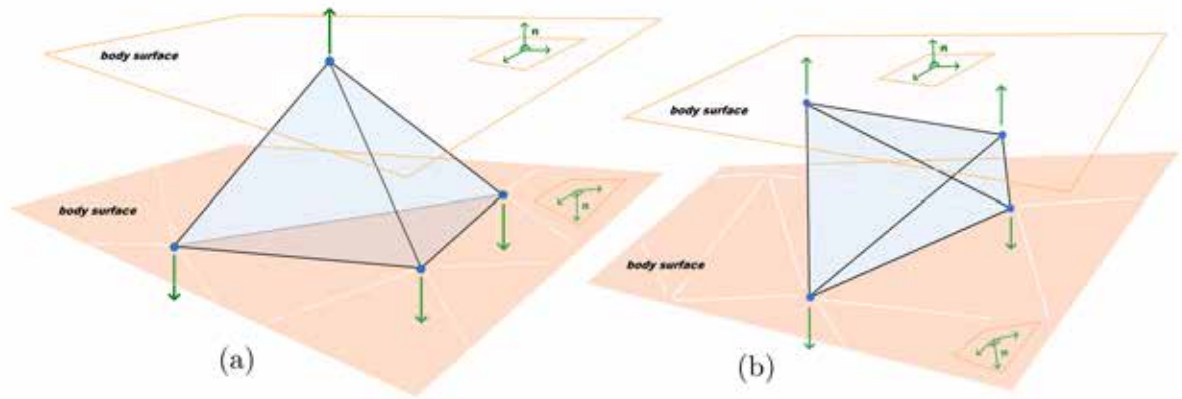
1. All contact element nodes are particles of the surface boundary
2. There is a prior definition of the particles of the domain connected to each element node before the contact search. Using these connections a contact element must fulfill:
  - In 2D: One node of the element has connection with others.
  - In 3D: One or two element nodes have no connection with others. This defines two different contact element types, *Node to face* element and *edge to edge* element. See Figure 5.11.
3. All normals associated to element nodes point towards the outside of the element:

The vector from the center of the element to each node is computed and projected on the normal of the corresponding node. If the projection is negative the condition is fulfilled.

Box 5.1: Conditions to detect a self-contact element

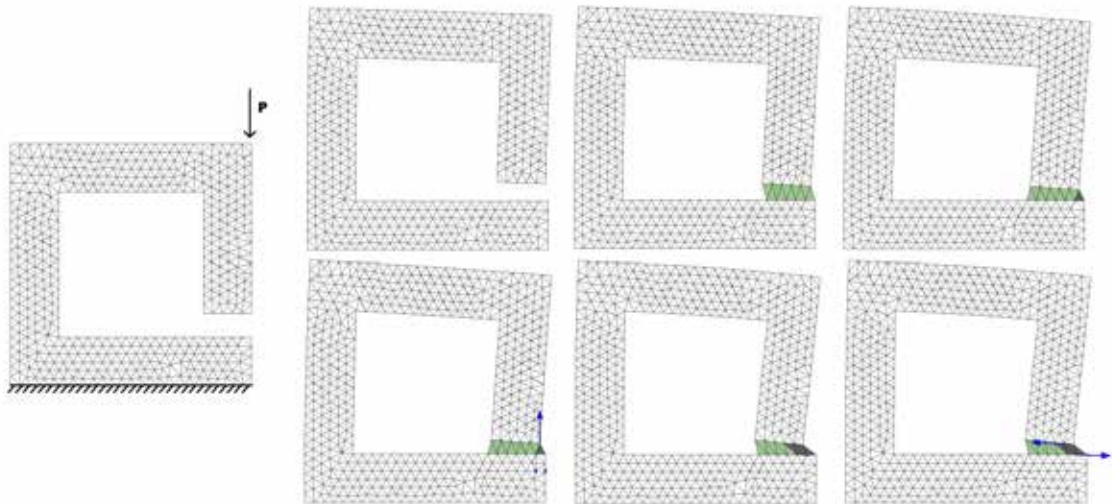
These conditions are illustrated in Figure 5.11.

The first condition simplifies the number of elements to be analyzed. Only elements on the edges and self contacting elements will be formed entirely by particles on the boundary of the same subdomain. The second condition rejects contact elements that are formed in the convex edges of the subdomain. The second condition identifies the position of the element. Hence, if the element is located outside the subdomain, then is a self-contact element.



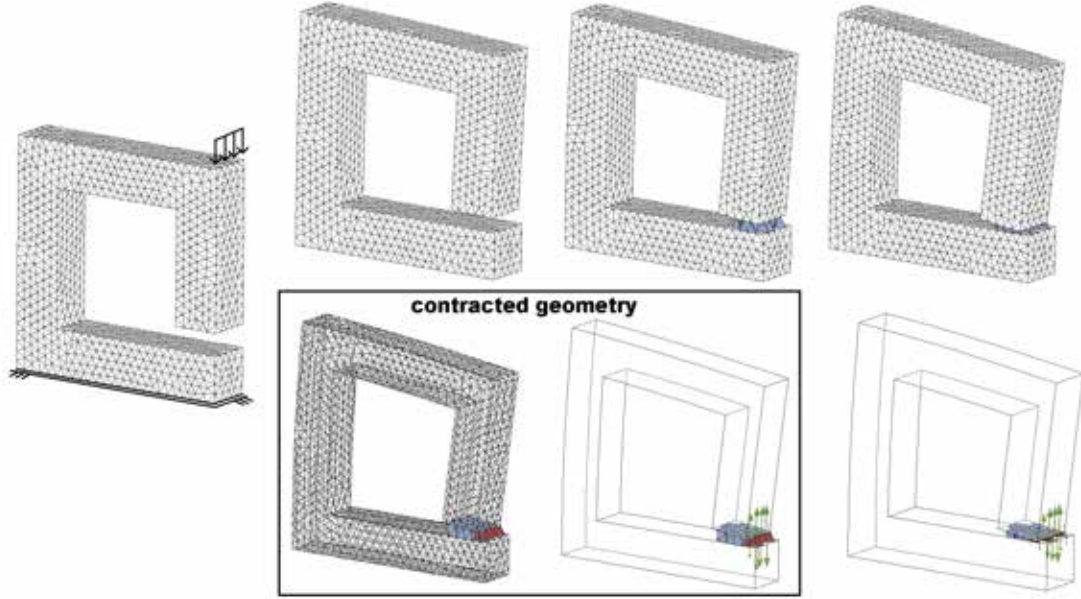
**Figure 5.11:** Self-contact elements in 3D formed between two exterior surfaces of the domain: (a) *Node to face* element, (b) *Edge to edge* element.

The self-contact search is not possible using just the  $\alpha$ -shapes. To perform the detection of self-contact the contour has to be recognized and the boundary normals have to be computed. Note that these requirements are the same as in the refinement of the *alpha*-shapes (Section 4.4.2). Therefore, the contact search for self contact is only an additional feature to the refining algorithm explained in Box 4.1.



**Figure 5.12:** Detection of self-contact in a two dimensional example

Figures 5.12 and 5.13 show the detection of self-contact in a two-dimensional and a three-dimensional geometry, respectively.



**Figure 5.13:** Detection of self-contact in a three dimensional example. The offset algorithm is applied to get the exact contact geometry

### 5.5.2 Contact detection

After the global search is completed, neighboring finite element meshes are joined by a layer of contact elements. These elements anticipate the interaction between subdomains and define the possible contact. The detection of active elements in the physical contact is performed a posteriori.

The detection algorithm of Box 5.2 is performed for each contact element.

The set of contact elements define the surface area in contact. The contact distance  $\hat{g}_N$  is defined by the element size and will be determined when the first layer of active contact elements are generated. The criteria to determine  $\hat{g}_N$  starts with the determination of the first active element. This is done in function of the non-uniform  $\alpha$ -shape parameters (4.1) associated to that element (see Section 4.4.1):

$$d_{act} = \frac{1}{n} \sum_{i=1}^n \alpha_i \quad (5.46)$$

where  $d_{act}$  is the normal distance associated to the first active element,  $n$  is the number of nodes of the contact element and  $\alpha_i$  the alpha-shape parameter associated to each node. The contact activation is defined by:

$$\begin{cases} d_i^n \leq d_{act} & \Rightarrow \hat{g}_N = d_i^n & \text{active contact} \\ d_i^n > d_{act} & \Rightarrow \hat{g}_N = 0 & \text{no contact} \end{cases} \quad (5.47)$$

For every contact element:

1. Compute the normal to the face residing in the body surface for *Node to face* elements .  
If the contact element has no face in the surface (*Edge to edge* elements), then one of the normals of the body surface is considered.
2. Check the normal distance  $d_i^n$  between the contacting surfaces. This is the normal distance from element-node to element-face.
3. When the computed distance  $d_i^n$  is closer than the *contact distance*  $\hat{g}_N$ , the contact element is considered to be active.

Box 5.2: Contact algorithm for the detection of active elements

The offset parameter for the contraction is determined by

$$d_c = \frac{\hat{g}_N}{2} \quad (5.48)$$

(Section 4.5.2). The contact distance  $\hat{g}_N$  remains constant while the *active set* does not change.

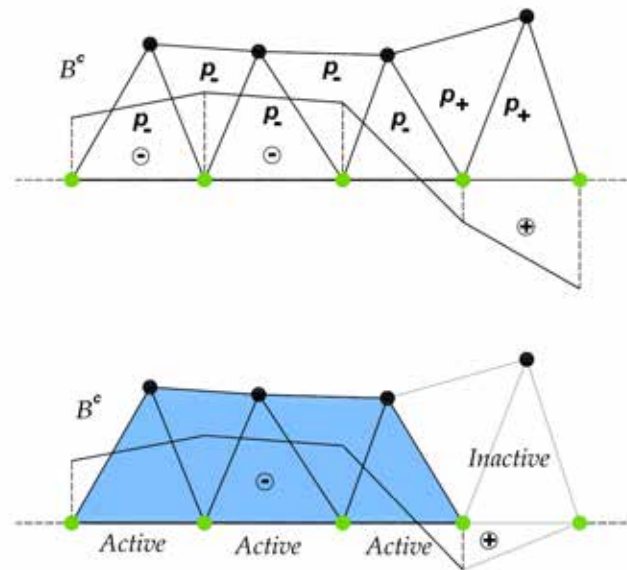
It is important to note that the geometrical contact conditions have to be formulated in the current configuration. The contact detection is checked at the end of the analysis to prove that the contact configuration is correct.

### 5.5.3 Enforcement of the contact constraints

The *Continuum Constraint Method* gives contact stresses that are proportional to the normal deformation of the contact elements. Hence, a local enforcement of the contact constraints in contact elements is reasonable. When single integration points (linear contact elements) are in tension, they are enforced not to contribute to the weak form of contact in the next iteration. Only compressive contact elements of the *active set* are considered. In the PFEM linear triangles and tetrahedra are used for the contact domain. This allows us to define a simple criteria to activate and deactivate contact elements. In case of having multiple integration points, an average enforcement is the best choice. The contact element will remain active as long as the average stresses at the integration points are compressive. The element-wise enforcement for the contact constraint method is presented in Figure 5.14.

With the element-wise enforcement, compressions on contact elements are ensured, but this does not ensures compression forces for all points on the contact surfaces. Tensile contact forces can appear locally due to deviated geometrical configurations of the contact elements (see Figure 5.15).

The assembly of the contribution of the entire active set results in compressive global forces for the contact surface points. This makes the weak enforcement of the contact



**Figure 5.14:** Activation and de-activation of contact elements and pressure distribution arising from a linear approximation.

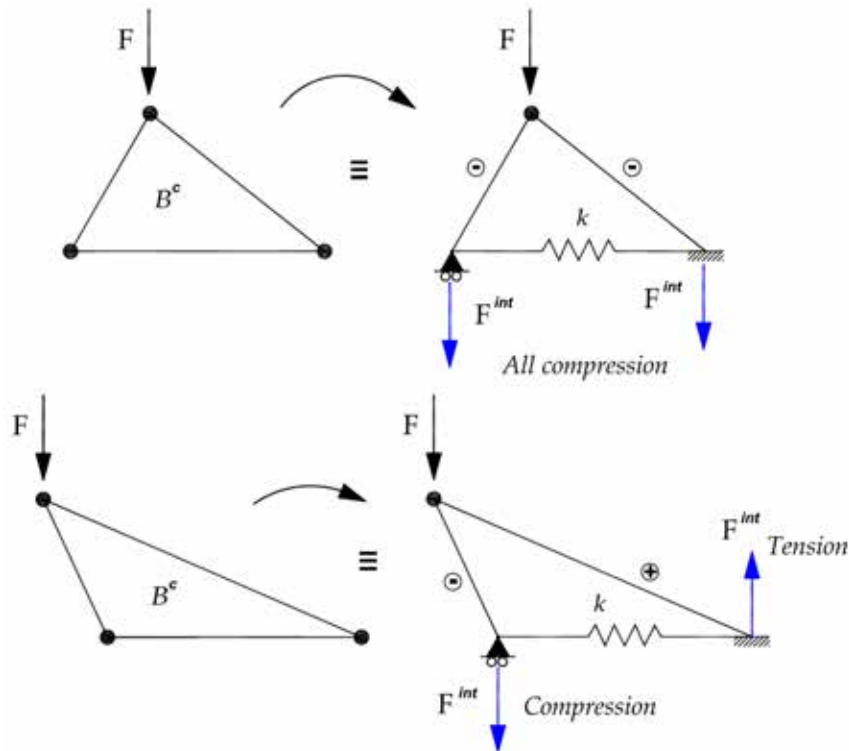
constraints consistent with the global contact forces. However, there is a transition tension-compression region in the end boundaries of the contact domain when the geometry of the contact elements is too inclined (see Figure 5.15). The solution to avoid any tensile forces is to ensure a more regular geometry of the contact element in the boundaries of the contact domain.

## 5.6 Contact mechanics in the PFEM

The penalty method or the Lagrange multiplier method can be used to treat the interaction between the deformable bodies in contact [132]. These two methods are used in the PFEM and they have the advantage that the detection of the contact regions is performed in an easy manner via the alpha-shape contact search. The size of the system remains the same when a penalty method is used as there are not additional degrees of freedom in the system. Even so the characterization of the penalty parameters and the convergence of the numerical solution are two difficulties that remain and preclude obtaining a good result.

Contact implies compressive and friction forces between the domains under interaction as defined in previous sections. The magnitude of these forces is large for moderate time steps. Big values for the contact forces change the condition of the numerical system adversely and push the analysis towards using small time steps. This jeopardizes the advantages of using an implicit integration scheme.

These problems can be overcome by treating the contact interface as a part of the continuum via the contact element and applying the adaptive tools of the PFEM. We note that elements laying in the interface between the interacting sub-domains are not used to define a spring or to assign direct correspondence in the relative movement of the



**Figure 5.15:** Geometrical configuration of a single element, irregular inclined elements can produce tensile forces.

contact domains. *These elements are in fact considered as finite elements with special contact properties.*

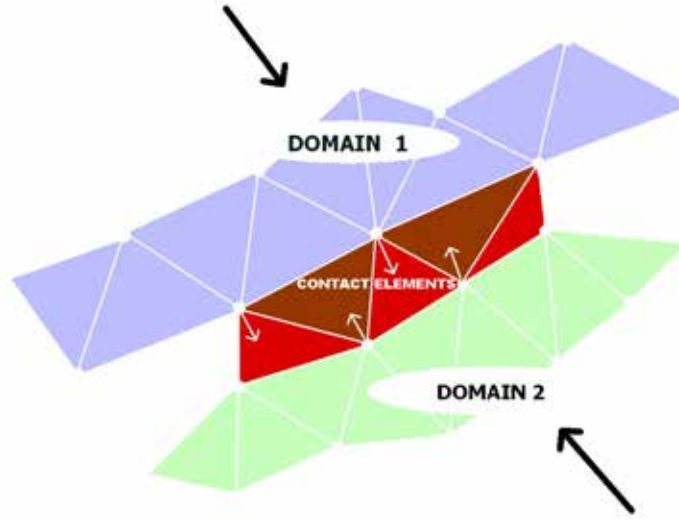
### 5.6.1 Discretization of the contact

The discretization of the contact is necessary for the numerical computation of the contact constraint. The contact element is the basic structure for the geometric detection and for the contact constraint implementation. There are several methodologies to discretize the contact. The classical ones are *Node-to-Face* and *Node-to-Node* formulations [132], [134]. There are other discretizations for non-matching meshes that look for a stabler scheme. An example is the *Mortar* method, based on *Lagrange Multipliers* or the *Nitsche* method [132].

For the PFEM, the discretization of contact does not require a special contact element for the contact constraints. The same finite elements used for discretizing the continuum (e.g. triangles or tetrahedra) are used for the contact discretization. This is an advantage of the PFEM, which provides the simplest discretization for contact.

The discrete equations for the *Continuum Constraint Method* (Section 5.4.3) are presented in the next sections for the triangle and the tetrahedron. A layer of these elements defines the *active set* when the contact search of the PFEM is employed (Figure 5.16).





**Figure 5.16:** Mesh of contact elements between two domains discretized with triangles

### 5.6.2 Computation of normal contact forces

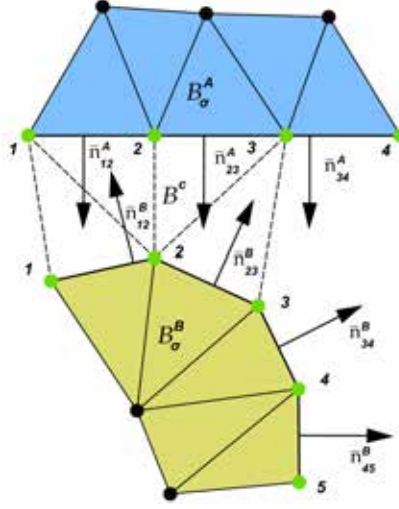
The expression of the contact constraint (5.42) is analogous to the virtual work for the internal forces of a finite element. This was presented in Section 5.4.3. The virtual work that represents the contact constraint defines the contact nodal forces for the contact elements. In fact, contact elements do not represent any continuum. Therefore, the stress tensor defined for the contact elements  $\sigma_c$  is the result of a constitutive law that enforces the contact constraint.

First the displacement field for the contact elements is projected in the direction of the normal to the contact faces only. This gives:

$$(\mathbf{u} \cdot \bar{\mathbf{n}}^1) > 0 \rightarrow \mathbf{u}^n = (\mathbf{u} \cdot \bar{\mathbf{n}}^1) \cdot \bar{\mathbf{n}}^1 = \mathbb{P} \mathbf{u} \quad (5.49)$$

where  $\mathbf{u}$  is the displacement of the nodes belonging to the contact elements,  $\mathbb{P} = \bar{\mathbf{n}}^1 \otimes \bar{\mathbf{n}}^1$  and  $\bar{\mathbf{n}}^1$  is the unit normal to the contact surface defined in the projected point into the other surface. This point is the node of the contact element that has no face on the contact surface. The normal is defined by the face of the second body. This definition introduces a discontinuity in the normals between elements. It is depicted in Figure 5.17.

The constitutive laws of the bodies in contact describe the material behavior and do not affect the main formulation of the contact problem. Indeed these laws can be arbitrary. This also means that the constitutive law for the contact constraint (required in some formulations) is independent of the material behavior of the bodies coming into contact. This is partially true when the contact constraint is modelled by penalty parameters or by constitutive laws that affect the contact behavior. As well as in friction laws the material pairings define the parameters of the frictional behavior. In standard contact models the contact parameters are defined in terms of the properties of the interacting domains, but



**Figure 5.17:** Definition of the normal for each element face. The normal to the surface  $\partial B_\sigma^B$  changes from element to element in the contact domain  $B^c$ .

not from their constitutive law.

The constitutive properties for contact elements are computed as a combination of the properties of the interacting domains. The elastic variables are the more representative ones. The stiffness of the interaction is governed by the elasticity modulus defined for the contact elements. For a two-body contact, the *Young* modulus for the contact element  $E_{B^c}$  is computed by

$$E_{B^c} = \epsilon_N \left( \frac{1}{2} \sum_{\gamma=1}^2 E_{B_\sigma^\gamma} \right) \quad (5.50)$$

where  $\epsilon_N$  is a penalty parameter which ensures that the stiffness of the contact elements is larger than that of the contacting bodies. Contact elements are also characterized for having zero density,  $\delta_c = 0$  and zero *Poisson* ratio,  $\nu_c = 0$ . The  $\sigma_c$  value is computed using a non-linear elastic law (2.90) or (2.91) and the volumetric part of the *Cauchy* stress tensor only ( $\sigma = -p \mathbf{1} + \mathbf{s}$ ). This yields

$$\sigma_c := \sigma_c^{vol} = \left( \frac{K}{J^n} \ln J^n \right) \mathbf{1} \quad (5.51)$$

where  $K$  is the bulk modulus  $K = \lambda + \frac{2\mu}{3}$  and  $J^n$  is the *Jacobi* determinant (2.11) computed using the projected displacements (5.49). The normal pressure can be written as

$$p_N = - \left( \frac{K}{J^n} \ln J^n \right) \quad (5.52)$$

Applying the *Khun-Tucker* conditions (5.16), and the definition of the gap function for the PFEM (5.14) the constitutive law leads to

$$\boldsymbol{\sigma}_c = \begin{cases} \left(\frac{K}{J^n} \ln J^n\right) \mathbf{1} & \rightarrow \text{if } (\bar{g}_N = \text{true} \text{ and } p_N < 0) \\ \mathbf{0} & \rightarrow \text{if } (\bar{g}_N = \text{false} \text{ or } p_N \geq 0) \end{cases} \quad (5.53)$$

This methodology can be viewed as a special nonlinear constitutive law for contact elements which takes into account volumetric changes of the contact elements (5.51) only, produced by the projected displacements on the normal to the contact interface (see equation (5.49)).

### The penalty parameter

The constitutive law for contact depends on an equivalent *Young* modulus  $E(\epsilon_N)$  and a penalization parameter  $\epsilon_N$ , (see equation (5.50)). This penalization parameter ensures that the stiffness of the contact elements is larger than the stiffness of the contacting bodies. This is important for predicting the correct contact forces. This parameter is, at the same time, a damping function for numerical instabilities which take place in dynamic contact. Therefore  $\epsilon_N$  is a function of the relative velocity of the contacting bodies  $v_r$ .  $\epsilon_N = \epsilon_N(v_r)$  is defined as follows for two colliding bodies:

$$\epsilon_N = \begin{cases} \left(\frac{\max\{E_{\mathcal{B}_\sigma^\gamma}\}}{E_{\mathcal{B}^c}}\right) & \text{if } v_r \geq \left(\frac{l_c}{\Delta t}\right) \\ \left(\frac{1}{v_r}\right) \cdot \left(\frac{\max\{E_{\mathcal{B}_\sigma^\gamma}\}}{E_{\mathcal{B}^c}}\right) & \text{if } 10^{-3} \cdot \left(\frac{l_c}{\Delta t}\right) \leq v_r < \left(\frac{l_c}{\Delta t}\right) \\ 10^{-3} \cdot \left(\frac{l_c}{\Delta t}\right) \cdot \left(\frac{\max\{E_{\mathcal{B}_\sigma^\gamma}\}}{E_{\mathcal{B}^c}}\right) & \text{if } v_r < 10^{-3} \cdot \left(\frac{l_c}{\Delta t}\right) \end{cases} \quad 0 < \gamma < 2 \quad (5.54)$$

where  $l_c$  is the characteristic length of the contact elements and  $\Delta t$  is the time step used in the analysis. Parameters  $E_{\mathcal{B}_\sigma^\gamma}$  is the *Young* modulus of the domains in contact and  $E_{\mathcal{B}^c}$  is the *Young* modulus of the contact elements. This definition of the penalty parameter introduces a new non-linearity in the contact constraint. This however gives a better performance in the analysis of dynamic contact problems. Examples are presented in Section 5.8.2 and Section 5.8.3.

For the static case  $\epsilon_N$  can be taken as constant if the next condition is satisfied

$$E_{\mathcal{B}^c} \geq \max\{E_{\mathcal{B}_\sigma^\gamma}\} \quad (5.55)$$

In some cases one can ignore this condition. The implication of this is a loss of accuracy in the prediction of the contact force and, even more, it could destroy the physical response. However, the positive consequence, is that it softens the contact constraint and the numerical scheme converges more rapidly.

### Normal contact forces

The discretization of the *Continuum Constraint* formulation (5.42) for the normal forces is analogous to the one presented in Section 2.6.4. The term related to the contact constraint in the normal direction, the force vector  $\mathbf{F}_N := (C_c^{CC})_N$ , is the first term of (5.39) which is written as

$$\mathbf{F}_N := (C_c^{CC})_N = \int_{\boldsymbol{\varphi}(\mathcal{B}^c)} (\mathbb{P} \boldsymbol{\sigma}_c) \cdot \nabla^S \boldsymbol{\eta}_c \, dv \quad (5.56)$$

Using the variation of the contact virtual valued functions  $\nabla^S \boldsymbol{\eta}_c$  (2.211) for the projected displacements (5.49) and using the isoparametric concept (2.193), the contact constraint for the *Continuum Constraint* formulation can be written as

$$\begin{aligned} (\boldsymbol{\eta}_c)^T (C_c^{CC})_N &= \int_{\boldsymbol{\varphi}(\mathcal{B}^c)} \mathbb{P} \boldsymbol{\sigma}_c \cdot \nabla^S \boldsymbol{\eta}_c \, dv = \bigcup_{ce=1}^{n_{ce}} \mathbb{P} \int_{\boldsymbol{\varphi}(\Omega_{ce})} (\nabla^S \boldsymbol{\eta}_c)^T \boldsymbol{\sigma}_{ce} \, dV \\ &= \bigcup_{ce=1}^{n_{ce}} \mathbb{P} \sum_{I=1}^{n_n} \boldsymbol{\eta}_I^T \int_{\boldsymbol{\varphi}(\Omega_{ce})} \mathbf{B}_{0I}^{n,T} \boldsymbol{\sigma}_{ce} \, dV \\ &= \bigcup_{ce=1}^{n_{ce}} \mathbb{P} \sum_{I=1}^{n_n} \boldsymbol{\eta}_I^T \int_{\Omega_{\square}} \mathbf{B}_{0I}^{n,T} \boldsymbol{\sigma}_{ce} \det \mathbf{j}_{ce}^n \, d\Omega_{\square} \end{aligned} \quad (5.57)$$

where subscript  $(\cdot)_{ce}$  means *contact element* and superscript  $(\cdot)^n$  means that the variables are computed using the normal projected displacements (5.49). The projection tensor is  $\mathbb{P} = \bar{\mathbf{n}}^1 \otimes \bar{\mathbf{n}}^1$ , considering that the normal to the contact surface  $\bar{\mathbf{n}}^1$  is rate independent.

### 5.6.3 Computation of tangential contact forces

The computation of the tangential forces is possibly the most complex part of the contact characterization. Frictional laws are highly non-linear, and therefore the determination of the friction coefficient is typically rate dependent and also depends on the surface microstructure. In the PFEM we will use a simple model to compute the tangential contact. However the formulation can be extended to model frictional behaviors as complex as the problem requires. A general view of constitutive laws for frictional contact is presented in Section 6.2.

#### Stick case

The tangential constraint for the stick case is described using the second term of equation (5.39):

$$\mathbf{F}_T := (C_c^{CC})_T = \int_{\boldsymbol{\varphi}(\mathcal{B}^c)} (\mathbb{T} \hat{\boldsymbol{\sigma}}_c) \cdot \nabla^S \boldsymbol{\eta}_c \, dv \quad (5.58)$$

where  $\mathbb{T} = (\mathbb{I} - \mathbb{P})$  and  $\hat{\boldsymbol{\sigma}}_c = \boldsymbol{\sigma}_c$  which is described in equation (5.53).

The definition of  $\boldsymbol{\sigma}_c$  presented in equation (5.51) uses the volumetric part of the *Cauchy* stress tensor. This gives a good response in predicting normal contact forces. However when  $(\mathbb{T} \boldsymbol{\sigma}_c)$  is computed for the stick case, some information related to the tangential stresses is missed. Therefore, to describe correctly the stick case,  $\hat{\boldsymbol{\sigma}}_c$  is defined as

$$\hat{\boldsymbol{\sigma}}_c := \boldsymbol{\sigma}_c^{dev} = \frac{\mu}{J^t} (\mathbf{b} - \mathbf{1}) \quad (5.59)$$

where  $J^n$  is the *Jacobi* determinant (2.11) computed using the projected displacements in the tangential direction. Equation (5.59) includes the deviatoric part of the *Cauchy* stress tensor and is computed taking in account the tangential displacements of the contact interface.

For simplicity and robustness, the tangential contact will be described in the PFEM using *Coulomb* law (Section 5.3.3). From the *Coulomb* frictional law the stick-slip behavior is computed by a regularization of the frictional forces (5.3.3). Thus the stick case is described using the same expression as for the slip case, which is presented next.

### Slip case

When sliding is considered the friction forces at the contact interface are described by *Coulomb* law. The tangential frictional force that appears at the contact surface  $\mathbf{F}_T$  is considered to be proportional to the normal contact force  $\mathbf{F}_N$ . This can be written as

$$(\mathbf{F}_T \cdot \bar{\mathbf{n}}^1) > 0 \rightarrow \mathbf{F}_T = -\mu |\mathbf{F}_N \cdot \bar{\mathbf{n}}^1| \cdot \bar{\mathbf{t}}^1 \quad (5.60)$$

where  $\bar{\mathbf{n}}^1$  and  $\bar{\mathbf{t}}^1$  are respectively the normal and the tangential directions to the contact surface associated to each contact element. The tangential contact constraint for the *Continuum Constraint* formulation in the slip case, considering *Coulomb* law, is written as

$$\mathbf{F}_T := (C_c^{CC})_T = \int_{\boldsymbol{\varphi}(B^c)} \boldsymbol{\kappa}_c \cdot \nabla^S \boldsymbol{\eta}_c = \int_{\boldsymbol{\varphi}(B^c)} -\mu \mathbb{T} \boldsymbol{\sigma}_c \cdot \nabla^S \boldsymbol{\eta}_c \quad (5.61)$$

where  $\mathbb{T} = \bar{\mathbf{t}}^1 \otimes \bar{\mathbf{n}}^1$  and  $\boldsymbol{\sigma}_c$  is described by (5.53). When the friction coefficient  $\mu$ , the normal  $\bar{\mathbf{n}}^1$  and tangent  $\bar{\mathbf{t}}^1$  are rate independent, last equation can be written as

$$(C_c^{CC})_T = -\mu \mathbb{T} \int_{\boldsymbol{\varphi}(B^c)} \boldsymbol{\sigma}_c \cdot \nabla^S \boldsymbol{\eta}_c \quad (5.62)$$

This expression can be applied when the contact surface does not suffer big changes in one time step. When the sliding condition is applied, the discretization of the tangential contact constraint for the *Continuum Constraint* method is given by

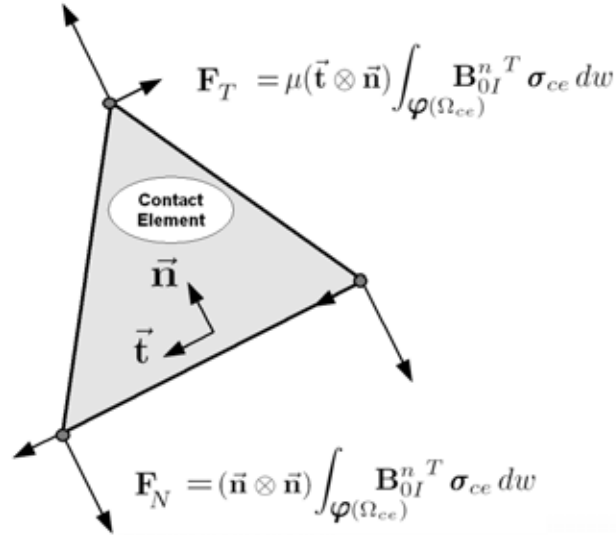
$$\begin{aligned}
(\boldsymbol{\eta}_c)^T (C_c^{CC})_T &= \int_{\varphi(\mathcal{B}^c)} -\mu \mathbb{T} \boldsymbol{\sigma}_c \cdot \nabla^S \boldsymbol{\eta}_c \, dv = \bigcup_{ce=1}^{n_{ce}} -\mu \mathbb{T} \int_{\varphi(\Omega_{ce})} (\nabla^S \boldsymbol{\eta}_c)^T \boldsymbol{\sigma}_{ce} \, dV \\
&= \bigcup_{ce=1}^{n_{ce}} -\mu \mathbb{T} \sum_{I=1}^{n_n} \boldsymbol{\eta}_I^T \int_{\varphi(\Omega_{ce})} \mathbf{B}_{0I}^{nT} \boldsymbol{\sigma}_{ce} \, dV \\
&= \bigcup_{ce=1}^{n_{ce}} -\mu \mathbb{T} \sum_{I=1}^{n_n} \boldsymbol{\eta}_I^T \int_{\Omega_{\square}} \mathbf{B}_{0I}^{nT} \boldsymbol{\sigma}_{ce} \det \mathbf{j}_{ce}^n \, d\Omega_{\square}
\end{aligned} \tag{5.63}$$

The friction coefficient for *Coulomb* law is given in Table 5.1. For a rate dependent coefficient the expression for  $\mu$  is given in (5.27). Note that in this case  $\mu$  must be considered inside the integral.

Looking at the equations (5.63) and (5.57) it can be deduced

$$(C_c^{CC})_T = -\mu \mathbb{T} \frac{(C_c^{CC})_N}{\mathbb{P}} \Rightarrow \sum_{I=1}^{n_n} (C_c^{CC})_T^I = \sum_{I=1}^{n_n} -\mu |(C_c^{CC})_N^I \cdot \bar{\mathbf{n}}^1| \cdot \bar{\mathbf{t}}^1 \tag{5.64}$$

For every node  $I$  the previous expression gives the same relation in forces that was presented in (5.60). The contact forces are shown schematically on a triangular element in Figure 5.18.



**Figure 5.18:** Contact forces in a two-dimensional contact element.

#### 5.6.4 Linearization of the contact constraint

To solve the boundary value problem in the contact problem, the linearization of the nonlinear mathematical model is necessary. As done in Section 2.5 for the weak form of balance of momentum the linearization of the contact constraint is developed. The linearization respect to the current configuration is based on equation (5.43)

The linearization is formulated at the deformation state  $\bar{\varphi}$  where the body is in equilibrium. Using the directional derivative yields to

$$L[\mathbf{C}_c^{CC}]_{\varphi=\bar{\varphi}} = \bar{\mathbf{C}}_c^{CC}(\bar{\varphi}, \eta_c) + D\bar{\mathbf{C}}_c^{CC}(\bar{\varphi}, \eta_c) \cdot \Delta \mathbf{u}^n \quad (5.65)$$

where

$$D\bar{\mathbf{C}}_c^{CC}(\bar{\varphi}, \eta_c) \cdot \Delta \mathbf{u}^n = \int_{\bar{\varphi}(\mathcal{B}^c)} [D\{\mathbb{P}\sigma_c(\bar{\varphi}) + \kappa_c(\bar{\varphi})\} \cdot \Delta \mathbf{u}^n] \cdot \overline{\text{grad}} \eta_c dv \quad (5.66)$$

Developing the expression

$$\begin{aligned} D\bar{\mathbf{C}}_c^{CC}(\bar{\varphi}, \eta_c) \cdot \Delta \mathbf{u}^n &= \int_{\bar{\varphi}(\mathcal{B}^c)} [D\mathbb{P}\sigma_c(\bar{\varphi}) \cdot \Delta \mathbf{u}^n] + [D\kappa_c(\bar{\varphi}) \cdot \Delta \mathbf{u}^n] \cdot \overline{\text{grad}} \eta_c dv \\ &= \underbrace{\int_{\bar{\varphi}(\mathcal{B}^c)} [D\mathbb{P}\sigma_c(\bar{\varphi}) \cdot \Delta \mathbf{u}^n] \cdot \overline{\text{grad}} \eta_c dv}_{\boldsymbol{\eta}_c^T \mathbf{K}_N \Delta \mathbf{u}^n} + \underbrace{\int_{\bar{\varphi}(\mathcal{B}^c)} [D\kappa_c(\bar{\varphi}) \cdot \Delta \mathbf{u}^n] \cdot \overline{\text{grad}} \eta_c dv}_{\boldsymbol{\eta}_c^T \mathbf{K}_T \Delta \mathbf{u}^n} \end{aligned} \quad (5.67)$$

Two different terms can be distinguished in the linearization, the  $\boldsymbol{\eta}_c^T \mathbf{K}_N \Delta \mathbf{u}^n$  term represents the contribution of the normal contact, and  $\boldsymbol{\eta}_c^T \mathbf{K}_T \Delta \mathbf{u}^n$  represents the contribution of the tangential contact. The notation for these terms has been selected so they can be adapted to the subsequent discretization.

#### Linearization of the normal contact

The linearization of the normal contact is expressed as

$$\boldsymbol{\eta}_c^T \mathbf{K}_N \Delta \mathbf{u}^n = \int_{\bar{\varphi}(\mathcal{B}^c)} [D\{\mathbb{P}\sigma_c(\bar{\varphi})\} \cdot \Delta \mathbf{u}^n] \cdot \overline{\text{grad}} \eta_c dv \quad (5.68)$$

Expanding the previous equation

$$\begin{aligned} \boldsymbol{\eta}_c^T \mathbf{K}_N \Delta \mathbf{u}^n &= \int_{\bar{\varphi}(\mathcal{B}^c)} (\overline{\text{grad}} \Delta \mathbf{u}^n \{\mathbb{P}\sigma_c\} + [D\{\mathbb{P}\sigma_c\} \cdot \Delta \mathbf{u}^n]) \cdot \overline{\text{grad}} \eta_c dv \\ &= \int_{\bar{\varphi}(\mathcal{B}^c)} \overline{\text{grad}} \Delta \mathbf{u}^n \{\mathbb{P}\sigma_c\} \cdot \overline{\text{grad}} \eta_c dv + \int_{\bar{\varphi}(\mathcal{B}^c)} \{[D\mathbb{P} \cdot \Delta \mathbf{u}^n] \sigma_c + \mathbb{P}[D\sigma_c \cdot \Delta \mathbf{u}^n]\} \cdot \overline{\text{grad}} \eta_c dv \end{aligned} \quad (5.69)$$

Rewriting the above equation and simplifying yields

$$\boldsymbol{\eta}_c^T \mathbf{K}_N \Delta \mathbf{u}^n = \boldsymbol{\eta}_c^T \mathbf{K}_N^M \Delta \mathbf{u}^n + \boldsymbol{\eta}_c^T \mathbf{K}_N^{nor} \Delta \mathbf{u}^n \quad (5.70)$$

where the first and second terms are

$$\boldsymbol{\eta}_c^T \mathbf{K}_N^M \Delta \mathbf{u}^n = \mathbb{P} \left( \int_{\bar{\varphi}(\mathcal{B}^c)} \overline{\text{grad}} \Delta \mathbf{u}^n \boldsymbol{\sigma}_c \cdot \overline{\text{grad}} \boldsymbol{\eta}_c \, dv + \int_{\bar{\varphi}(\mathcal{B}^c)} [D \boldsymbol{\sigma}_c \cdot \Delta \mathbf{u}^n] \cdot \overline{\text{grad}} \boldsymbol{\eta}_c \, dv \right) \quad (5.71)$$

$$\boldsymbol{\eta}_c^T \mathbf{K}_N^{nor} \Delta \mathbf{u}^n = \int_{\bar{\varphi}(\mathcal{B}^c)} [D \mathbb{P} \cdot \Delta \mathbf{u}^n] \boldsymbol{\sigma}_c \cdot \overline{\text{grad}} \boldsymbol{\eta}_c \, dv \quad (5.72)$$

Equation (5.71) is similar to the linearization of the weak form (2.178) multiplied by the projection tensor  $\mathbb{P}$ . For the linearization of the stresses  $\boldsymbol{\sigma}_c$  an equation similar to (2.179) is used:

$$D \bar{\boldsymbol{\sigma}}_c \cdot \Delta \mathbf{u}^n = \bar{\boldsymbol{\epsilon}}^{vol} [\bar{\nabla}^S \Delta \mathbf{u}^n] \quad (5.73)$$

where  $\bar{\boldsymbol{\epsilon}}^{vol}$  is given by

$$\bar{\boldsymbol{\epsilon}}^{vol} = \frac{K}{J^n} [\mathbb{I} \otimes \mathbb{I} - 2 \ln J^n \mathbb{I}] \quad (5.74)$$

Using the trace operation and considering the symmetry of  $\bar{\boldsymbol{\epsilon}}^{vol}$ , a compact form for (5.71) is obtained

$$\boldsymbol{\eta}_c^T \mathbf{K}_N^M \Delta \mathbf{u}^n = \mathbb{P} \int_{\bar{\varphi}(\mathcal{B}^c)} \{ \overline{\text{grad}} \Delta \mathbf{u}^n \bar{\boldsymbol{\sigma}}_c \cdot \overline{\text{grad}} \boldsymbol{\eta}_c + \bar{\nabla}^S \boldsymbol{\eta}_c \bar{\boldsymbol{\epsilon}}^{vol} [\bar{\nabla}^S \Delta \mathbf{u}^n] \} \, dv \quad (5.75)$$

The last equation can be written as

$$\boldsymbol{\eta}_c^T \mathbf{K}_N^M \Delta \mathbf{u}^n = \boldsymbol{\eta}_c^T (\mathbf{K}_N^{mat} + \mathbf{K}_N^{geo}) \Delta \mathbf{u}^n = \boldsymbol{\eta}_c^T \mathbb{P} (\hat{\mathbf{K}}_N^{mat} + \hat{\mathbf{K}}_N^{geo}) \Delta \mathbf{u}^n \quad (5.76)$$

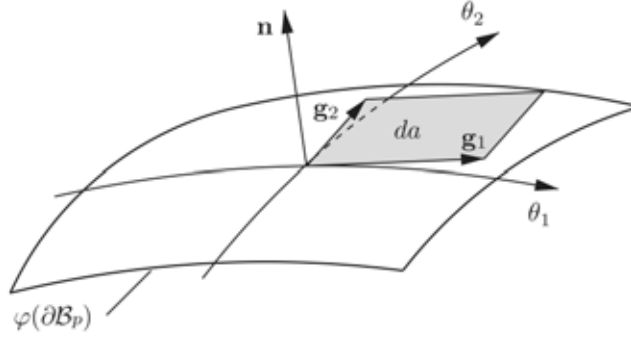
Equation (5.72) describes the linearization of projection tensor  $\mathbb{P} = \bar{\mathbf{n}}^1 \otimes \bar{\mathbf{n}}^1$ . This tensor depends on the normal to the contact surface  $\bar{\mathbf{n}}^1$ . Hence the linearization of tensor  $\mathbb{P}$  corresponds to the linearization of  $\bar{\mathbf{n}}^1$ ; i.e.

$$D \mathbb{P} \cdot \Delta \mathbf{u}^n = D (\bar{\mathbf{n}}^1 \otimes \bar{\mathbf{n}}^1) \cdot \Delta \mathbf{u}^n = 2 \bar{\mathbf{n}}^1 [D \bar{\mathbf{n}}^1 \cdot \Delta \mathbf{u}^n] \quad (5.77)$$



where the tangent vectors  $\mathbf{g}_\alpha$  ( $\alpha = 1, 2$ ) are used for the definition of the normal unit vector  $\mathbf{n}$  (see Figure 5.19). The normal  $\bar{\mathbf{n}}^1$  has the same characteristics as  $\mathbf{n}$  but it is defined in the minimum projection point of the contacting surface.

$$\bar{\mathbf{n}}^1 \simeq \mathbf{n} = \frac{\mathbf{g}_1 \times \mathbf{g}_2}{\|\mathbf{g}_1 \times \mathbf{g}_2\|} \quad (5.78)$$



**Figure 5.19:** Tangent vectors to the contact surface [132].

Using equation (5.78) in (5.77) gives

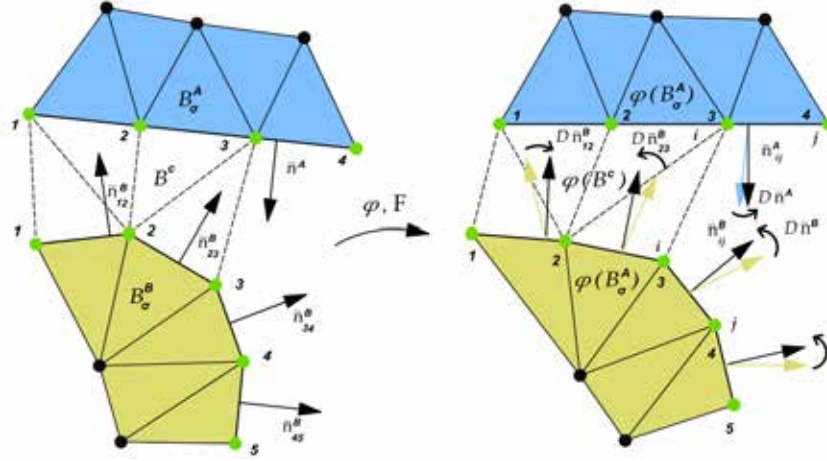
$$\begin{aligned} D \bar{\mathbf{n}}^1 \cdot \Delta \mathbf{u}^n &= D \left( \frac{\mathbf{g}_1 \times \mathbf{g}_2}{\|\mathbf{g}_1 \times \mathbf{g}_2\|} \cdot \Delta \mathbf{u}^n \right) \\ &= \frac{1}{\|\mathbf{g}_1 \times \mathbf{g}_2\|} [D(\mathbf{g}_1 \times \mathbf{g}_2) \cdot \Delta \mathbf{u}^n] + (\mathbf{g}_1 \times \mathbf{g}_2) \left[ D \left( \frac{1}{\|\mathbf{g}_1 \times \mathbf{g}_2\|} \right) \cdot \Delta \mathbf{u}^n \right] \end{aligned} \quad (5.79)$$

The tangent vectors can be computed from the deformation state using equation  $\mathbf{g}_\alpha = \boldsymbol{\varphi}_{,\alpha}$  which comes from the definition of convective coordinates (see [15], [133] and [71]). With the introduction of the displacement field, the tangent vectors have the form  $\boldsymbol{\varphi}_{,\alpha} = (\mathbf{X} + \mathbf{u}^n)_{,\alpha}$ . Hence the linearization of (5.79) yields

$$\begin{aligned} D \bar{\mathbf{n}}^1 \cdot \Delta \mathbf{u}^n &= \frac{1}{\|\mathbf{g}_1 \times \mathbf{g}_2\|} (\Delta \mathbf{u}^n_{,1} \times \boldsymbol{\varphi}_{,2} + \boldsymbol{\varphi}_{,1} \times \Delta \mathbf{u}^n_{,2}) - \frac{(\mathbf{g}_1 \times \mathbf{g}_2)}{\|\mathbf{g}_1 \times \mathbf{g}_2\|^2} [D \|\mathbf{g}_1 \times \mathbf{g}_2\| \cdot \Delta \mathbf{u}^n] \\ &= \frac{1}{\|\mathbf{g}_1 \times \mathbf{g}_2\|} (\Delta \mathbf{u}^n_{,1} \times \boldsymbol{\varphi}_{,2} + \boldsymbol{\varphi}_{,1} \times \Delta \mathbf{u}^n_{,2}) - \frac{(\mathbf{g}_1 \times \mathbf{g}_2)}{\|\mathbf{g}_1 \times \mathbf{g}_2\|^2} \left[ D \left( \sum_i |\mathbf{g}_1 \times \mathbf{g}_2|_i^2 \right)^{\frac{1}{2}} \cdot \Delta \mathbf{u}^n \right] \end{aligned} \quad (5.80)$$

To solve equation (5.80) the parametrization of the contact surface is necessary. It can be performed using the same discretization as for contact elements. This means that for

each triangle or tetrahedra the face on the contact surface defines the tangent vectors and the normal to that surface. The variation of the normal is simply defined by taking into account the displacements of the element face. An example can be seen in Figure 5.20.



**Figure 5.20:** Variation of the surface normals from the reference to the current configuration.

Finally, the contribution of the normal constraint of the contact element to the tangent stiffness matrix is given by the discretization of  $\mathbf{K}_N$ :

$$\Delta(\mathbf{C}_c^{CC})_N \approx \bigcup_{ce=1}^{n_{ce}} \boldsymbol{\eta}_{ce}^T \mathbf{K}_{Nce} \Delta \mathbf{u}_{ce}^n \quad (5.81)$$

Substituting (5.76) into (5.70) gives

$$\boldsymbol{\eta}_c^T \mathbf{K}_N \Delta \mathbf{u}_c^n = \boldsymbol{\eta}_c^T (\mathbf{K}_N^{mat} + \mathbf{K}_N^{geo} + \mathbf{K}_N^{nor}) \Delta \mathbf{u}^n \quad (5.82)$$

In the discretization of the contact domain,  $\mathbf{K}_N^{mat}$  and  $\mathbf{K}_N^{geo}$  are given by

$$\boldsymbol{\eta}_c^T \mathbf{K}_N^{mat} \Delta \mathbf{u}^n = \bigcup_{ce=1}^{n_{ce}} \mathbb{P} \sum_{I=1}^{n_n} \sum_{K=1}^{n_n} \boldsymbol{\eta}_{cI}^T \int_{\varphi(\Omega_{ce})} [\bar{\mathbf{B}}_{0I}^n{}^T \bar{\mathbf{D}}^{vol} \bar{\mathbf{B}}_{0K}^n] dV \Delta \mathbf{u}_K^n \quad (5.83)$$

$$\boldsymbol{\eta}_c^T \mathbf{K}_N^{geo} \Delta \mathbf{u}^n = \bigcup_{ce=1}^{n_{ce}} \mathbb{P} \sum_{I=1}^{n_n} \sum_{K=1}^{n_n} \boldsymbol{\eta}_{cI}^T \int_{\varphi(\Omega_{ce})} [(\nabla_{\bar{x}} N_I^T) \bar{\boldsymbol{\sigma}}_{ce} \nabla_{\bar{x}} N_K] \mathbb{I} dV \Delta \mathbf{u}_K^n \quad (5.84)$$

where  $\bar{\mathbf{D}}^{vol}$  is the incremental material constitutive tensor  $\bar{\mathbf{c}}^{vol}$  in the current configuration.

The discretization of  $\mathbf{K}_N^{nor}$  results in a non general complex expression depending on the parametrization of the surface in the contact element. It can be obtained using (5.80) and operating with following equation

$$\boldsymbol{\eta}_c^T \mathbf{K}_N^{nor} \Delta \mathbf{u}^n = \int_{\bar{\varphi}(\mathcal{B}^c)} 2 \bar{\mathbf{n}}^1 [D \bar{\mathbf{n}}^1 \cdot \Delta \mathbf{u}^n] \boldsymbol{\sigma}_c \cdot \overline{\text{grad}} \boldsymbol{\eta}_c dv \quad (5.85)$$

$\mathbf{K}_N^{nor}$  yields a non-symmetric contribution of the normal constraint into the tangent stiffness matrix.

### Linearization of the tangential contact

For the tangential contact, the linearization is expressed as follows

$$\boldsymbol{\eta}_c^T \mathbf{K}_T \Delta \mathbf{u}^n = \int_{\bar{\varphi}(\mathcal{B}^c)} [D \boldsymbol{\kappa}_c(\bar{\varphi}) \cdot \Delta \mathbf{u}^n] \cdot \overline{\text{grad}} \boldsymbol{\eta}_c dv \quad (5.86)$$

Using *Coulomb* law,  $\boldsymbol{\kappa}_c$  can be replaced by (5.44) and the above equation leads to

$$\boldsymbol{\eta}_c^T \mathbf{K}_T \Delta \mathbf{u}^n = \int_{\bar{\varphi}(\mathcal{B}^c)} [D \{-\mu \mathbb{T} \boldsymbol{\sigma}_c(\bar{\varphi})\} \cdot \Delta \mathbf{u}^n] \cdot \overline{\text{grad}} \boldsymbol{\eta}_c dv \quad (5.87)$$

The most general form has to include the rate dependency of the normal and tangential vector contained in  $\mathbb{T} = \bar{\mathbf{t}}^1 \otimes \bar{\mathbf{n}}^1$  and the rate dependency of a non constant friction coefficient  $-\mu(\bar{\varphi})$ . Under these assumptions the linearization of (5.87) turns to a very complex expression which is sometimes useless for numerical computations.

Some simplifications can be done. The first one is considering the friction coefficient as a constant variable. This yields

$$\begin{aligned} \boldsymbol{\eta}_c^T \mathbf{K}_T \Delta \mathbf{u}^n &= -\mu \int_{\bar{\varphi}(\mathcal{B}^c)} [D \{\mathbb{T} \boldsymbol{\sigma}_c(\bar{\varphi})\} \cdot \Delta \mathbf{u}^n] \cdot \overline{\text{grad}} \boldsymbol{\eta}_c dv \\ &= -\mu \int_{\bar{\varphi}(\mathcal{B}^c)} (\overline{\text{grad}} \Delta \mathbf{u}^n \{\mathbb{T} \boldsymbol{\sigma}_c\} + [D \{\mathbb{T} \boldsymbol{\sigma}_c\} \cdot \Delta \mathbf{u}^n]) \cdot \overline{\text{grad}} \boldsymbol{\eta}_c dv \\ &= -\mu \left( \int_{\bar{\varphi}(\mathcal{B}^c)} \overline{\text{grad}} \Delta \mathbf{u}^n \{\mathbb{T} \boldsymbol{\sigma}_c\} \cdot \overline{\text{grad}} \boldsymbol{\eta}_c dv + \int_{\bar{\varphi}(\mathcal{B}^c)} \{[D \mathbb{T} \cdot \Delta \mathbf{u}^n] \boldsymbol{\sigma}_c + \mathbb{T} [D \boldsymbol{\sigma}_c \cdot \Delta \mathbf{u}^n]\} \cdot \overline{\text{grad}} \boldsymbol{\eta}_c dv \right) \end{aligned} \quad (5.88)$$

Comparing last equation with (5.69) it can be seen that they are very similar. Hence expression (5.88) can be rewritten as

$$\boldsymbol{\eta}_c^T \mathbf{K}_T \Delta \mathbf{u}^n = \boldsymbol{\eta}_c^T \mathbf{K}_T^M \Delta \mathbf{u}^n + \boldsymbol{\eta}_c^T \mathbf{K}_T^{tan} \Delta \mathbf{u}^n \quad (5.89)$$

where the first and second term are

$$\boldsymbol{\eta}_c^T \mathbf{K}_T^M \Delta \mathbf{u}^n = -\mu \mathbb{T} \left( \int_{\bar{\varphi}(\mathcal{B}^c)} \overline{\text{grad}} \Delta \mathbf{u}^n \boldsymbol{\sigma}_c \cdot \overline{\text{grad}} \boldsymbol{\eta}_c dv + \int_{\bar{\varphi}(\mathcal{B}^c)} [D \boldsymbol{\sigma}_c \cdot \Delta \mathbf{u}^n] \cdot \overline{\text{grad}} \boldsymbol{\eta}_c dv \right) \quad (5.90)$$

$$\boldsymbol{\eta}_c^T \mathbf{K}_T^{tan} \Delta \mathbf{u}^n = -\mu \int_{\bar{\varphi}(\mathcal{B}^c)} [D \mathbb{T} \cdot \Delta \mathbf{u}^n] \boldsymbol{\sigma}_c \cdot \overline{\text{grad}} \boldsymbol{\eta}_c dv \quad (5.91)$$

Through the inference to the normal contact, the contribution of the contact constraint on the tangent stiffness matrix is built using the discretization of  $\mathbf{K}_T$ :

$$\Delta(\mathbf{C}_c^{CC})_T \approx \bigcup_{ce=1}^{n_{ce}} \boldsymbol{\eta}_{ce}^T \mathbf{K}_{Tce} \Delta \mathbf{u}_{ce}^n \quad (5.92)$$

Substituting (5.76) into (5.89) for the tangential contact

$$\boldsymbol{\eta}_c^T \mathbf{K}_T \Delta \mathbf{u}_c^n = \boldsymbol{\eta}_c^T (\mathbf{K}_T^{mat} + \mathbf{K}_T^{geo} + \mathbf{K}_T^{tan}) \Delta \mathbf{u}^n \quad (5.93)$$

$\mathbf{K}_T^{mat}$  and  $\mathbf{K}_T^{geo}$  are given after the discretization of the contact domain by

$$\boldsymbol{\eta}_c^T \mathbf{K}_T^{mat} \Delta \mathbf{u}^n = \bigcup_{ce=1}^{n_{ce}} (-\mu) \mathbb{T} \sum_{I=1}^{n_n} \sum_{K=1}^{n_n} \boldsymbol{\eta}_{cI}^T \int_{\boldsymbol{\varphi}(\Omega_{ce})} [\bar{\mathbf{B}}_{0I}^{nT} \bar{\mathbf{D}}^{vol} \bar{\mathbf{B}}_{0K}^n] dV \Delta \mathbf{u}_K^n \quad (5.94)$$

$$\boldsymbol{\eta}_c^T \mathbf{K}_T^{geo} \Delta \mathbf{u}^n = \bigcup_{ce=1}^{n_{ce}} (-\mu) \mathbb{T} \sum_{I=1}^{n_n} \sum_{K=1}^{n_n} \boldsymbol{\eta}_{cI}^T \int_{\boldsymbol{\varphi}(\Omega_{ce})} [(\nabla_{\bar{\mathbf{x}}} N_I^T)^T \bar{\boldsymbol{\sigma}}_{ce} \nabla_{\bar{\mathbf{x}}} N_K] \mathbb{I} dV \Delta \mathbf{u}_K^n \quad (5.95)$$

where  $\bar{\mathbf{D}}^{vol}$  is the incremental material constitutive tensor  $\bar{\boldsymbol{\epsilon}}^{vol}$  in the current configuration.

The discretization of  $\mathbf{K}_T^{tan}$  in (5.91) has the same problems as  $\mathbf{K}_N^{nor}$  in Section 5.6.4. Thus, its linearization depends on the particular discretization of the contact surface. This makes the operation and expansion of the equation very complex. A similar thing happens when the friction coefficient  $\mu$  is considered as rate dependent. The linearization gets even more complicated.

The contribution of the total contact in the tangent stiffness matrix is given by

$$\Delta(\mathbf{C}_c^{CC}) = \Delta(\mathbf{C}_c^{CC})_N + \Delta(\mathbf{C}_c^{CC})_T \approx \bigcup_{ce=1}^{n_{ce}} \boldsymbol{\eta}_{ce}^T (\mathbf{K}_{Nce} + \mathbf{K}_{Tce}) \Delta \mathbf{u}_{ce}^n \quad (5.96)$$

Taking in account that it does not exist a practical expression for some terms of the contact tangent stiffness, some simplifications are needed. In contact problems the convergence of the implicit solution scheme is a weak property that does not depend only on the tangent matrix for the contact constraint. Experience says that a good approximation for the tangent has a better performance than the real one. Therefore, with some simplifying assumptions a good approximation for the tangent of the *Continuum Constraint Method* can be obtained.

### 5.6.5 Approximation for the tangent

The linear model or linearized model of the nonlinear equations was presented in (3.27)

$$\mathbf{r} - \mathbf{A} \Delta \mathbf{u} = 0 \quad (5.97)$$

where  $\mathbf{A}$  is the *Jacobian* matrix of the system of equations in the *Newmark* scheme given by

$$\mathbf{A} = \frac{\mathbf{M}}{\beta \Delta t^2} + (\mathbf{K}^{mat} + \mathbf{K}^{geo} + \mathbf{K}^{cont}) - \mathbf{K}^{ext} \quad \text{for } \beta > 0 \quad (5.98)$$

In the above equation the contribution to the tangent related to the contact constraints has been introduced by  $\mathbf{K}^{con}$  and is written as

$$\boldsymbol{\eta}_c^T \mathbf{K}^{con} \Delta \mathbf{u} = \boldsymbol{\eta}_c^T (\mathbf{K}_N + \mathbf{K}_T) \Delta \mathbf{u}^n \quad (5.99)$$

From equation (5.49) the increment of the normal displacements can be written as

$$\Delta \mathbf{u}^n = \mathbf{P} \Delta \mathbf{u} \quad (5.100)$$

From equation (5.100) we deduce

$$\boldsymbol{\eta}_c^T \mathbf{K}^{con} \Delta \mathbf{u} = \boldsymbol{\eta}_c^T (\mathbf{K}_N + \mathbf{K}_T) \mathbf{P} \Delta \mathbf{u} \Rightarrow \mathbf{K}^{con} = (\mathbf{K}_N + \mathbf{K}_T) \mathbf{P} \quad (5.101)$$

There are two hypothesis that can be applied for obtaining a practical expression for the tangent matrix  $\mathbf{K}^{con}$ . The first one is to consider that the normal and tangent vectors to the contact surface have a negligible change during a time step. This is true when the contact surfaces do not deform much during the contact interaction. The second hypothesis is applied to the frictional contact and consists in neglecting the change of the friction coefficient  $\mu = \mu(\mathbf{g}_T, p_N, \theta)$  during a time step. This happens when the variables that define  $\mu$  (e.g relative velocity between contact surfaces) undergo small changes within the same time step. These two hypothesis are very restrictive and can be considered not suitable for finite deformations. However, the resultant tangent matrix is also a good approximation for this case. It gives a good convergence rate when the hypothesis are fulfilled and a small degradation of the convergence when they are not. Anyway this is not so important. There are other numerical problems related to contact physics that degrade the convergence rate more. One example is that is the so called *Jamming* problem, also known as zig-zagging. It means that the algorithm jumps in consecutive iterations between two values. This happens when variables oscillate too much within the same time step.

Using the last hypotheses the expressions for  $\mathbf{K}_N$  and  $\mathbf{K}_T$  are

$$\mathbf{K}_N = (\mathbf{K}_N^{mat} + \mathbf{K}_N^{geo}) \mathbf{P} \quad (5.102)$$

$$\mathbf{K}_T = (\mathbf{K}_T^{mat} + \mathbf{K}_T^{geo}) \mathbf{P} \quad (5.103)$$

where the finite element matrices for (5.102) and (5.103) were developed in Sections 5.6.4 and 5.6.4. The last expressions can be written in elemental form as

$$\begin{aligned} \bigcup_{ce=1}^{n_{ce}} \underbrace{(\mathbf{K}_{N_{ce}} + \mathbf{K}_{T_{ce}})}_{\mathbf{K}_{ce}^{con}} &= \bigcup_{ce=1}^{n_{ce}} (\mathbb{P} (\hat{\mathbf{K}}_{N_{ce}}^{mat} + \hat{\mathbf{K}}_{N_{ce}}^{geo}) \mathbb{P} - \mu \mathbb{T} (\hat{\mathbf{K}}_{T_{ce}}^{mat} + \hat{\mathbf{K}}_{T_{ce}}^{geo}) \mathbb{P}) \\ &= \bigcup_{ce=1}^{n_{ce}} (\mathbb{P} (\hat{\mathbf{K}}_{ce}^{con}) \mathbb{P} - \mu \mathbb{T} (\hat{\mathbf{K}}_{ce}^{con}) \mathbb{P}) \end{aligned} \quad (5.104)$$

with

$$\hat{\mathbf{K}}_{ce}^{cont} = \hat{\mathbf{K}}_{ce}^{mat} + \hat{\mathbf{K}}_{ce}^{geo} \quad (5.105)$$

and

$$\hat{\mathbf{K}}_{ce}^{mat} = \hat{\mathbf{K}}_{N_{ce}}^{mat} = \hat{\mathbf{K}}_{T_{ce}}^{mat} \quad \& \quad \hat{\mathbf{K}}_{ce}^{geo} = \hat{\mathbf{K}}_{N_{ce}}^{geo} = \hat{\mathbf{K}}_{T_{ce}}^{geo} \quad (5.106)$$

where matrices  $\hat{\mathbf{K}}_{ce}^{mat}$  and  $\hat{\mathbf{K}}_{ce}^{geo}$  are defined by

$$\hat{\mathbf{K}}_{ce}^{mat} = \bigcup_{ce=1}^{n_{ce}} \sum_{I=1}^{n_n} \sum_{K=1}^{n_n} \int_{\boldsymbol{\varphi}(\Omega_{ce})} [\bar{\mathbf{B}}_{0I}^{nT} \bar{\mathbf{D}}^{vol} \bar{\mathbf{B}}_{0K}^n] dV \quad (5.107)$$

$$\hat{\mathbf{K}}_{ce}^{geo} = \bigcup_{ce=1}^{n_{ce}} \sum_{I=1}^{n_n} \sum_{K=1}^{n_n} \int_{\boldsymbol{\varphi}(\Omega_{ce})} [(\nabla_{\bar{\mathbf{x}}} N_I^T) \bar{\boldsymbol{\sigma}}_{ce} \nabla_{\bar{\mathbf{x}}} N_K] \mathbb{I} dV \quad (5.108)$$

$$(5.109)$$

Note that  $(\mathbb{P} \hat{\mathbf{K}}_{ce}^{con} \mathbb{P})$  is symmetric. That gives a symmetric tangent matrix for the normal contact  $(\mathbf{K}_N \mathbb{P}) = (\mathbf{K}_N \mathbb{P})^T$ . This is not the case for  $(\mathbf{K}_T \mathbb{P})$  which is non-symmetric. Summarizing the above equations, the approximated tangent matrix for the contact contribution is given by

$$\boxed{\mathbf{K}^{con} = \bigcup_{e=1}^{n_{ce}} (\mathbb{P} (\hat{\mathbf{K}}_{ce}^{con}) \mathbb{P} - \mu \mathbb{T} (\hat{\mathbf{K}}_{ce}^{con}) \mathbb{P})} \quad (5.110)$$

where  $\hat{\mathbf{K}}_{ce}^{con}$  is defined in (5.105). This matrix introduces a non-symmetry in the contact elemental stiffness matrices. The non-symmetry is transferred to the global system by means of  $\mathbf{K}^{con}$ . With this scheme the *Jacobian* matrix of the global system (5.98) is not symmetric but the convergence improves substantially.

## 5.7 Implicit contact scheme

The global solution algorithms for implicit contact can not deal with a constraint variational inequality. Hence, as introduced in Section 5.4, the *active set* strategy, which assumes that the contact interface is known, is applied to replace the variational inequality by an unconstrained equality including the contact constraints. Active set strategies are well known from optimization theory, ([75], [14]). In the PFEM the contact set is an interface mesh of triangles or tetrahedra depending on the problem dimensions (2D or 3D).

Once the active set is determined, the contact constraints are assembled with the internal and external virtual work. Next, the *Newton* method is chosen to solve the nonlinear discrete system of equations iteratively.

The number of components of the equations can change due to changes in the active contact set. Hence a stable solution using *Newton* method can not be guaranteed. There are two strategies to handle an algorithm including an active set strategy and *Newton* method.

The first strategy is denoted *Embedded Newton* iteration. The *Newton* iteration is computed within an inner loop embedded in the outer loop of the contact search. The strategy is presented in Figure 5.21.

Within the inner loop, convergence of the *Newton* iteration can be guaranteed because during the computation the active set is kept fix. After the convergence, the active set is checked by comparing the actual and the previous residual vector.

When the convergence of the *Newton* loop is achieved the consistence of the contact constraints has to be evaluated. This means checking if the configuration of the contact has changed during the time step. If the variation is small, the analysis continues otherwise, the *Newton* loop is computed again with the new configuration of active contact elements. If the difference between the current residual vector and the one of the previous active set remains within the computer accuracy, the active set is converged and therefore is unique. Then, the solution is found with the previous converged *Newton* loop.

In the flowchart of Figure 5.21 some general schemes are intrinsically included. These are: the solution scheme or *Newton* loop, presented in Figure 3.13 and the remeshing scheme, presented in Figure 4.13. The details can be found in Sections 3.6.4 and 4.4.3. For the contact solution scheme internal forces and contact forces have to be computed. In the system tangent matrix the contact stiffness is added. These are the new variables incorporated to the standard system of solution for contact.

The second strategy is depicted in Figure 5.22 and is denoted *Rotational Execution*. In this case, the active set is updated before every single *Newton* iteration. The convergence for the uniqueness of the active set is checked within the *Newton* iteration. The stop criterion consists in two checks: the first one checks if the equilibrium condition is fulfilled for the actual active and the second one checks if the active set is unique. This procedure does not converge quadratically, although it yields the solution faster than the *Embedded Newton* scheme as only one loop has to be run through.

Both strategies converge but they have to be handled with care. There is a problem

which often occurs and is known from optimization literature as jamming or zig-zagging. It happens when the active set is not unique, and there are more than one possible solutions. Then, the algorithm jumps between the solutions of each active set. This problem can be typically overcome by increasing the load, as additional loads can change the contact constraints such that the active set becomes unique. For that, an adaptation of the time step is considered in the active set loop. After some iterations the time step is changed in order to find a position of equilibrium. If that does not work, the active set update can be interrupted and the current configuration can be accepted along some few time steps. Recall that the contact surfaces are approximated with finite elements, so the unique solution of the continuous problem is modified due to the discretization and becomes non unique.

In the previous strategies the critical time step for contact  $\Delta t_c$  is computed at the beginning of the active set loop. This object refers to a kinematic criterion which is used to detect if the current time step is too large. To obtain this critical value  $\Delta t_c$  the relative velocity of the contacting bodies and the size of the contact elements are taken in account as will be explained in detail in Section 5.8.4.

It also has to be pointed out that the contact search of the PFEM automatically gives a potential contact elements. Then, the checks for the active contact set is reduced to these elements and the computational cost is minimum.



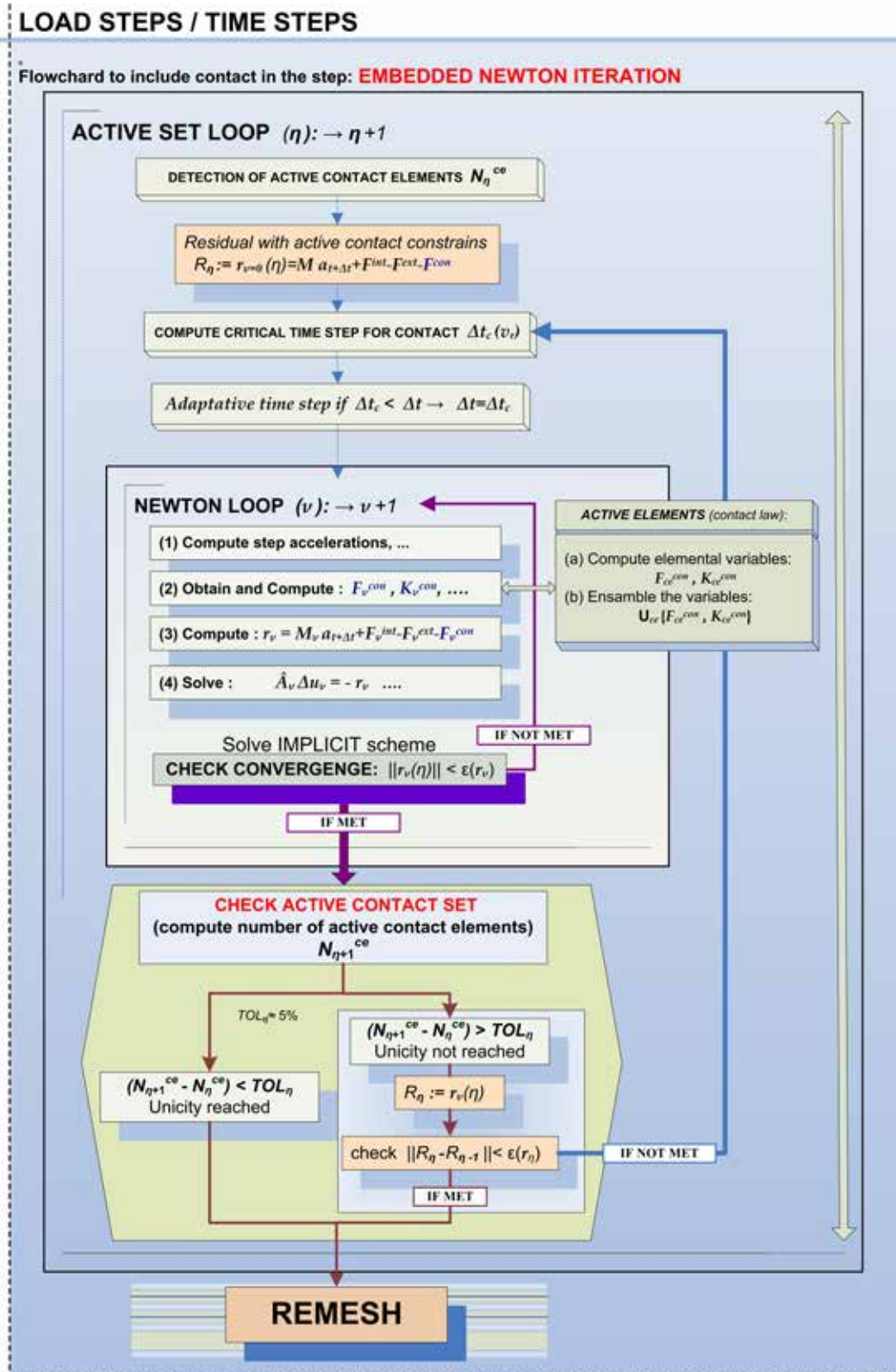


Figure 5.21: Embedded Newton iteration with contact search.

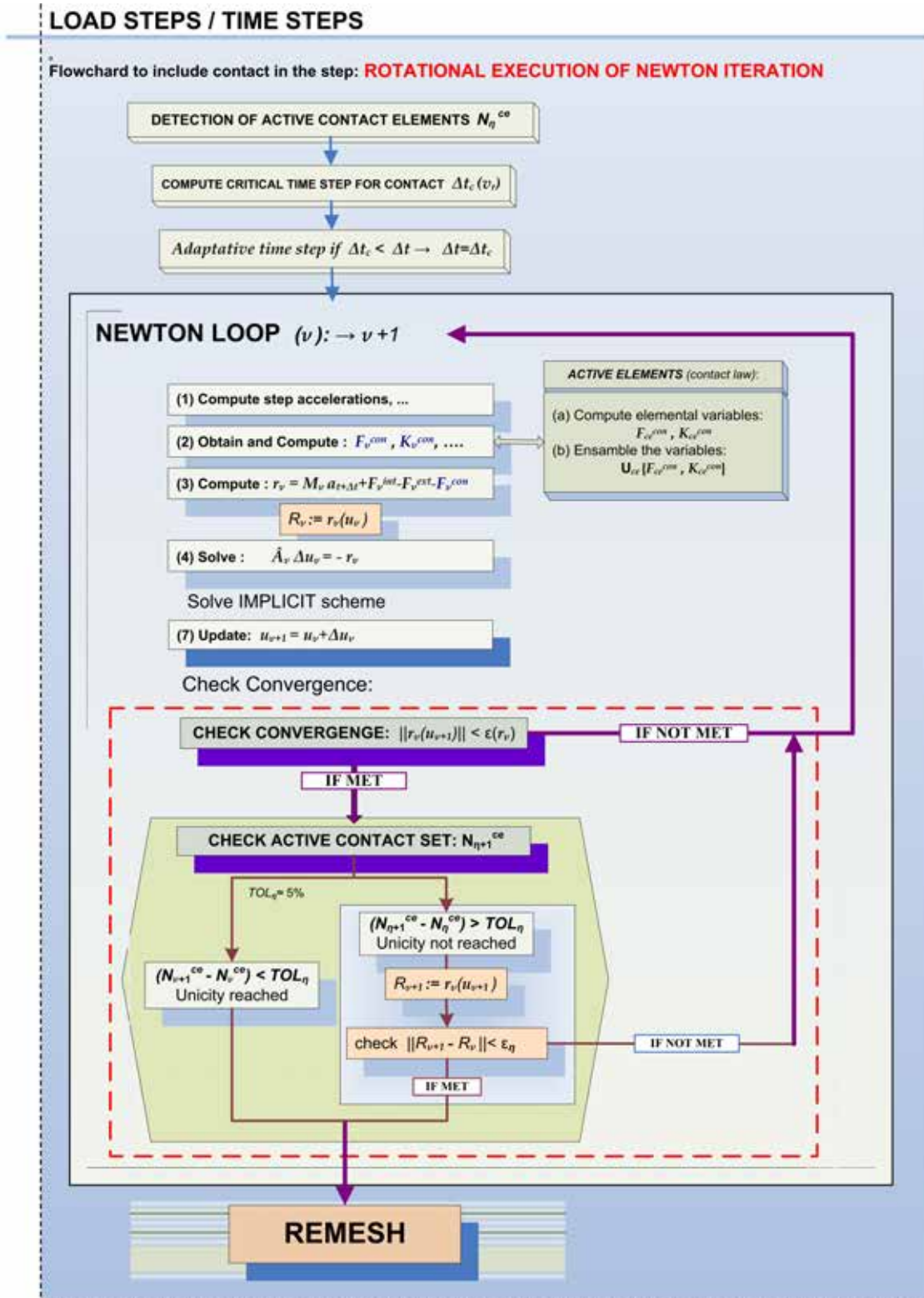


Figure 5.22: Rotational Execution of contact search and Newton iteration.

## 5.8 Impact

Physical impact can be defined as the process involved in the collision of two or more objects. Commonly the collisions in which the mass effect of both impinging bodies must be taken in account is limited. The term at once conveys the idea of abruptness and is different from the case of static or rapid loading by the nature of its application. Forces created by collisions are exerted and removed in a very short interval of time and initiate stress waves which travel away from the region of contact. The static loading is regarded as a series of equilibrium states and requires no consideration of accelerating wave effects. Rapid loading is usually not produced by means of collision, and normally involves longer loading times than found in impact processes and contact deformations do not occur.

When two bodies which have different velocities come into contact an impact occurs. Within an impact analysis one is interested in the velocities of the bodies after impact and in the impact force as a function of time. The velocities are associated with the waves traveling along the bodies. At the same time there is also a stress pulse which travels with the same velocity. When such a pulse reaches the free ends of the bodies, reflection occurs at the free boundaries with equal amplitude in the stress pulse but with opposite velocity. The most simple examples of this phenomena are one-dimensional problems of impacting bars; see Figures 5.23 and 5.27.

### 5.8.1 Longitudinal collinear impact of rods

The solution of problems of impact of bars with the same cross-section can be derived from the 1D wave equation

$$EA \frac{\partial^2 u}{\partial x^2} = -\rho A \frac{\partial^2 u}{\partial t^2} \quad (5.111)$$

The classical theory of elasticity provides the basis of wave analysis in elastic solids. The solution of (5.111) is

$$u(t) = f(x - ct) + g(x + ct) \quad \text{with} \quad c = \sqrt{\frac{E}{\rho}} \quad (5.112)$$

where  $c$  denotes the speed of wave travelling in the bars. Function  $f$  corresponds to a wave travelling in the  $x$ -direction of the bar, while  $g$  is associated with a wave travelling in the opposite direction. The differentiation of (5.112) with respect to  $x$  and  $t$  yields a linear relationship between the stress at any point in the bar and the particle velocity.

$$\sigma = E \varepsilon = \frac{E}{c} \frac{\partial u}{\partial t} = \rho c \frac{\partial u}{\partial t} \quad (5.113)$$

The derivations of the solution can be found in [47]. The bodies will be in contact until the reflected dynamic wave generated in the contact reaches again the contact point. This determines the end of the contact. For two colliding bars, this time corresponds to

$$T = \frac{2l}{c} \quad (5.114)$$

where  $l$  is the length of the longer bar and  $c$  is the speed wave travelling in the bar. The contact force generated in the collision is determined by equation (5.112).

There are two important implications in the numerical implementation. The first one is that the impact time is very short and the stresses generated are high. Hence, numerical methods to solve impact problems have to include nonlinear material behavior and have to be designed for short time responses. To capture the dynamic behavior of the contact an appropriate time step must be computed. The second one is that due to the possibility of high oscillatory responses near wave fronts, one has to be careful when constructing algorithms for impact problems, in the sense that one should not destroy the wave front characteristics within the numerical scheme.

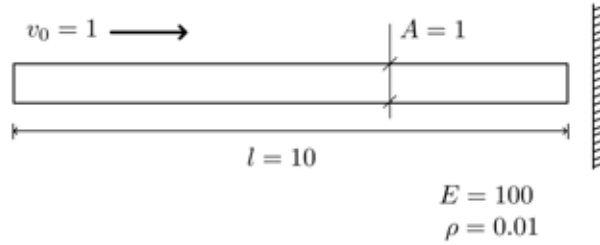
There are also some other aspects to consider. The standard contact conditions describe that no penetration can occur at the contact point, and also that the contact force has to be a compression force. Usually the situation is more complex. We must consider that impact of bodies with curved or pointed surfaces is accompanied by penetration of one member into the other. This phenomena can generate permanent deformations in the contact zone. Also the transmission of stress pulses can involve plastic strains.

### 5.8.2 Impact of one rod on a rigid wall

This is a 1D contact example of one plane-ended bar colliding a rigid wall. The classical theory postulates a uniform-stress distribution across any section of the rod. Impact occurs when a bar moving with a constant velocity  $v_0$  collides the stiff wall. At that instant a dynamic wave is produced and evolves along the bar. The wave reflects in the free end and returns to the contact point. When it reaches the contact point the contact ends and the bar moves with a velocity  $(-v_0)$ .

Classical impact theory for rigid bodies, under the assumption of an elastic impact, does not consider the oscillations remaining in the elastic bodies after the impact. The impact time  $T_c$  is also neglected for rigid body impact. This gives some differences in the results computed in the deformable body theory using numerical methods. This means that after onset of contact the bar will move with a velocity similar to  $(-v_0)$  and a velocity wave will remain travelling in the body from one edge to the other.

For the numerical analysis of the problem the *Continuum Constrain Method* is used for the contact treatment with the PFEM. The method is designed for 2D and 3D geometries but it will be constrained properly to simulate the 1D problem. The discretization of the geometry is simple in order to represent uniform-stress distribution across any section of the bar. The problem characteristics are shown in Figure 5.23.



**Figure 5.23:** Longitudinal impact of a bar against a rigid wall

With the defined properties of the problem, the speed wave inside of the bar is computed as

$$c = \sqrt{\frac{E}{\rho}} = \sqrt{\frac{100}{0.01}} = 100 \quad (5.115)$$

The contact time can be determined by (5.114) as

$$T_c = \frac{2l}{c} = 2 \cdot \frac{10}{100} = 0.2 \quad (5.116)$$

and the theoretical contact force is

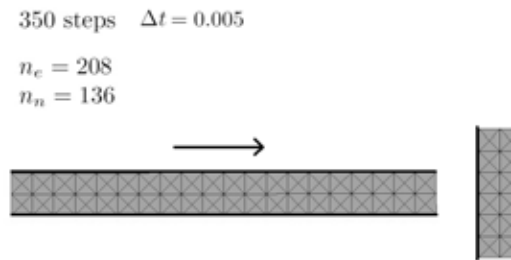
$$F_c = \rho c v_0 = 0.01 \cdot 100 \cdot 1 = 1 \quad (5.117)$$

Note that the units of the variables must be consistent.

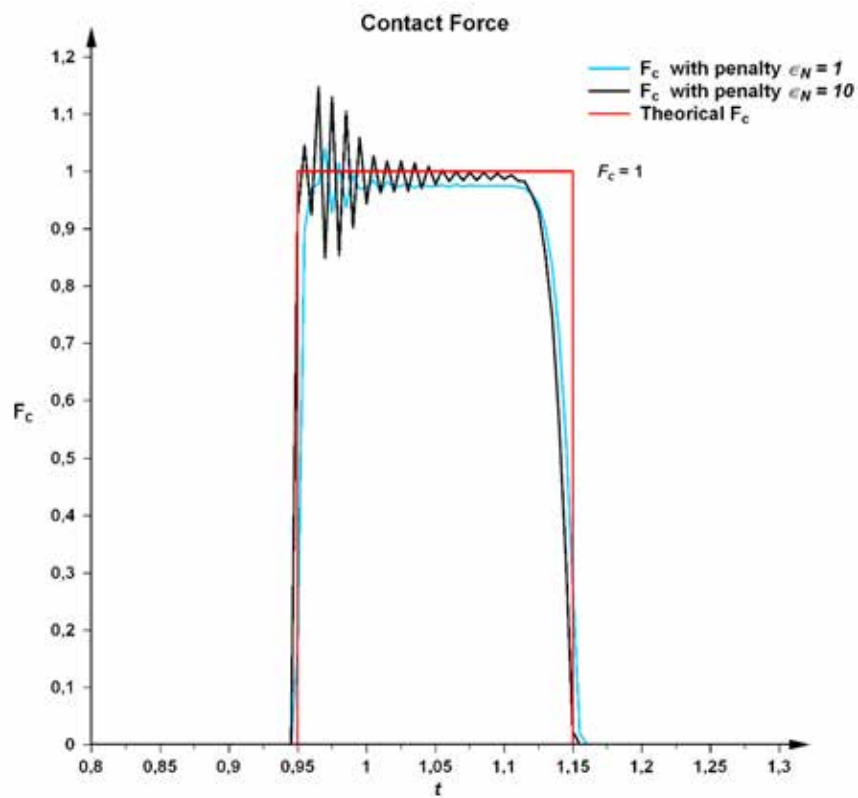
For the numerical analysis the time step must be smaller than the contact time  $T_c$  in order to capture correctly the interaction. The parameters and mesh used are summarized in Figure 5.24.

Figure 5.23 shows the results of the numerical analysis. It reproduces the contact time correctly. It can be seen that the contact is active during  $T_c = 0.2$ . Also, the contact force is slightly different than the theoretical one and presents an oscillation along the interval. The curves represent the obtained force using two different stiffness penalizations for the contact elements. Results show that with a larger penalty factor  $\epsilon_N$  the average force is closer to the theoretical one but the oscillation is also much bigger. The tests with larger and smaller penalizations present the same relative behavior. See Figure 5.26.

The penalty factor used in the *Continuum Constrain Method* ensures that the stiffness of the contact elements is larger than the stiffness of the contacting bodies. This parameter is not constant during the analysis. When a dynamic problem is modeled the relative



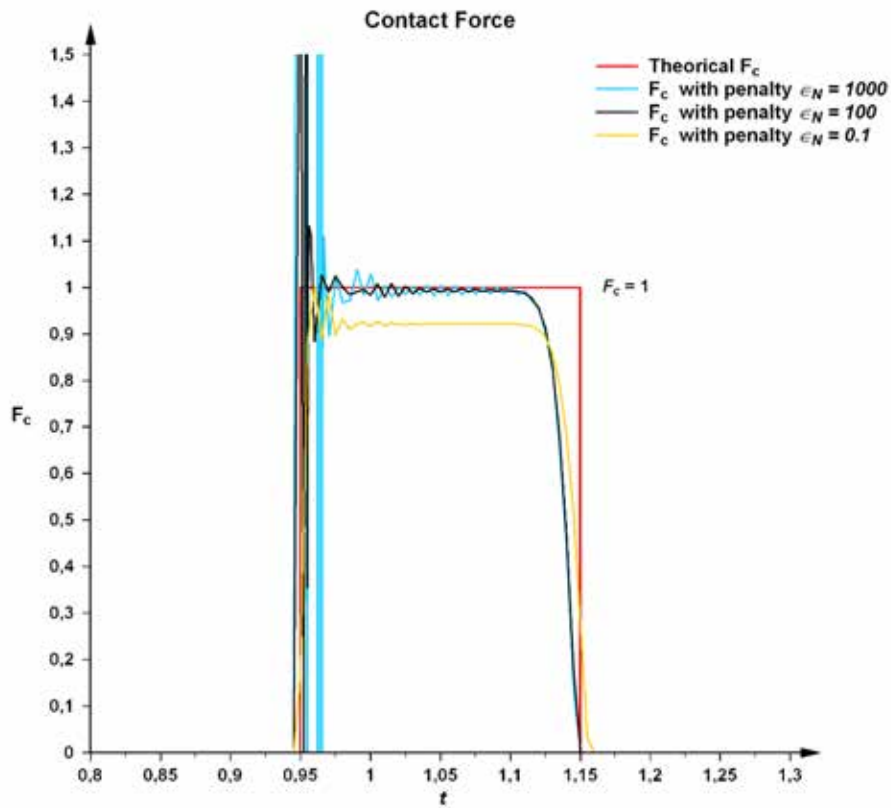
**Figure 5.24:** Mesh and boundary constrained lines for the numerical analysis of a bar colliding a rigid wall



**Figure 5.25:** Evolution of the contact force for a bar colliding a rigid wall

velocity of the contact areas contribute to reduce or increase this parameter. The algorithm is presented in Section 5.6.2. The non-linear penalty factor improves the stability of the contact forces reducing the force oscillation.

At the same time the adaptation of the penalty parameter ensures the good charac-



**Figure 5.26:** Evolution of the contact force for a bar colliding a rigid wall for bigger and smaller penalizations

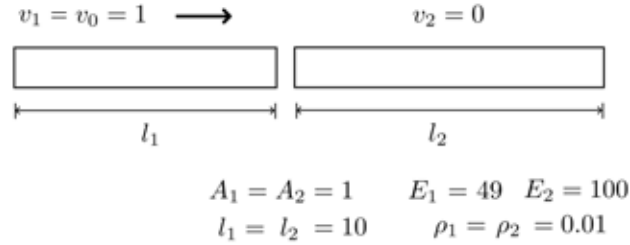
terization of the contact with an error lower than 10% for  $\epsilon_N \geq 1$ .

### 5.8.3 Impact of two rods

The next example is the 1D impact of two rods. A bar of length  $l_1$  impacts on a bar of length  $l_2$ . Both bars have some different material properties  $E_1 = 49$  and  $E_2 = 100$ , densities  $\rho_1 = \rho_2 = 0.01$  and areas  $A_1 = A_2 = 1$ . The left bar has an initial velocity of  $v_0 = 1$ , whereas the right bar is at rest. The specific parameters for the problem are presented in Figure 5.27.

The solution of this problem can be found in [47]. The contact times can be determined by (5.114) as

$$T_1 = \frac{2l_1}{c_1} = \frac{2l_1}{\sqrt{\frac{E_1}{\rho_1}}} = 0.285 \quad (5.118)$$



**Figure 5.27:** Longitudinal impact of two bars

$$T_2 = \frac{2l_2}{c_2} = \frac{2l_2}{\sqrt{\frac{E_2}{\rho_2}}} = 0.2 \quad (5.119)$$

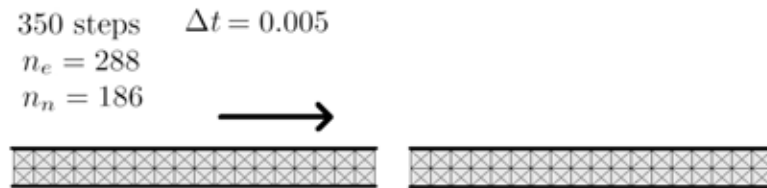
and the theoretical contact forces are

$$F_{c1} = \rho_1 \frac{c_1 c_2}{c_1 + c_2} (v_1)_0 = 0.412 \quad 0 < t < T_2 \quad (5.120)$$

$$F_{c2} = \rho_2 \frac{c_2 - c_1}{4} (v_1)_0 = 0.075 \quad T_2 < t < T_1 \quad (5.121)$$

As in the previous example the units of the variables must be consistent.

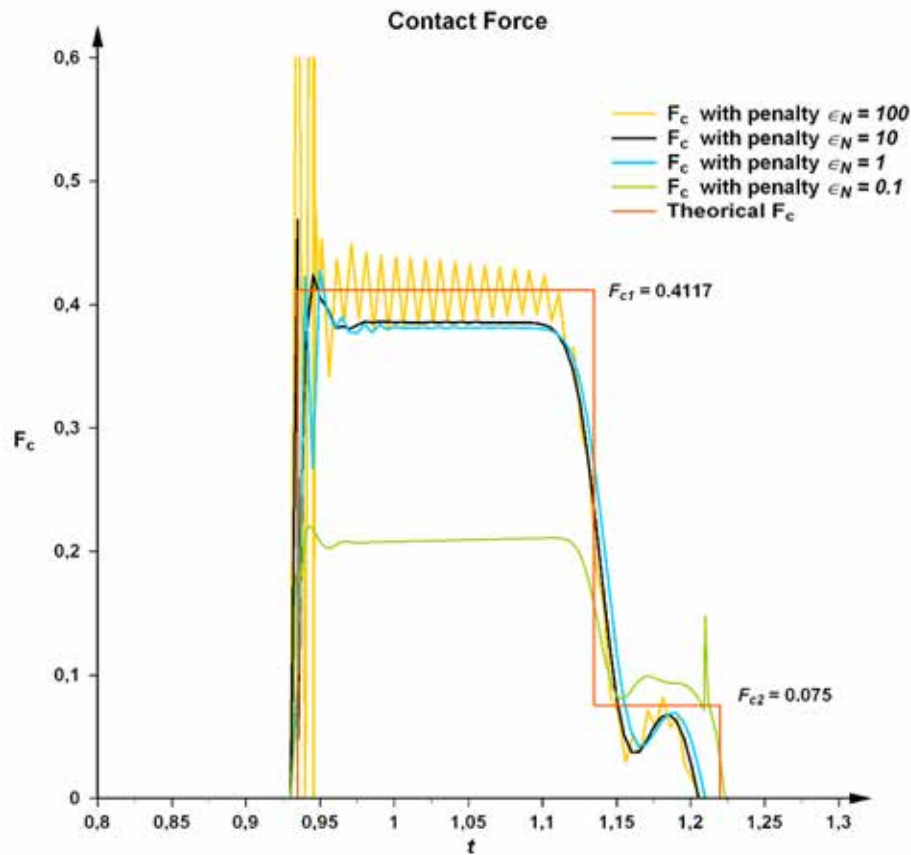
The time step of the numerical analysis must be smaller than the contact time  $T_c$  in order to capture correctly the interaction. The mesh and the parameters for the numerical model are summarized in Figure 5.28.



**Figure 5.28:** Mesh and boundary constrained lines for the numerical analysis of two colliding bars

The results are presented in Figure 5.29. It can be observed how two steps in the contact forces are present. The contact time is also captured and is similar to the the-





**Figure 5.29:** Evolution of the contact force for two colliding bars

oretical one where for the first force is  $T_1 = 0.2$  and for the second is  $T_2 = 0.285$ . The numerical scheme has difficulties to represent exactly the theoretic jump from the contact force  $F_{c1} = 0.412$  to the contact force  $F_{c2} = 0.075$ . That is because a sudden contact force produces a numerical oscillation in the solution.

The magnitude of the contact force is also slightly different from the theoretical one and experiences some oscillation and a smoothing at the edges of the interval. The curves represent the obtained force using different types of penalization for the contact elements. The results show the same trend as for the previous example (Section 5.8.2). With a larger penalty factor  $\epsilon_N$  the average force is closer to the theoretical one but the oscillation increases. The error in the correct characterization of the contact force is due to the difficulty of reproducing a *Heavyside* function and also to the loss of energy in the transition of the dynamic wave between finite elements.

#### 5.8.4 Adaptive time step

In the examples of Section 5.8.2 and Section 5.8.3 there was a previous reference about the contact time. In most dynamic problems the contact time is unknown and there is no way to predict it. From it derives the problem of choosing a proper time step for the analysis. If the prediction of the contact step is too large the computed contact forces will be also too large. This is because the relative velocity of the bodies is high in the first contact and it produces a large reduction of the gap function (5.14). Consequently very large forces appear. From that, it starts a zig-zagging in the value of the contact force until it reaches equilibrium. This phenomenon is solved with a reduction of the time step. The choice of the time step is also very important to capture the impact waves and to get the proper physical response. However, a small time step is not desirable if the solution scheme is implicit. Hence, in order to solve this problem an algorithm for the reduction and adaptation of the time step has been developed.

The goal is to reduce the time step to the proper value when the first contact occurs and increase it again for the subsequent analysis. This is because the first penetration or reduction of the gap function, has to react with a force large enough to stop the contact surface but small enough not to push the surface away. When the prediction of the first force is correct, the contact will remain stable during the contact time.

The adaptation of the time step starts with a criterion to determine the suitable time step. This time increment is denoted as critical time step  $\Delta t_c$  for the contact force prediction. It is predicted prior to the iteration of the implicit solution method as

1. Compute the relative velocity of the contact surfaces in normal direction  $v_r$
2. The minimum time step allowed in the analysis  $\Delta t_m$  is defined
3. If the subdomains are approaching  $v_r > 0$ , the critical time step  $\Delta t_c$  is determined as:

$$\Delta t_c = \begin{cases} \frac{l_c}{v_r} & \rightarrow \text{if } \left(\frac{l_c}{\Delta t}\right) \leq v_r < \left(\frac{l_c}{\Delta t_m}\right) \\ \Delta t_m & \rightarrow \text{if } v_r \geq \left(\frac{l_c}{\Delta t_m}\right) \\ \Delta t & \rightarrow \text{otherwise} \end{cases} \quad (5.122)$$

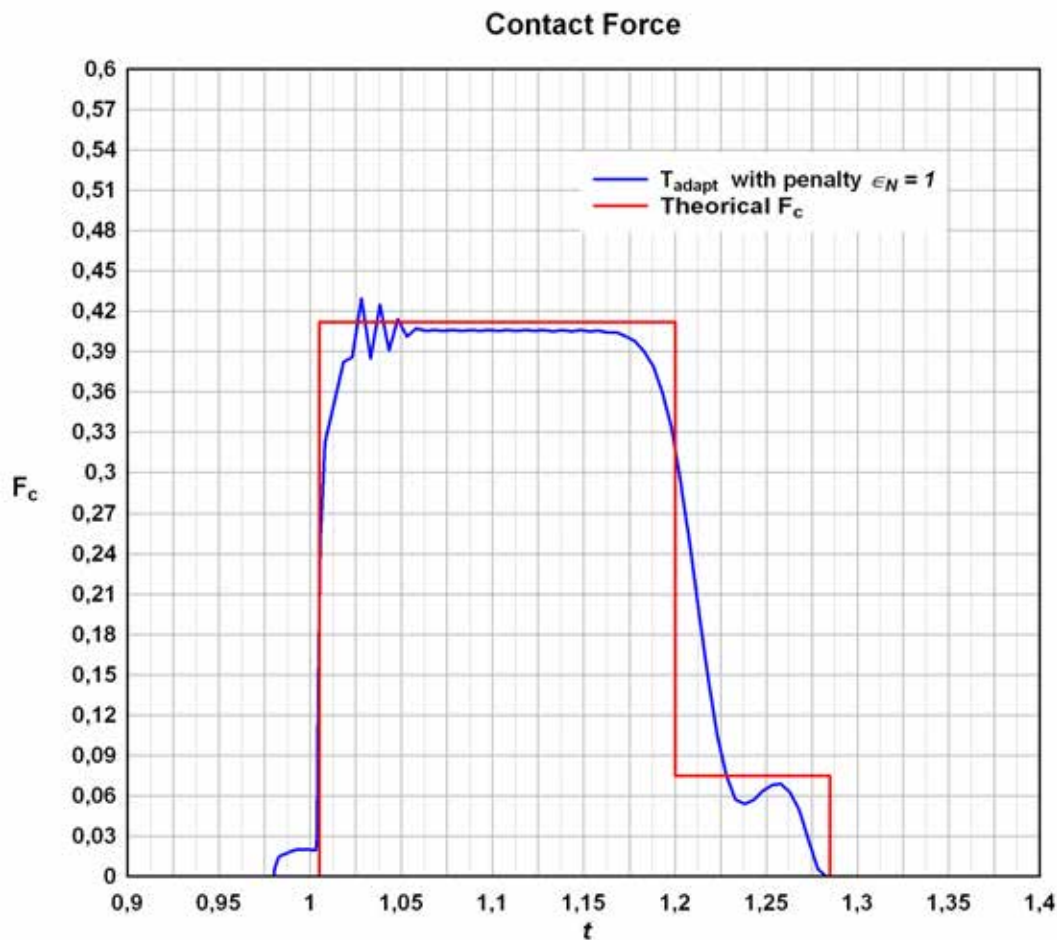
Box 5.3: Adaptive time step prediction

In Box 5.3,  $l_c$  is the characteristic length of the contact elements and  $\Delta t_c$  is the critical

time step. If the time step of the analysis  $\Delta t$  is larger than the critical one  $\Delta t > \Delta t_c$ ,  $\Delta t$  is replaced by  $\Delta t_c$ .

$\Delta t_c$  is initialized for every step and depends on the current *active set*. If the contact is deactivated or the relative velocity is reduced, the prediction recovers the initial time step  $\Delta t_c = \Delta t$ . When that happens, a progressive recovery of the time step is applied until the time step of the analysis is again  $\Delta t$ .

The impact problem of two rods (5.8.3) is analyzed again using the adaptive time stepping scheme. The result is presented in Figure 5.30.



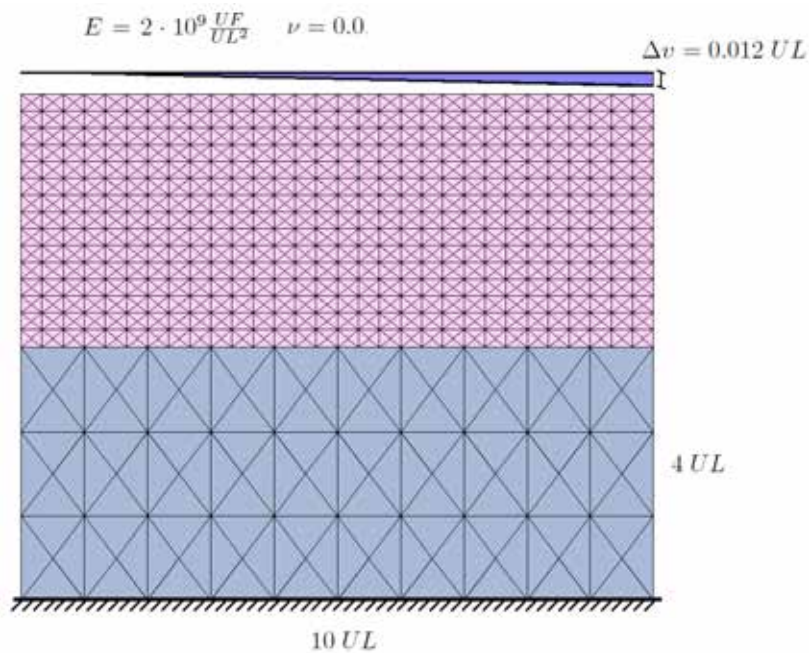
**Figure 5.30:** Evolution of the contact force for two colliding bars using an adaptive time stepping

It can be seen that the behavior is not much different as for the curve in Figure 5.29. However, a large time step is now used ( $\Delta t = 0.01$ ) and during the contact spacing the time step was automatically reduced to capture the contact response.

## 5.9 Numerical examples

### 5.9.1 Contact patch test

In this example the influence of the discretization between two contact domains is tested. A simple problem with two elastic blocks is considered. The meshes of the two blocks do not match in the interface, as is depicted in Figure 5.31. There is a upper block compressed over a foundation block. The problem deals with frictionless contact and small elastic deformations.



**Figure 5.31:** Patch test between two blocks. Different discretizations are used for each block.

The upper side of the upper block is clamped and a downward non-uniform linearly varying displacement field  $\Delta v$  is applied on it. The foundation is supported in the vertical direction at its lower edge. The problem conditions are depicted in Figure 5.31. A *Neo-Hookean* material is used for both blocks with  $E = 2 \cdot 10^9 \frac{UF}{UL^2}$  and  $\nu = 0.0$ .

The contact scheme converges without any problem. The vertical displacements are shown in Figure 5.32. There is a total continuity of the displacement field in the vertical direction. Stresses are shown in Figure 5.33. Some distortion on the block edges appear due to finite deformation effects. They are not a result of a lack of contact accuracy. To avoid these effects and reproduce the analytical linear elastic solution all nodes in the horizontal direction should be fixed.

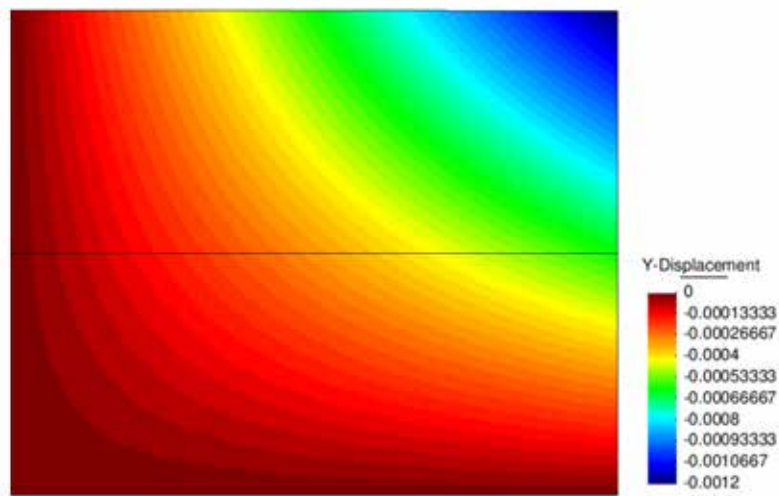


Figure 5.32: Displacement field on non-conforming meshes.

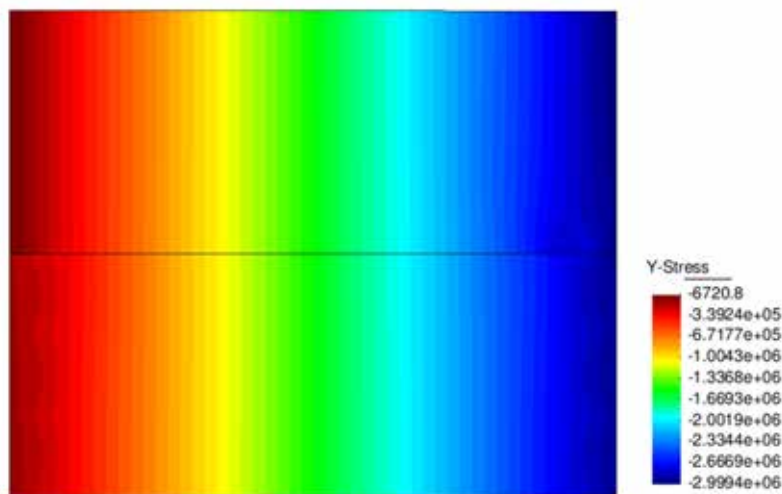
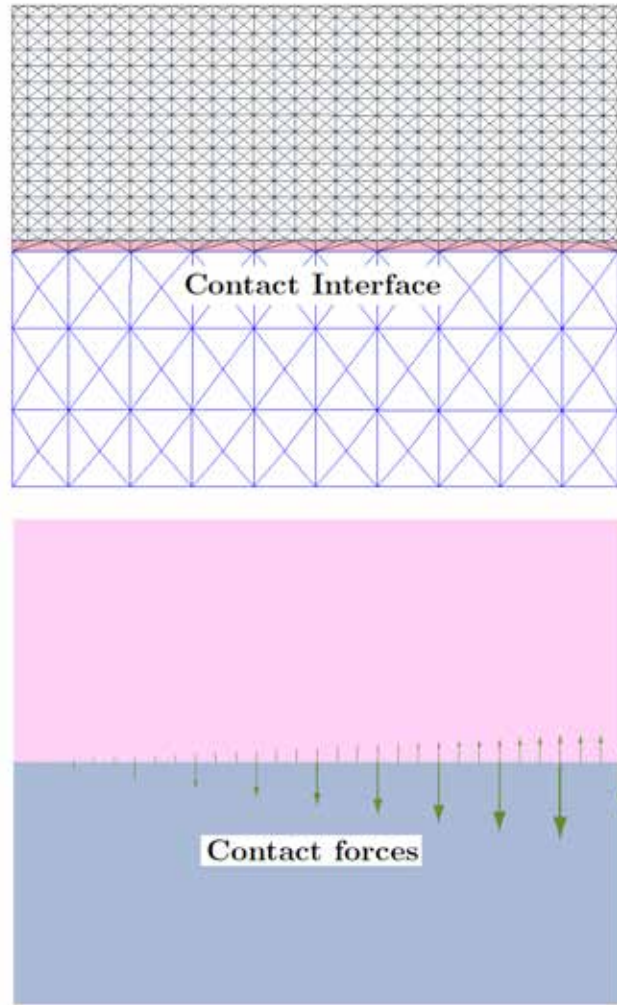


Figure 5.33: Mesh and vertical normal stress for a non-uniform displacement field.

Contact is detected and computed by means of triangular contact elements between the two bodies. The normal deformation of these elements yields the resultant normal contact forces on both contact surfaces. In Figure 5.34 the contact interface and the contact forces are shown.



**Figure 5.34:** Mesh and vertical normal stress for a non-uniform displacement field.

### 5.9.2 Hertzian contact problem

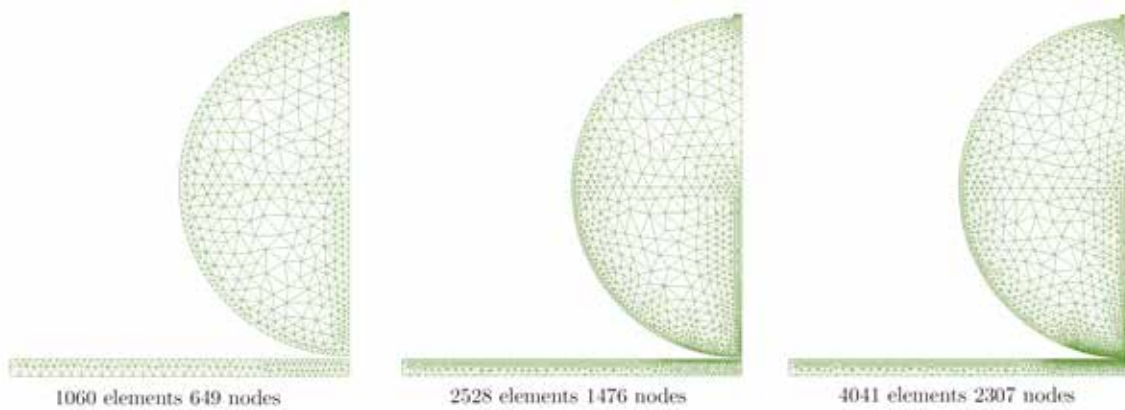
The force and contact area are calibrated in this well known Hertzian problem of an elastic cylinder contacting planar surface. For this problem an exact solution for the contact pressure can be obtained analytically as:

$$p_{max} = \sqrt{\frac{F_N}{\pi r} \frac{E}{(1 + \nu)(1 - \nu)}} \quad (5.123)$$

There is a similar expression for the evolution of the contact area given by the *Hertz's* equation for elastic deformation:

$$A_r = 2.9 \left[ F_N r \left( \frac{1}{E_1} + \frac{1}{E_2} \right) \right]^{\frac{2}{3}} \quad (5.124)$$

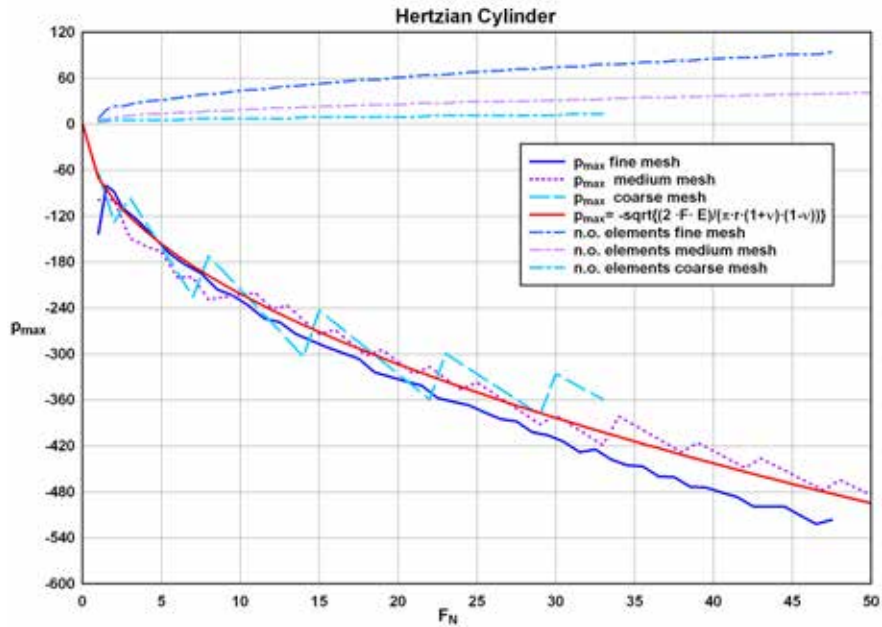
The parameters for the analysis are the following for the cylinder: Young modulus  $E = 7000$ , Poisson ratio  $\nu = 0.3$ . The cylinder has a radius of  $r = 1$ , and is loaded by an progressive load from  $F_N = 0$  to  $F_N = 100$ . The computed results are compared directly with the exact solution. To omit problems related to a point load in elasticity, the load is distributed over a small surface on top of the cylinder. In order to simulate the rigid surface the rigid domain in the finite element model is set with a Young modulus of  $E = 100000$  and Poisson ratio of  $\nu = 0.45$ . The initial meshes are depicted in Figure 5.35.



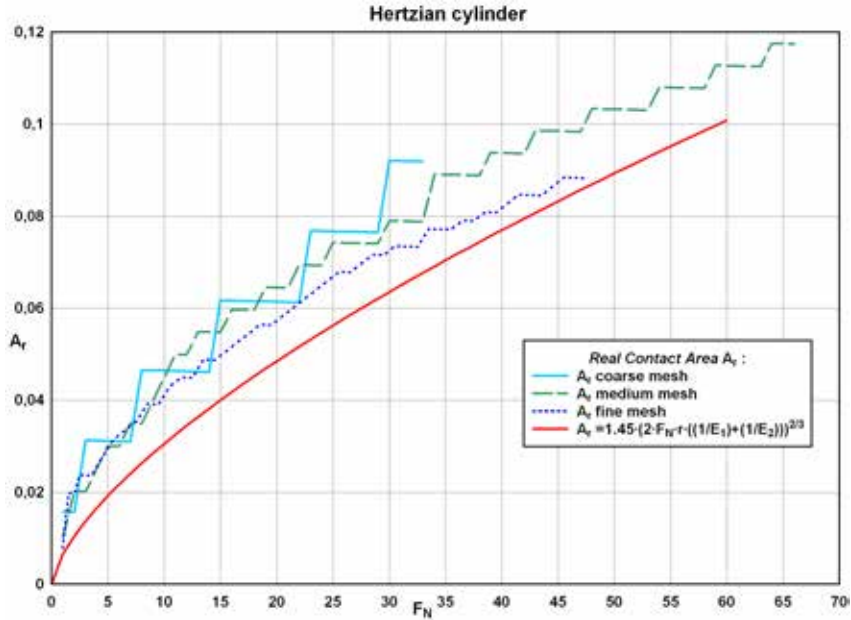
**Figure 5.35:** The problem is modelled with three different meshes that refine successively the contact zone.

The problem is symmetric, hence modeling half of the cylinder is enough to obtain the mechanical response. Figure 5.36 compares the evolution of the theoretical contact pressure with the numerical results for the three meshes. In Figure 5.37 the same comparison is made for the area of contact between the cylinder and the rigid base.

Some results from the force and strain distribution on the cylinder and the rigid base are shown in Figure 5.38.

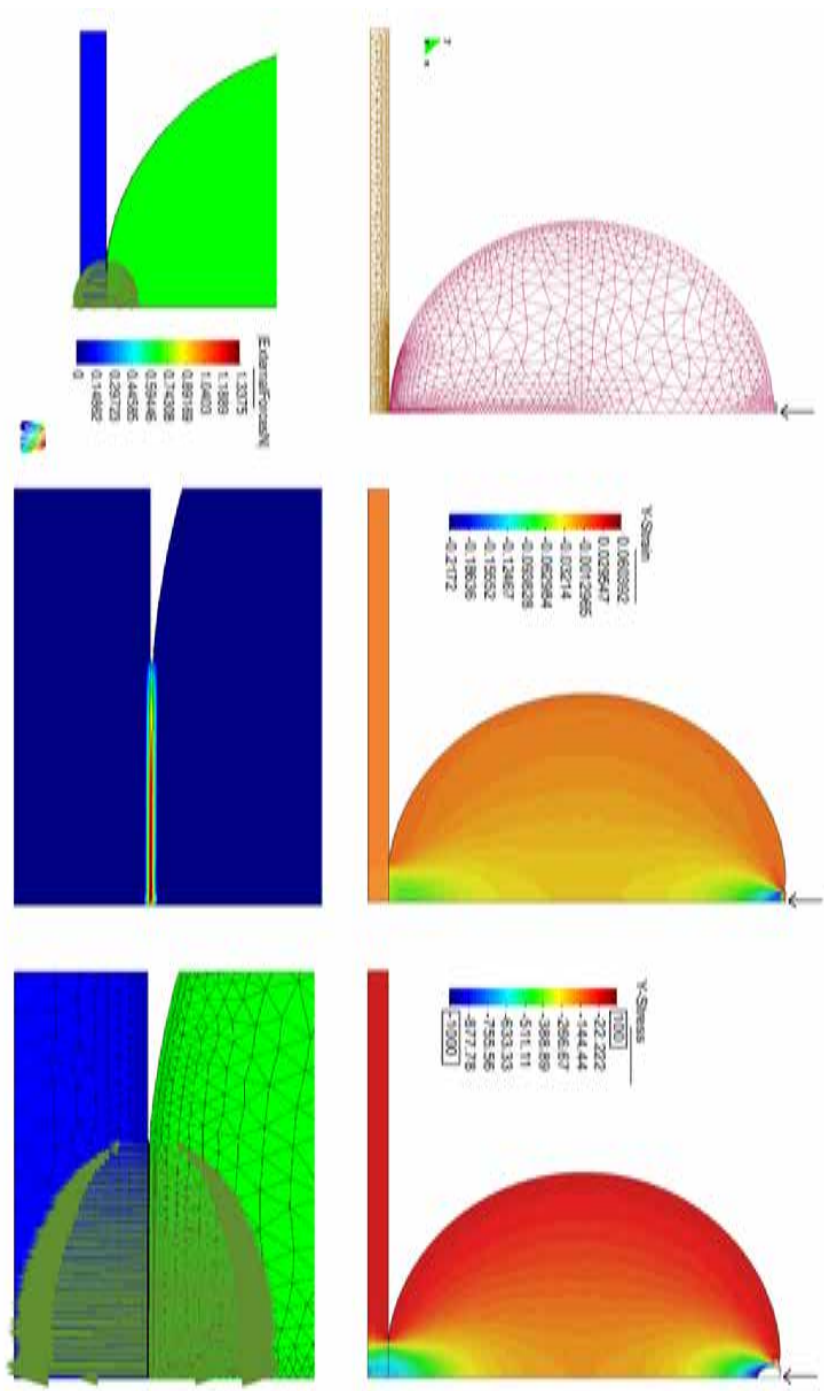


**Figure 5.36:** Evolution of the contact pressure and number of contact elements in the contact zone for the 2D *Hertzian* cylinder.



**Figure 5.37:** Evolution of the real contact area for the 2D *Hertzian* cylinder when the normal force increases. The model is considered as a hardly polished surface.



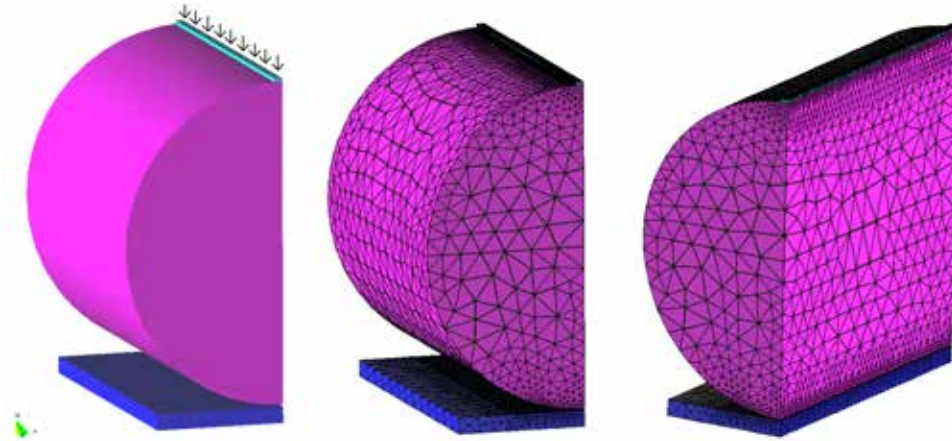


**Figure 5.38:** Numerical results on the 2D finer analyzed mesh. Strains and contact forces and distribution of the normal contact forces.

### 3D Hertzian cylinder

The PFEM formulation has been also applied for three-dimensional problems. In this problem the force and contact area are calibrated in 3D using the Hertzian problem.

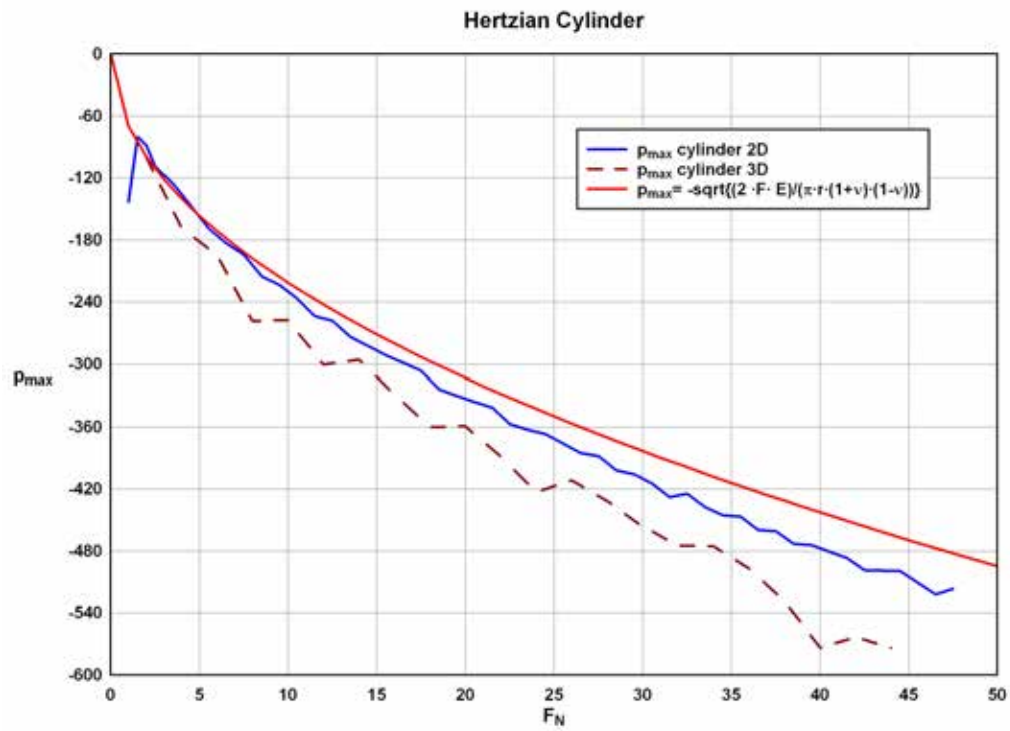
The problem is symmetric. Hence modeling half of the cylinder is enough for the mechanical response. The parameters for the analysis are the same as in the 2D case. The initial finite element model can be seen in Figure 5.39.



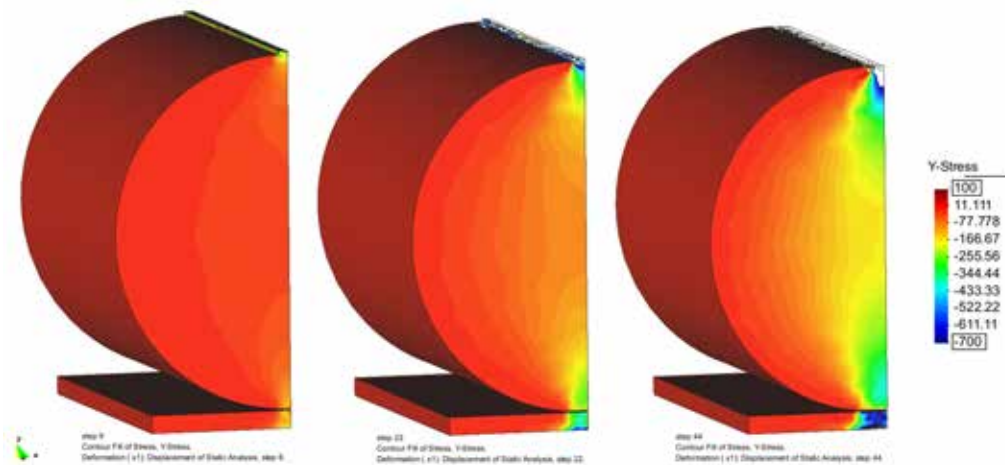
**Figure 5.39:** Model of a half cylinder in 3D.

The evolution of the contact pressure is monitored. In Figure 5.40 the function for the theoretical contact pressure is compared with the numerical results. In Figure 5.42 the same comparison is made for the area of contact between the cylinder and the rigid base. The approximation of the real contact area  $A_r$  is much better when a 3D mesh is used.

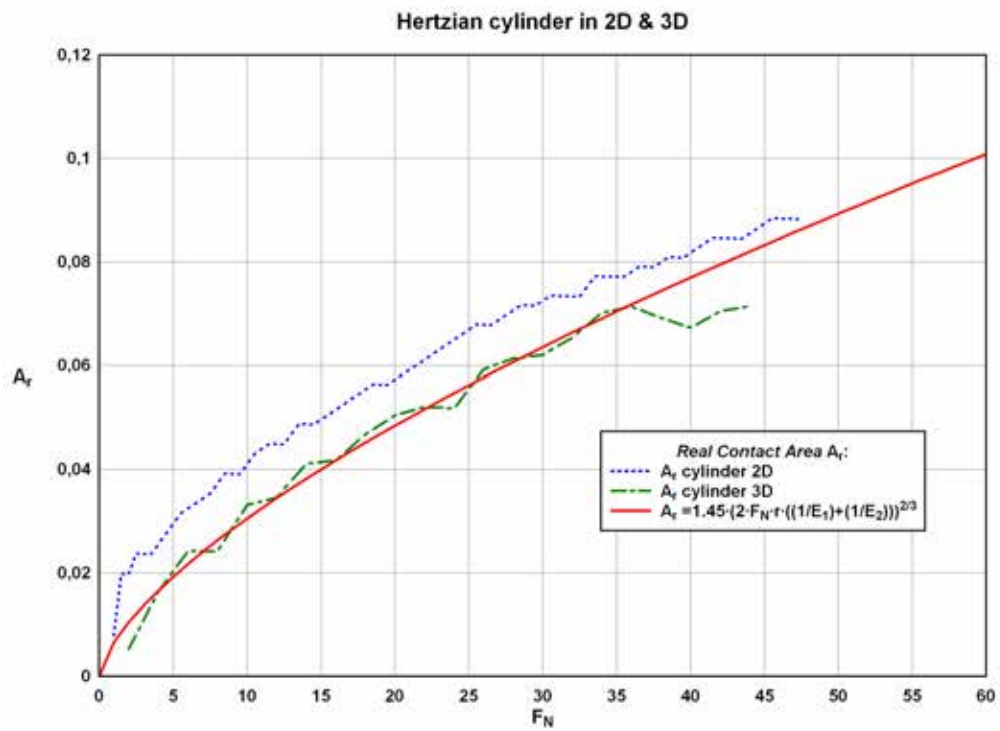
Some results from force and strain distribution on the cylinder and the rigid base are shown in Figure 5.41, Figures 5.43 and 5.44.



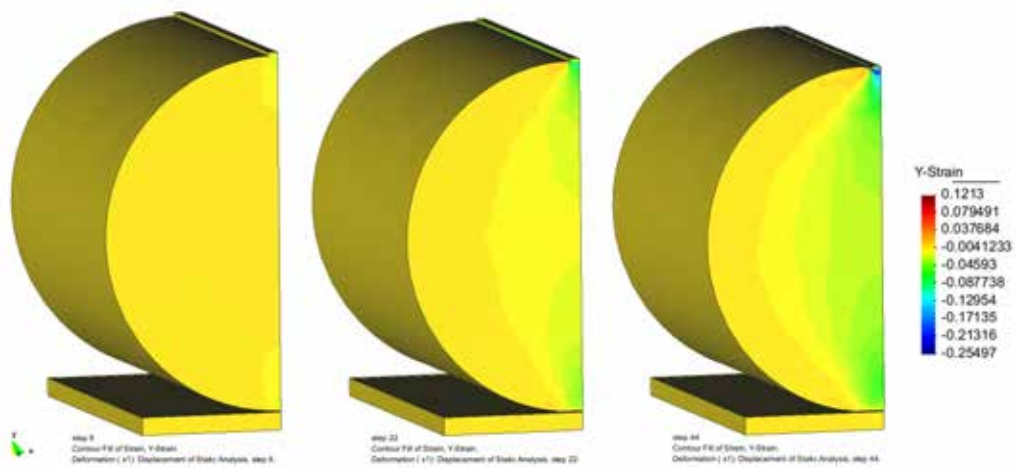
**Figure 5.40:** Evolution of the contact pressure on contact surfaces for the 2D and 3D *Hertzian* cylinders.



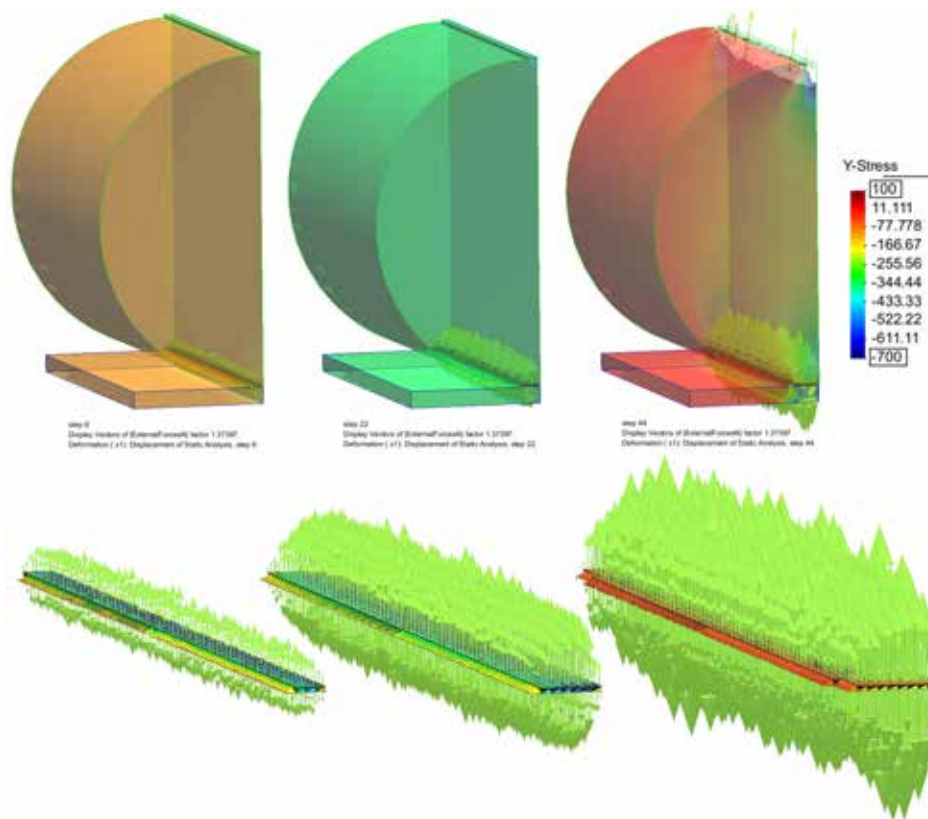
**Figure 5.41:** Numerical results 3D cylinder. Stress distribution in the vertical direction.



**Figure 5.42:** Evolution of the real contact area for the 2D and 3D *Hertzian* cylinders when the normal force increases.



**Figure 5.43:** Numerical results for 3D cylinder. Strain distribution in the vertical direction.



**Figure 5.44:** Numerical results for 3D cylinder. Contact normal forces on contact surfaces.

### 3D Hertzian sphere

Finally another example in 3D for a different geometry is presented. In this case the *Hertzian sphere* is modelled in 3 dimensions. The purpose is to compare the results with the *Hertzian cylinder* and to prove that the contact in the PFEM can be applied to any geometry.

The problem has two symmetry planes, but the model is only simplified using one symmetry plane. Hence half of the sphere is analyzed to get the mechanical response. The analysis parameters are the same as in the previous cases. The initial finite element model can be seen in Figure 5.45.

The evolution of the contact pressure is monitored. In Figure 5.46 the theoretical contact pressure is compared with the numerical results of the cylinder and the sphere. It can be seen that the pressure on the *Hertzian sphere* is much larger. In Figure 5.47 the same comparison is made for the area of contact between the sphere and the rigid base. The real contact area  $A_r$  for the sphere is smaller than for the cylinder. This is the reason for having larger pressures on the sphere for the same applied force.

Some results for the force and strain distribution on the sphere are shown in Figures 5.48, 5.49 and 5.50.

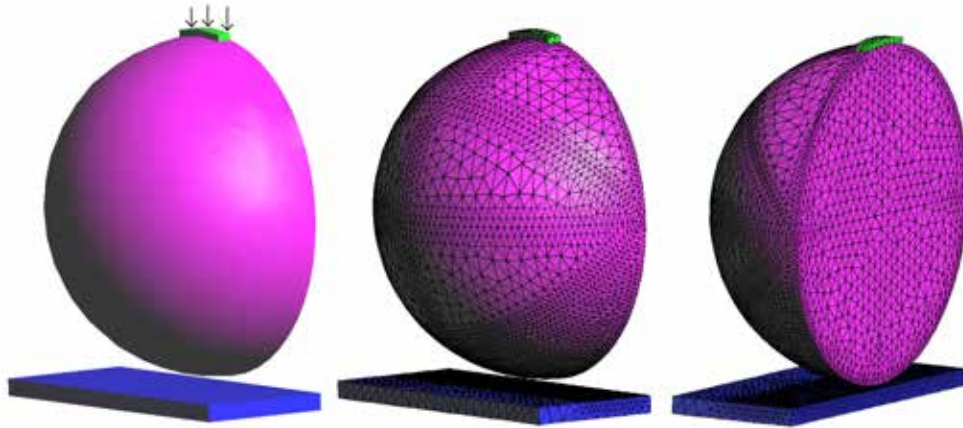


Figure 5.45: Model of a half sphere in 3D.

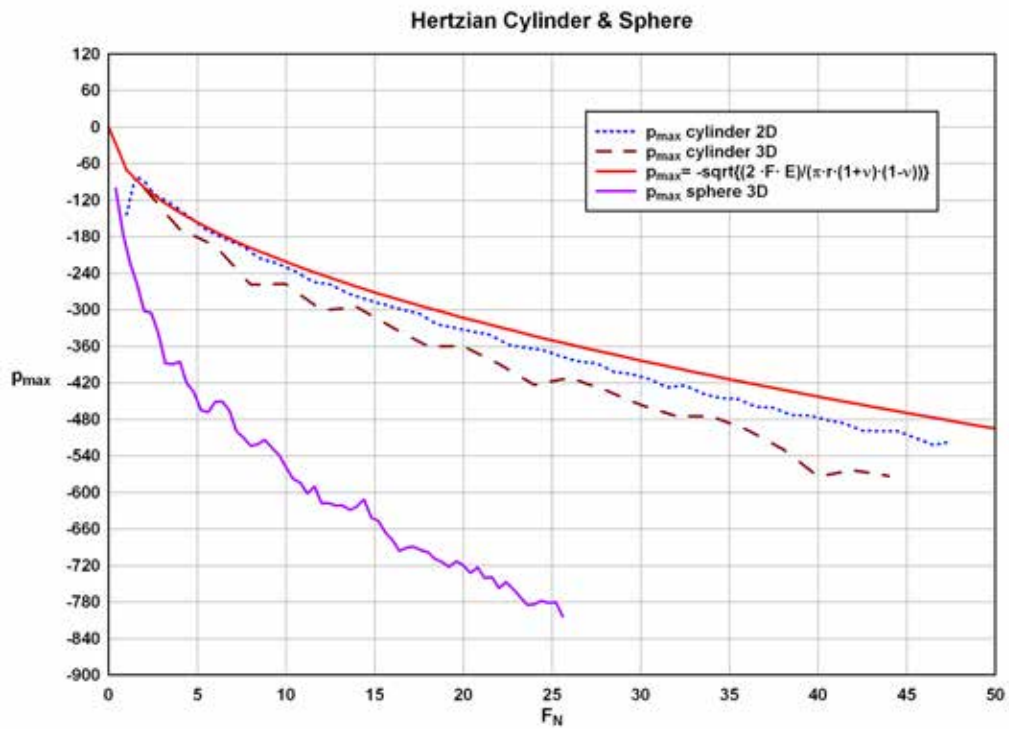
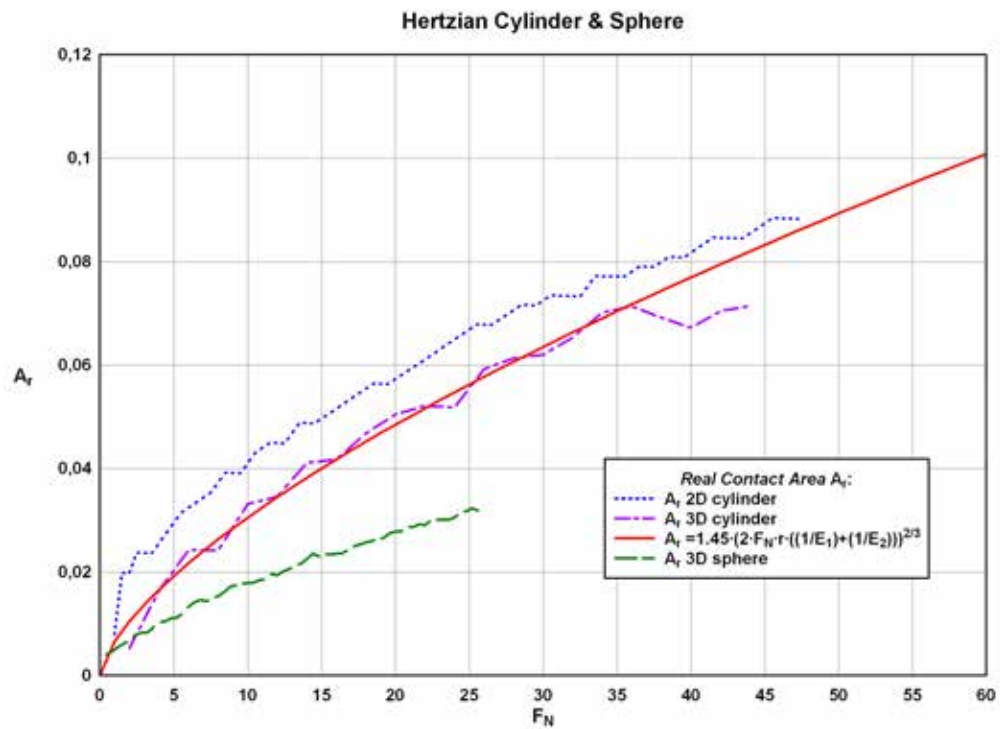
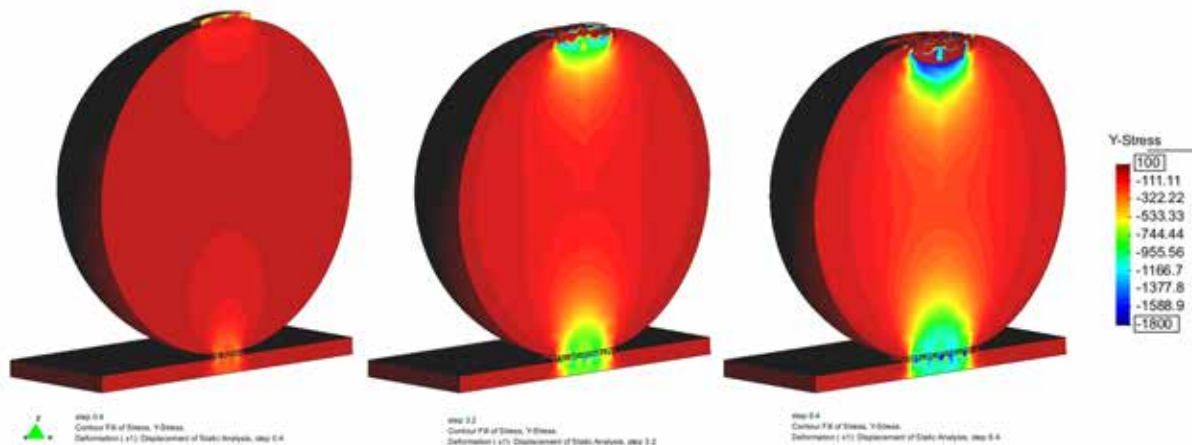


Figure 5.46: Evolution of the contact pressure on contact surfaces. Comparison between the *Hertzian* sphere and the *Hertzian* cylinder.



**Figure 5.47:** Evolution of the real contact area comparing the *Hertzian* sphere and the *Hertzian* cylinder.



**Figure 5.48:** Numerical results for 3D sphere. Stress distribution in the vertical direction.

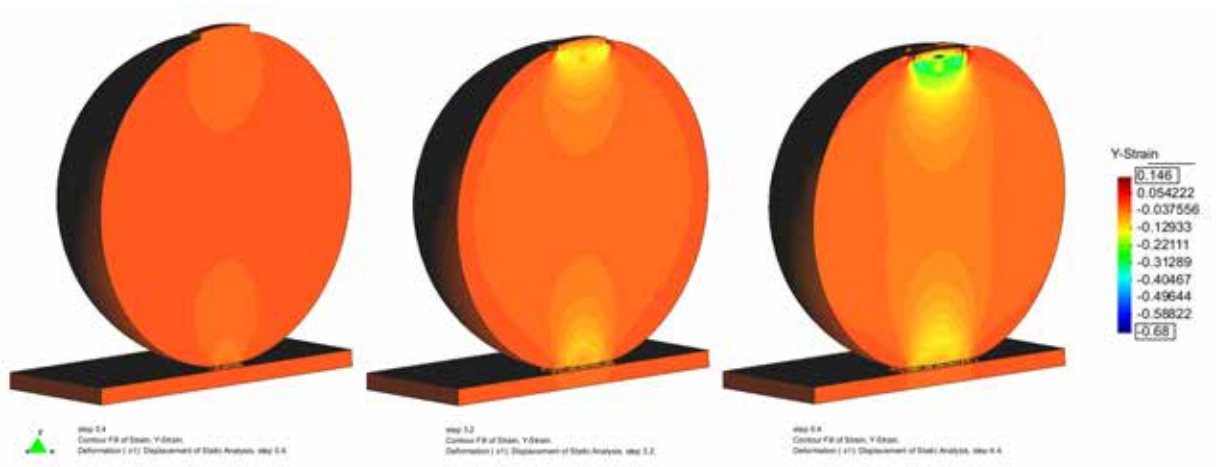


Figure 5.49: Numerical results for 3D sphere. Strain distribution in the vertical direction.

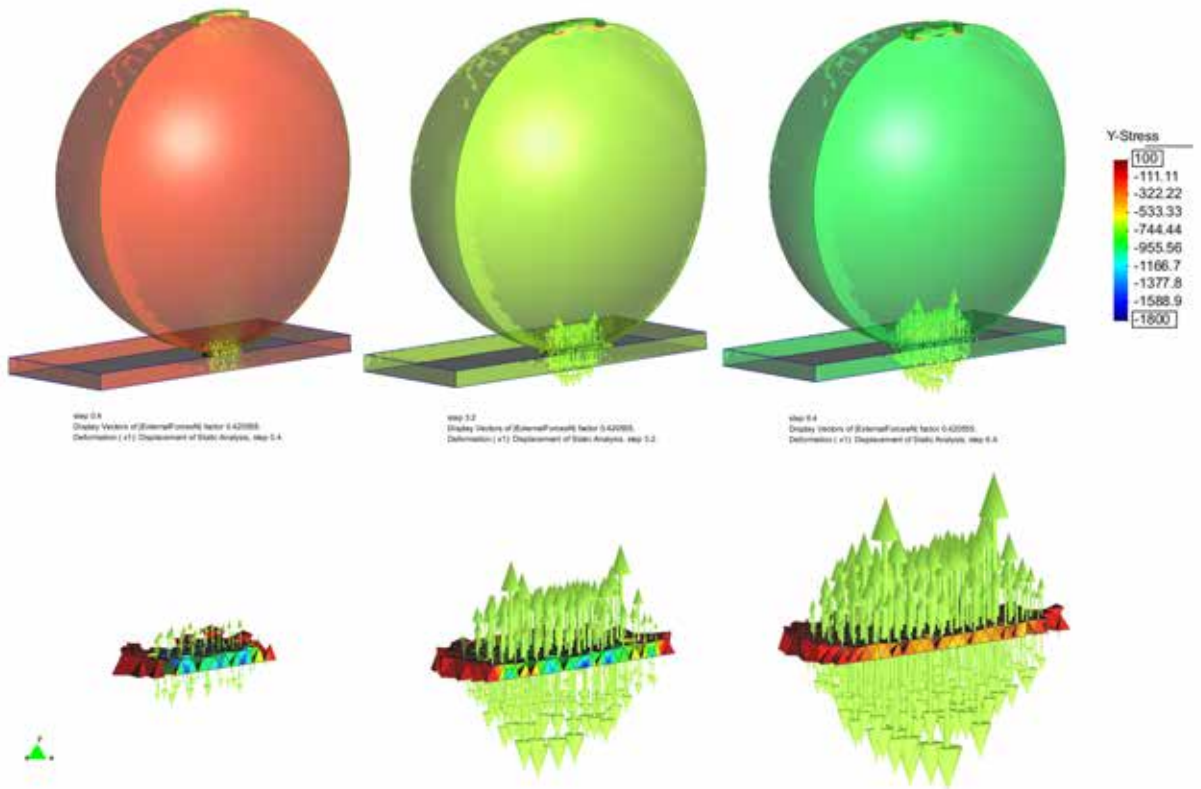
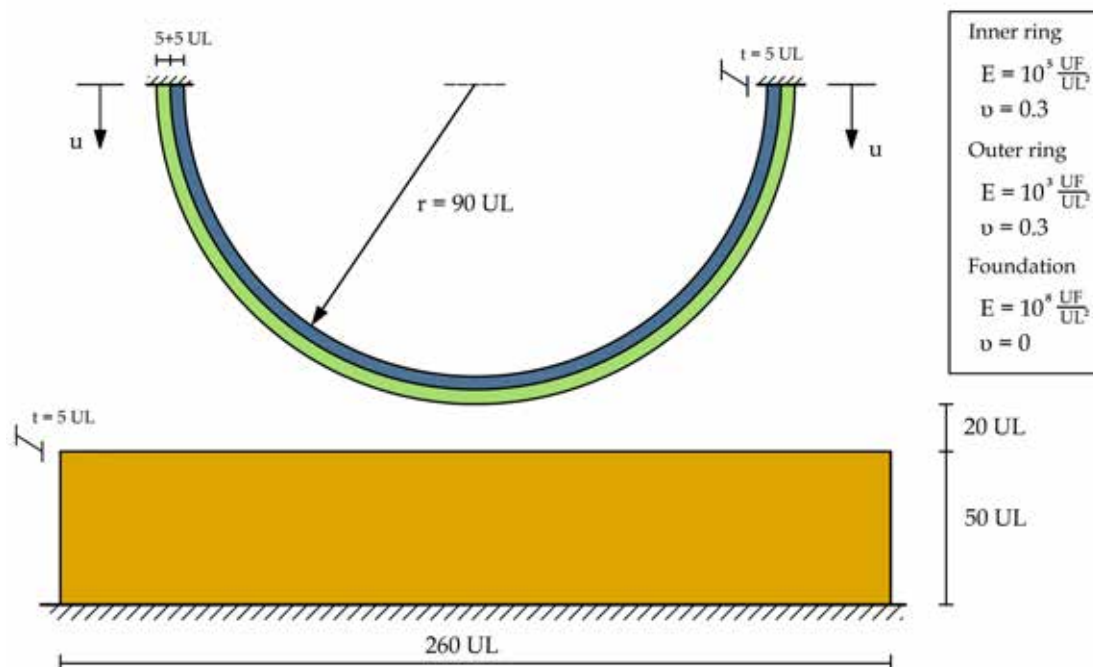


Figure 5.50: Numerical results 3D sphere. Contact normal forces on contact surfaces.



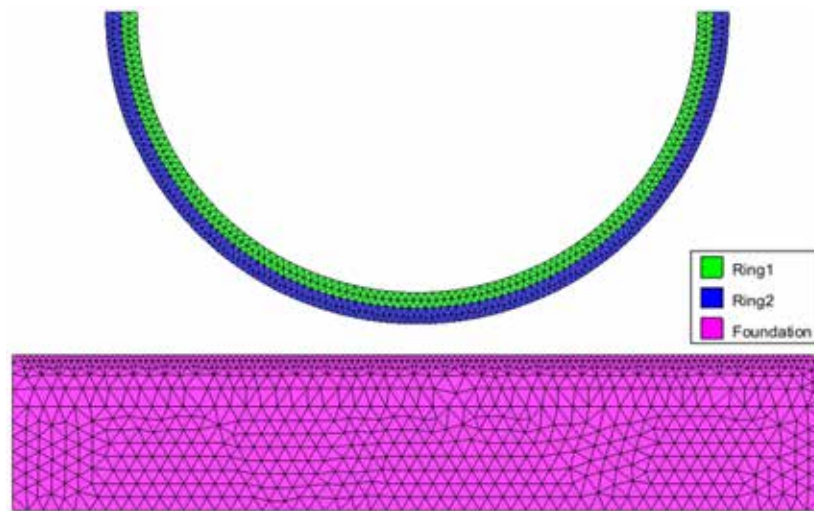
### 5.9.3 Contact between an elastic ring and a foundation

The contact between an elastic ring undergoing finite deformations and a rigid surface is considered in this example. The elastic ring consists of an outer and inner ring of the same thickness with different materials. The inner ring is 100 times stiffer than the outer ring. The foundation is almost rigid. Both bodies are  $5 UL$  depth. The values of the material parameters as well as the geometry are given in Figure 5.51. A *Neo-Hookean* material is assumed.



**Figure 5.51:** Compounded elastic ring in contact with a foundation.

A vertical displacement is applied to the ring at its top ends. A larger displacement to initialize the contact and a smaller displacement until the ring has a lift-off in the middle. The problem is discretized with a non uniform mesh. On the contact surfaces a regular mesh size has been imposed in order a better characterization of the contact forces. The contact is considered frictionless. The mesh and the material distribution used are depicted in Figure 5.52.



**Figure 5.52:** Mesh used for the discretization of the elastic ring and the foundation.

The problem is characterized by the change of the contact area during the loading process. The contact area changes, due to the finite deformations, from one to two spots. In the final state, the ring has a lift-off in the middle. This is shown in Figure 5.53, where the stresses for the ring and for the foundation are also depicted.

The force distribution and the partition of the contact areas due to finite deformations are presented in Figure 5.54

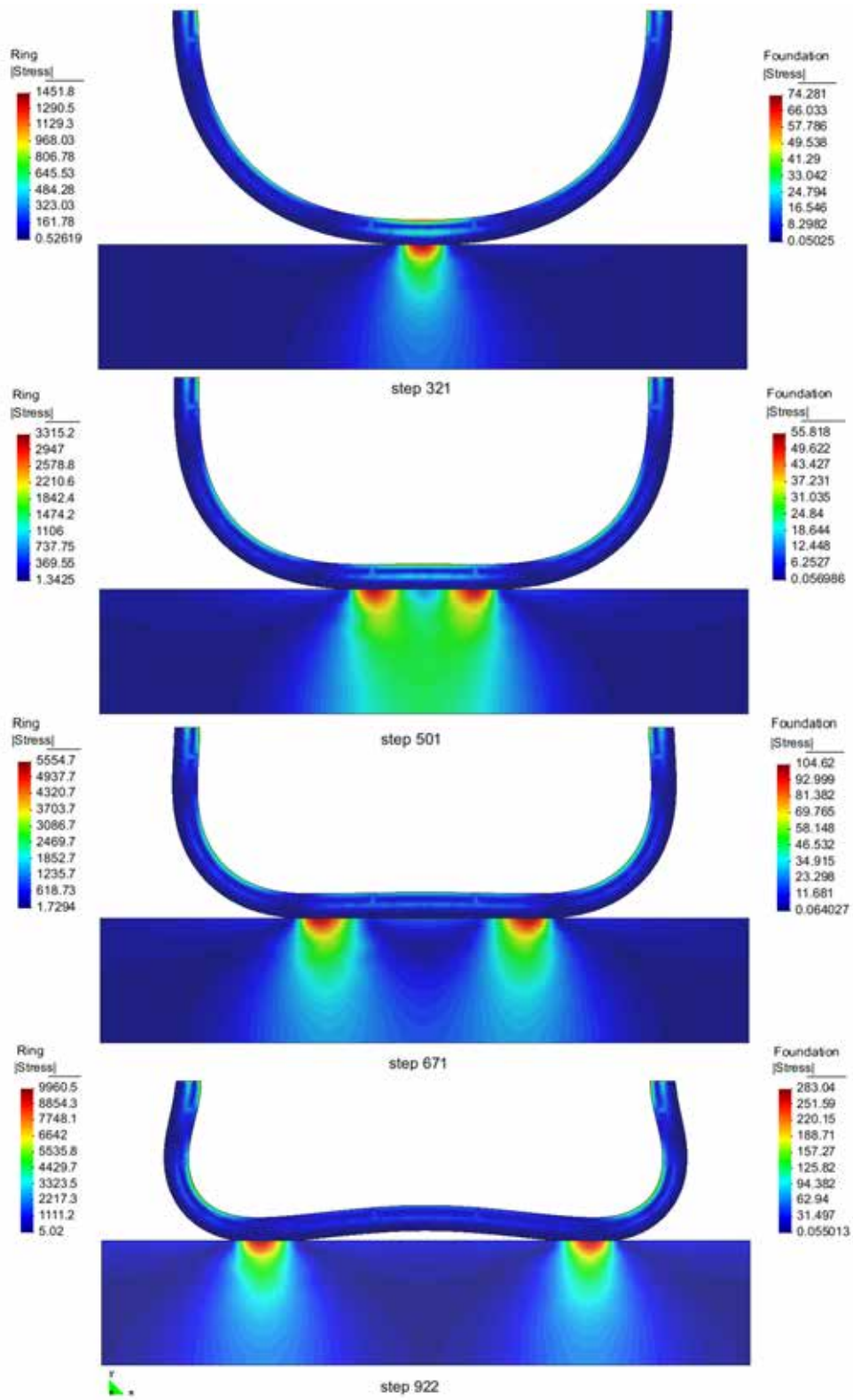


Figure 5.53: Vertical stress distribution and deformation state.

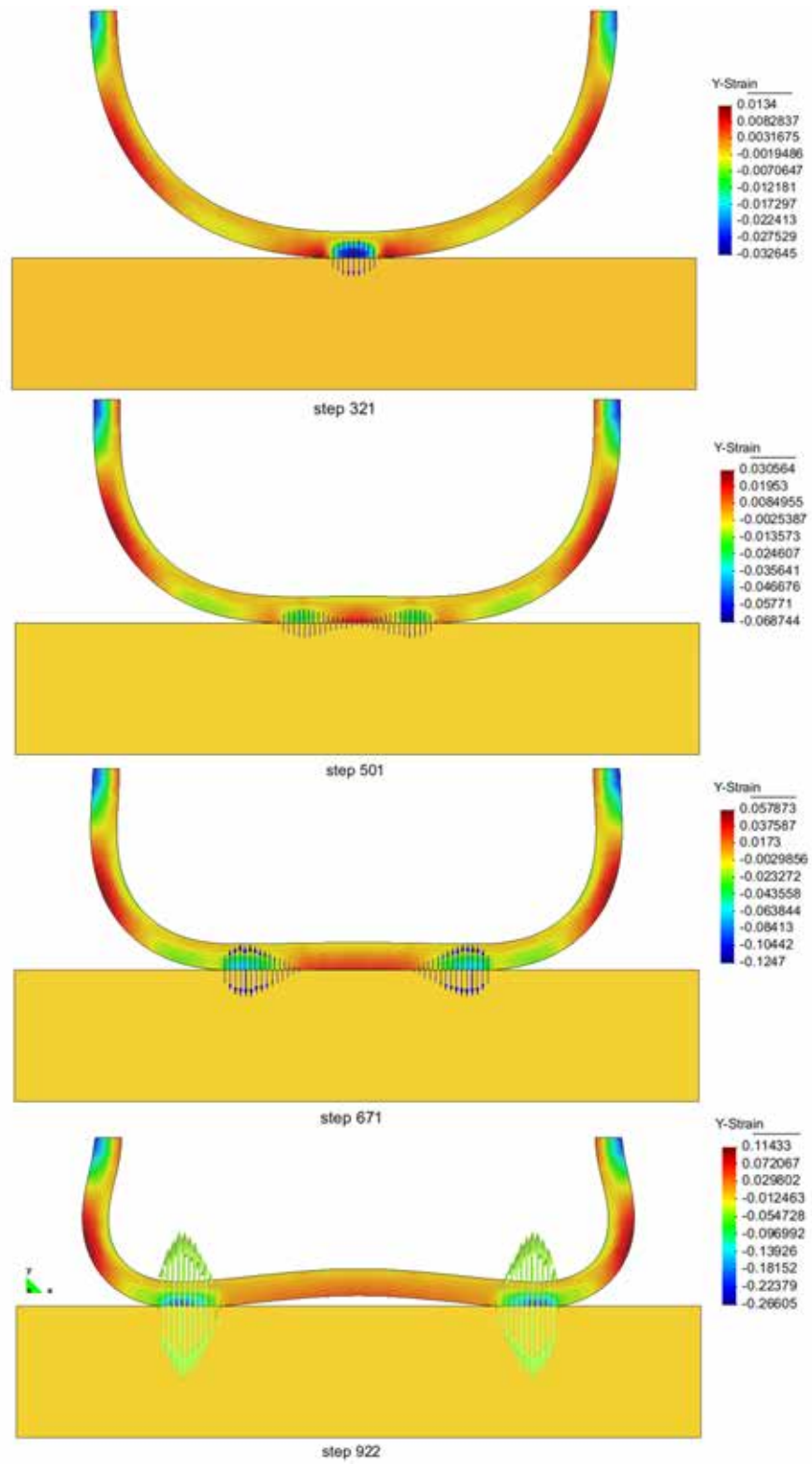
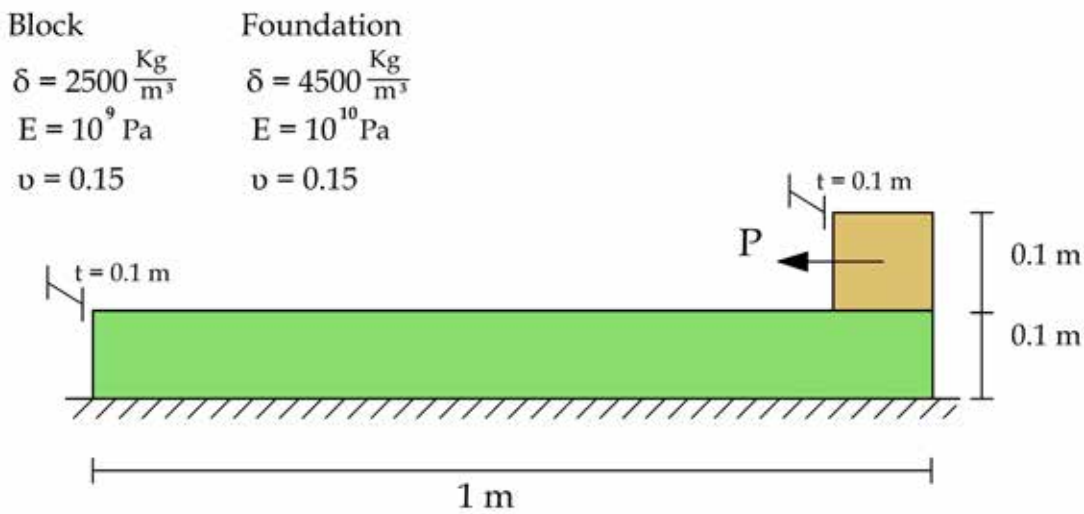


Figure 5.54: Vertical strain, contact force distribution and deformation state.

### 5.9.4 Friction problem

Frictional contact has not been considered in the previous examples. In frictional contact the interest is to capture the stick-slip condition of the tangential behavior. To model this phenomenon a regularized *Coulomb* law is used (see Section 5.3.3). A simple example has been designed for the measurement of friction accuracy. The problem consist on a block resting on a plane surface. A horizontal load is applied progressively on the block until it starts to slide. The problem is considered dynamic. The values of the material parameters and the geometry of the problem are depicted in Figure 5.55. A *Neo-Hookean* material is assumed.

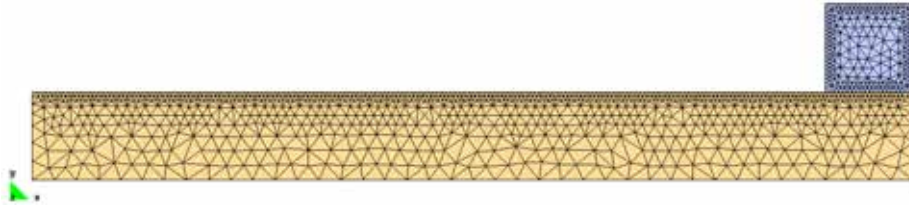


**Figure 5.55:** Elastic block on a elastic foundation.

For the friction between solid surfaces  $\mu_D = 0.38$  and  $\mu_S = 0.40$  are considered, which are the dynamic and static friction coefficient respectively. Equation (5.27) is used for the determination of the friction coefficient depending on the relative sliding velocity. The constitutive parameter that describes how fast the static coefficient approaches the dynamic one has the value of  $c = 0.1$ .

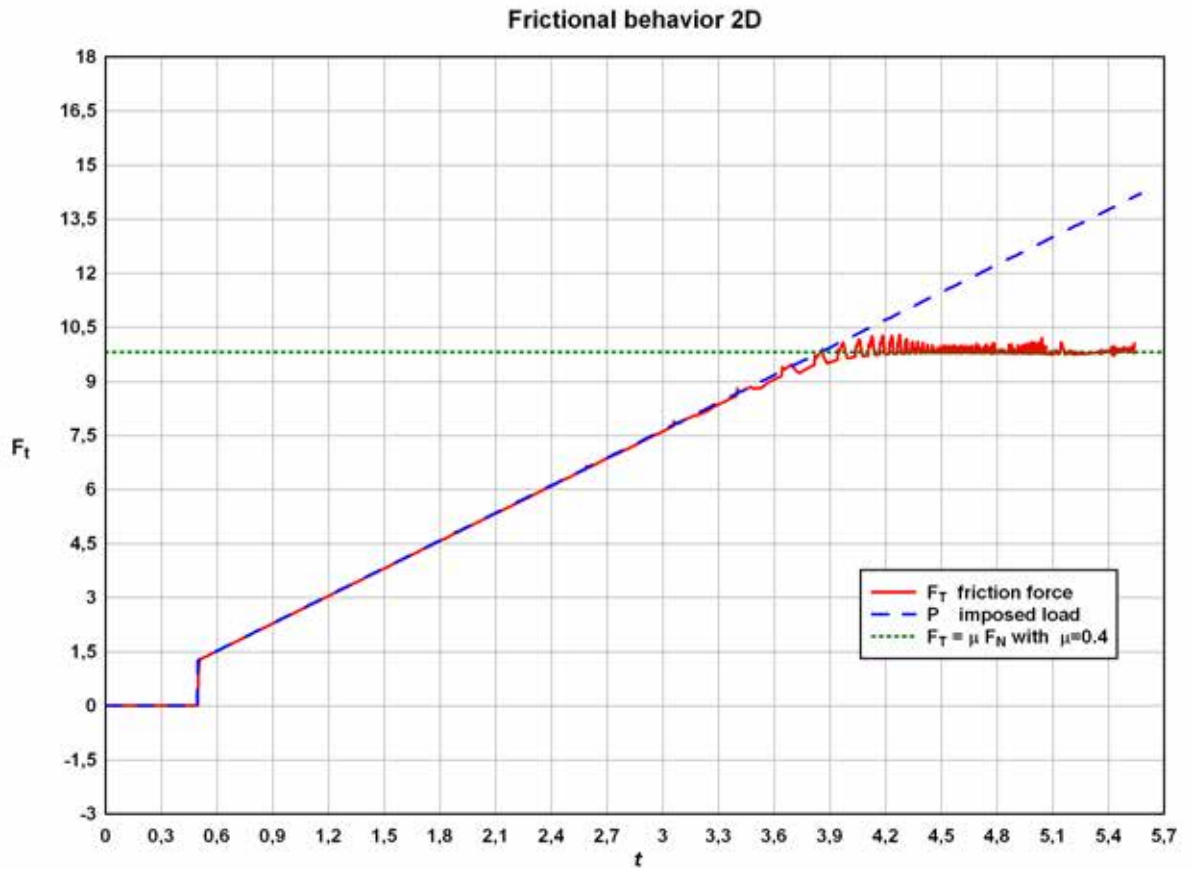
The regularization of the stick-slip behavior of *Coulomb's* law is performed by equation (5.29). For this equation the scalar parameter  $\varepsilon$  denotes the regularization variable and is taken as  $\varepsilon = 0.01$ .

In order to prevent shock loading and to keep the stick state of the block under control, firstly the block is left under self-weight load over the foundation during 50 time steps (0.5 seconds). After these 50 time steps of  $\Delta t = 0.01s$  a horizontal load  $\mathbf{P}$  is applied to all nodes of the block. The mesh used is shown in Figure 5.56.



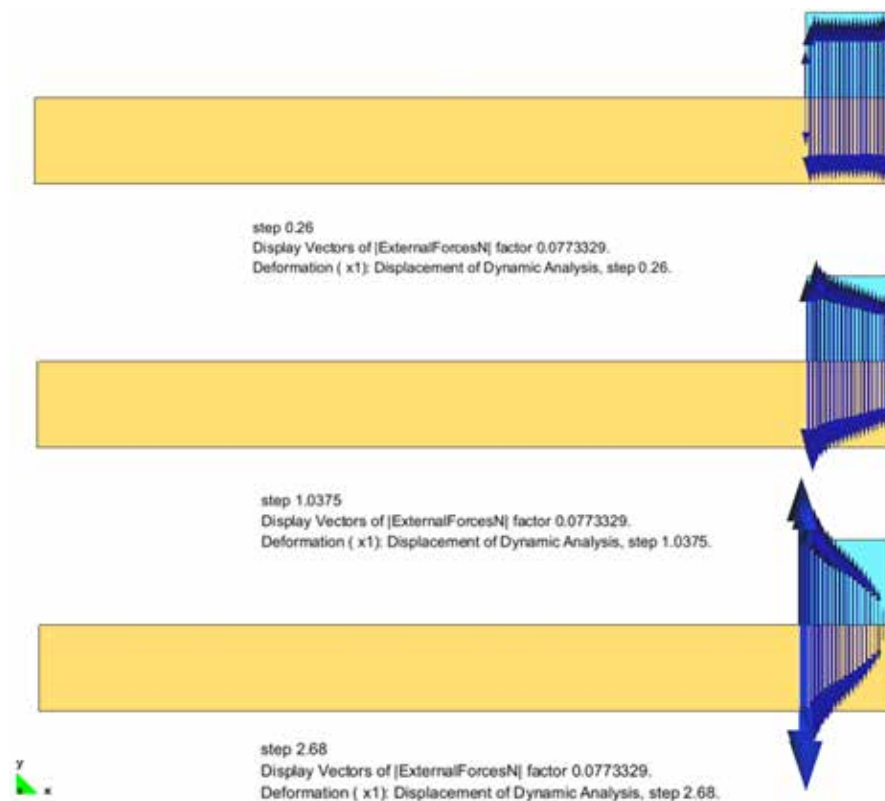
**Figure 5.56:** Unstructured mesh used for the discretization of the two elastic domains.

The imposed load  $\mathbf{P}$  is incremented progressively until the static tangential force is reached and the block starts to slide. The results for the evolution of the tangential forces are presented in Figure 5.57.



**Figure 5.57:** Evolution of the friction force in the two-dimensional case.

The results show how the block remains stick with the equilibrium of the tangential



**Figure 5.58:** Evolution of the normal contact forces when the progressive tangential load is applied.

forces. When it starts to slide the applied force  $\mathbf{P}$  and the friction force  $\mathbf{F}_T$  do not coincide anymore. Some oscillation on the friction forces is observed and it is explained because of the dynamic contact effects.

A small displacement is observed in the numerical results when the first load is applied and friction is activated. After that the block remains stick. Figure 5.58 shows the evolution of the normal contact forces from stick to slip. Figure 5.59 extends the contact zone showing the contact force vectors in the normal and tangential directions. Note that the shrunk geometry of the boundaries is used in Figure 5.59 in order to display the contact mesh.

A last picture is presented in Figure 5.60 showing the stress field on bodies with the corresponding contact forces.

Friction is captured correctly with the numerical algorithms presented in Section 5.3.3. The drawback is that the stick-slip motion is described only in an approximate way with the regularized laws. Therefore the parameters for the regularization function must be chosen with care in order to predict real frictional behavior.

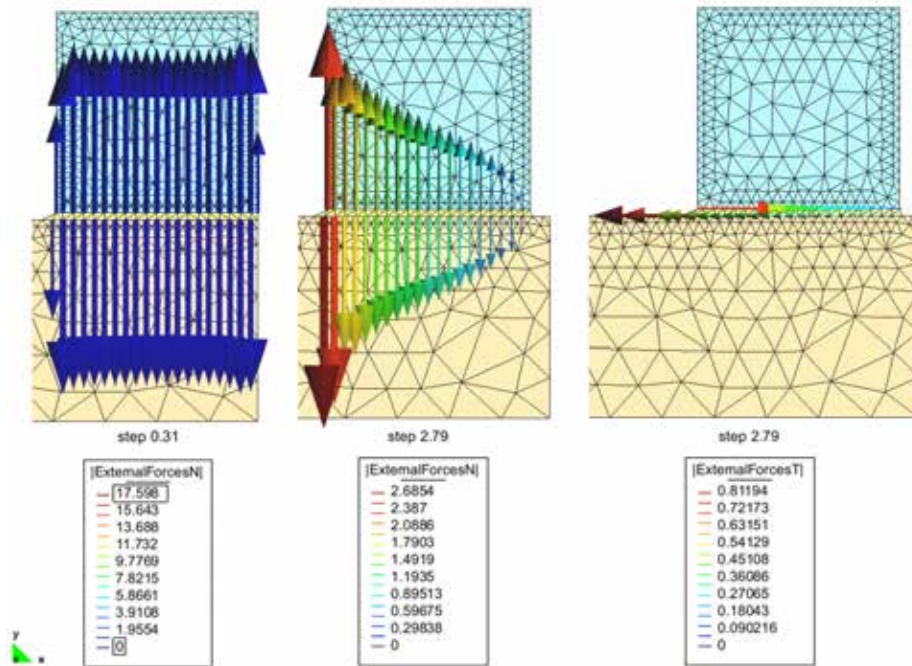


Figure 5.59: Shrinkage of the contact boundaries, contact elements and contact forces.

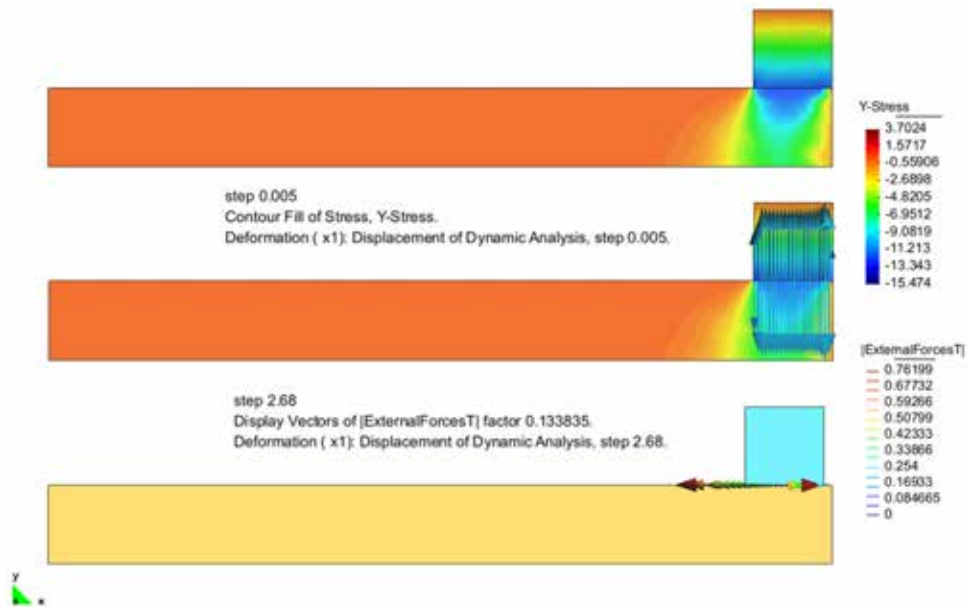
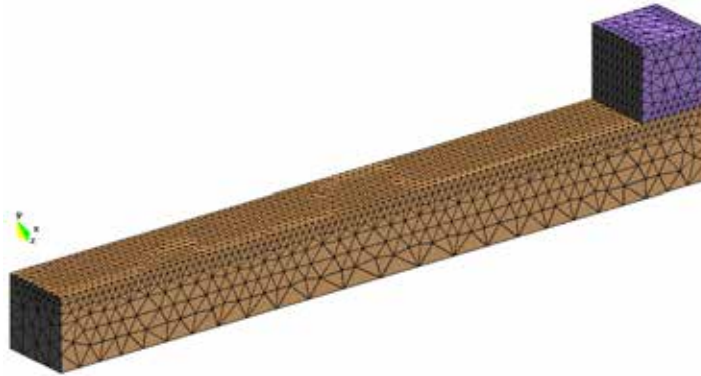


Figure 5.60: Vertical stresses and forces with the block resting and tangential forces when the imposed load is applied.

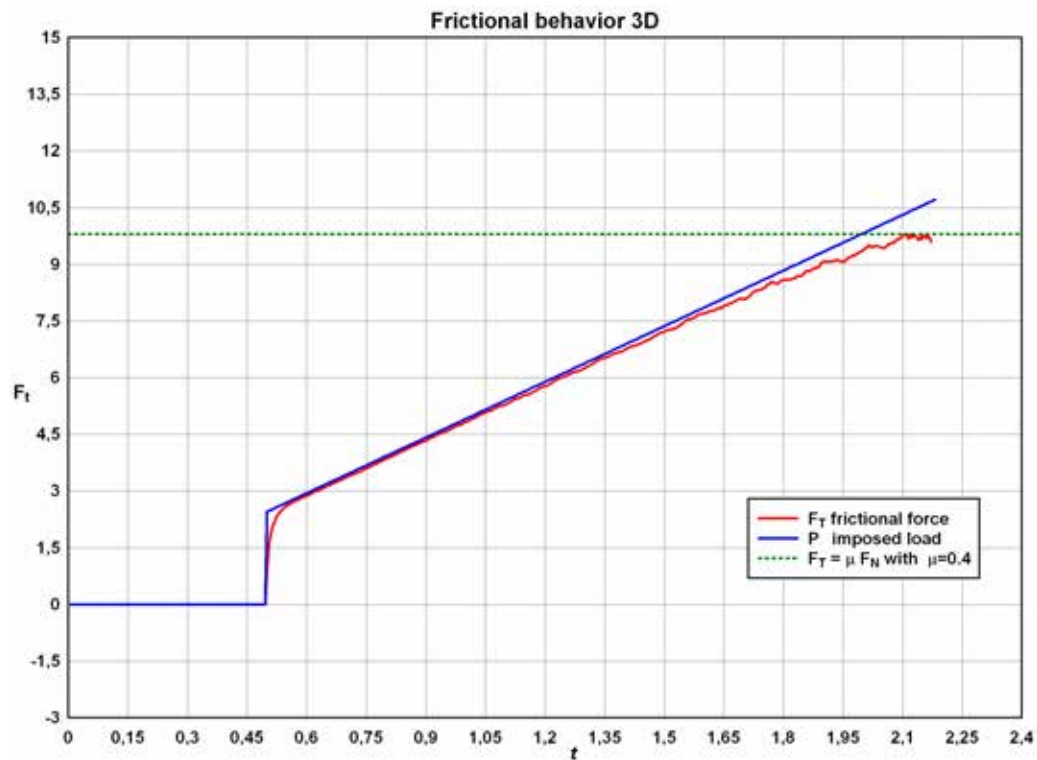


### Friction problem in 3D



**Figure 5.61:** Unstructured mesh used for the discretization of the two elastic domains of Figure 5.55 in 3D.

Last example was analyzed in the 2D case. In order to check the frictional contact for problems in three-dimensions, the same example with the same geometry and conditions, is computed with a 3D model. The used mesh is shown in Figure 5.61

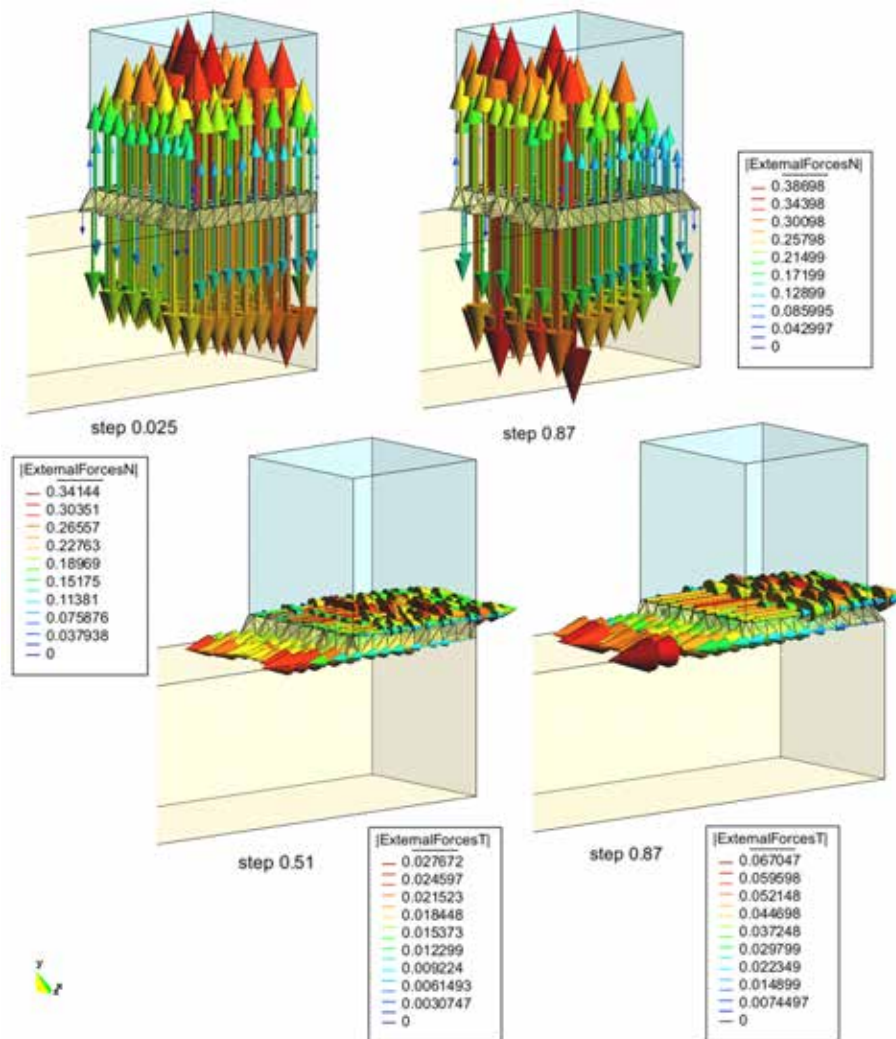


**Figure 5.62:** Evolution of the friction force in the 3D case.

As expected the results are very similar to the ones presented for the 2D case. After a period of resting, a load  $\mathbf{P}$  is imposed and incremented progressively until the static tangential force is reached and the block starts to slide. For this case the applied force was a little bit larger than for the 2D case. The results for the evolution of the tangential forces are presented in Figure 5.62.

Figure 5.64 shows the evolution of the normal contact forces from stick to slip and the resulting stress field. Figure 5.63 extends the contact zone with the shrunk geometry, showing the contact force vectors in the normal and tangential directions.

The numerical results for the 2D and 3D cases are very similar. The observed behavior for the frictional contact is the same.



**Figure 5.63:** Shrinkage of the contact boundaries, contact elements and contact forces.

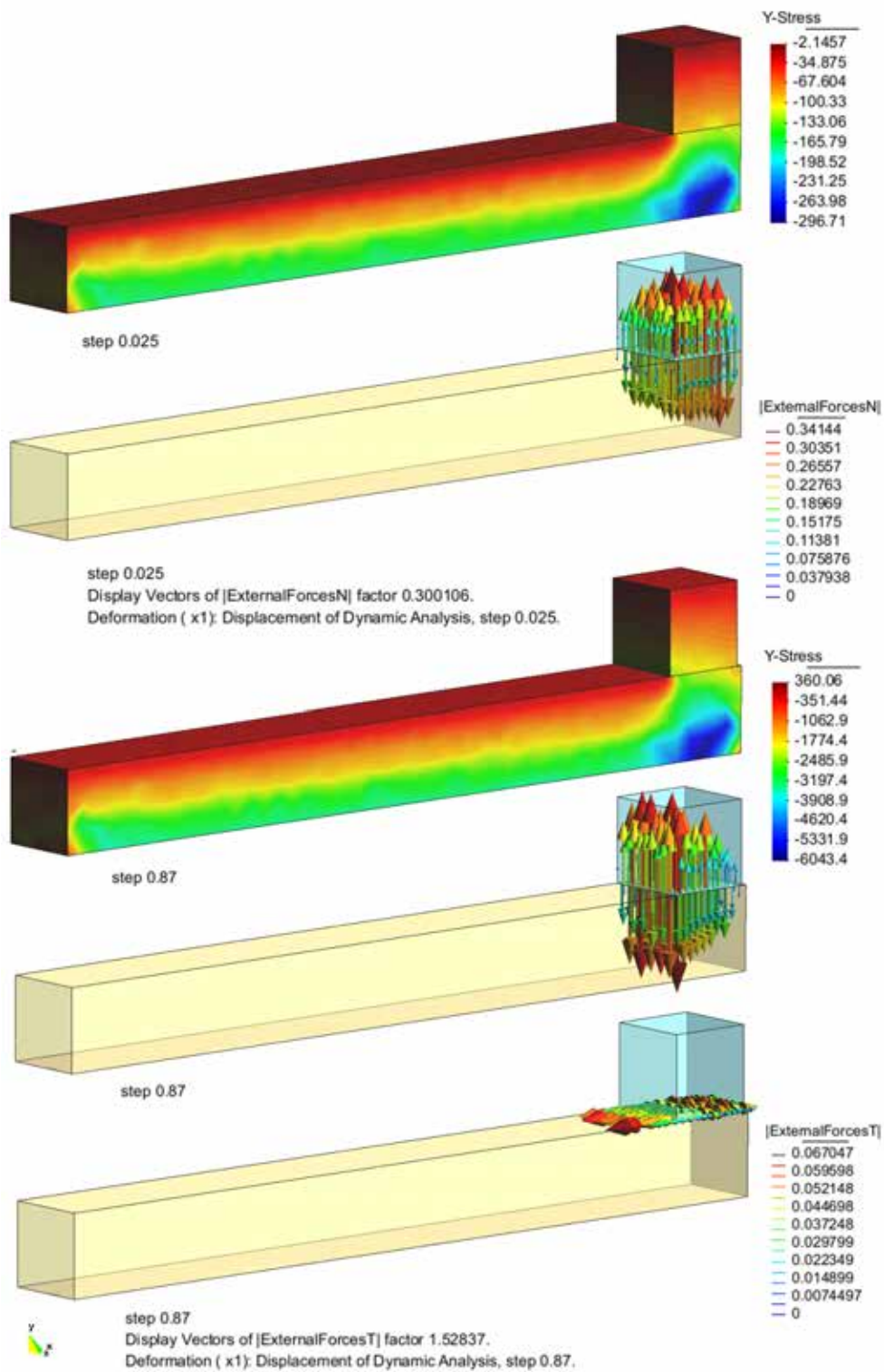


Figure 5.64: Evolution of the normal contact forces when the progressive tangential load is applied.

*The best way to predict the future is to invent it.*

Alan Kay



## Modeling excavation processes

### 6.1 Introduction

This chapter presents an application of the PFEM to model excavation problems. During an excavation process the geometry of the problem is changing continuously. Geomaterials are excavated and the cutting tools are worn. The definition of the excavation domain and the geometry of the tools change at the time these phenomena occur.

To model this problem not only the computation of the contact forces between solid domains is needed, also a scheme to capture and treat the changing geometries is required. The features of the PFEM allows us to model rapidly changing boundaries and adapt the geometry to every mechanical process. These modelling capabilities allow us to account for finite deformations and several contacts at the same time. That opens a new way for the treatment of wear in solid materials and also for modeling excavation problems with dynamic boundary generation.

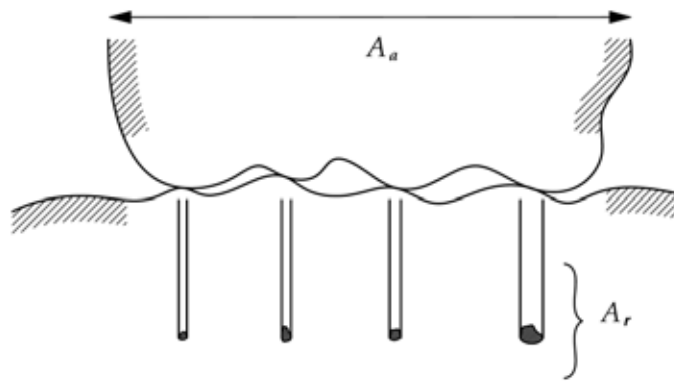
When the process of crushing and digging a solid material is studied several complicated phenomena must be taken in account. Most of them are related to the constitutive behavior of materials and the mechanical contact. To pay a specific attention to each particular physical phenomenon several multi-scale analysis are needed. These are usually focussed on what happens on the material surface where contact is active. To simplify these processes usually the problem is seen from the macro-scale and modeled with particular laws.

The constitutive laws for the bodies coming into contact describe the material behavior, and do not affect the main formulation of the contact problem. This means that for contact mechanics these laws can be arbitrary. However, it is clear that the physical properties of the surfaces of the bodies are influenced by the general constitutive behavior. The usual form to describe wear and friction of materials is by means of the surface properties. They are the result of the constitutive behavior of the material in a narrow zone adjacent to the surface.

In the following sections the general theory for modelling friction and wear is introduced. After that the model for excavation is explained. The developed model tries to be independent of the constitutive behavior of the bodies in order to not restrict its general application.

## 6.2 Friction

Frictional contact was introduced first in Section 5.3.3. Here, some basic properties are presented in order to introduce later the theory of wear and excavation. The general definition is that *friction* is the resistance to motion that exists when a solid object is moved tangentially with respect to the surface of another that it touches, or when an attempt is made to produce such a motion. When a tangential motion occurs, the friction force always acts in a direction opposite to that of the relative velocity of the surfaces. Friction is an outcome of the superficial process that produces wear on surfaces. It will be seen that the frictional force is the consequence not the cause of most phenomena explained in this chapter. However it is important to present an overview of its behavior.



**Figure 6.1:** Schematic view of an interface showing the apparent and real areas of contact

There are some quantitative laws that govern friction. These laws express the magnitude of the friction force as a function of the principal macroscopically observable variables: the applied load, the size of the region of contact and the sliding velocity. These quantitative relations state that:

1. The friction force  $F_T$  is proportional to the normal force  $F_N$ .
2. The friction force is independent of the apparent area of contact  $A_a$ . Thus large and small objects have the same coefficients of friction.
3. The friction force is independent of the roughness of the sliding surfaces. An exception is the sliding of a rough hard body on a much softer one, because the rough surface digs into it and the sliding is on a much larger real contact area  $A_r$ .
4. The friction force is slightly dependent of the sliding velocity  $v$ . It is well known that the friction force required to start sliding is usually greater than the force required to maintain sliding. However the static friction coefficient is a function of the time of contact. In fact only the kinetic friction coefficient is a function of velocity but throughout a certain range of velocities [39], [103], [20].

The previous relations are defined from the observation of the physical properties of contact interaction. A quite convincing explanation for each one is the following:

1. The proportionality between friction force and applied load is a direct consequence of the fact that each is equal to a material constant, characteristic of the surfaces in contact, multiplied by the same area of contact  $A_r$ .
2. The independence of the friction force within the apparent area of contact  $A_a$  is readily explained in terms of the concept that it is the real area, rather than the apparent area, which governs the interaction between the two materials. The real area of contact is the sum of contact junctions and is not dependent on the apparent area (see Figure 6.1). The value of the real area is given by the relation:

$$A_r \geq \frac{F_N}{H} \quad (6.1)$$

where  $F_N$  is the normal contact force and  $H$  the penetration hardness.

3. The weak dependence of friction force on the sliding velocity may be explained as a part of a more general phenomenon, namely the small dependence of the strengths of most solids on the rate of application of the stress. Solids that have a pronounced dependence of strength on strain rate generally show a pronounced dependence of friction force on sliding velocity.
4. The weak dependence of friction on the surface roughness is to be expected. Since little frictional work is done in overcoming surface roughness, most of the work is done in inducing a shear displacement at the junction interface.

The friction mechanism has other influences related to other frictional phenomena. For example, there is the role of lubricants or the relation of temperature and friction. A comprehensive analysis of all friction sources can be found in [104].

A compilation of friction coefficient values is shown in Table 5.1. The models for the numerical simulation of friction were explained in Section 5.3.3. In our work we will use the PFEM for contact problems with contact sliding surfaces. The normal and friction forces, together with the relative displacements of bodies in contact establish the mechanical behavior of the contact interface used for computing wear.

### 6.3 Wear

In order to define wear, a simple process from everyday's life can be used, that is writing. What one sees when using a pencil to write on a piece of paper is just the outcome of a wearing process. It is clear that wear means the removal of material from solid surfaces under mechanical action. In general, wear is related to sliding contact. Wear is generally thought of as destructive but it can be used for the production of surfaces, the preservation of sharp edges or for the measurement of mechanical properties. Usually it involves many different disciplines, e.g. material science, chemistry and applied mechanics.

This is because besides the geometrical properties of surfaces and the frictional forces in the interface, surface chemistry and the material properties close to the surface also play a major role.

Basically, there are four different wear mechanisms:

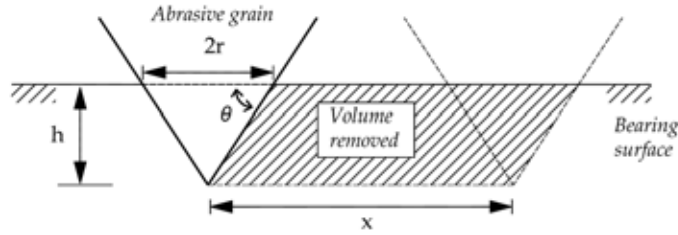
- **Adhesive wear:** It occurs when two smooth bodies slid over each other and fragments are pulled off one surface and adhere to the other. If later these particle come off the surface, they can form loose wear particles.
- **Abrasive wear:** It occurs when a rough hard surface, or a soft surface containing hard particles, slides on a softer surface. The hard surface ploughs grooves in the soft one. The material stemming from the grooves then forms the wear particles.
- **Corrosive wear:** It occurs when sliding takes place in a corrosive environment. The film that protects the surfaces is removed by a sliding action. In that case, the attack due to corrosion starts due to sliding contact.
- **Surface fatigue wear:** It is produced by a repeated sliding or rolling. The material undergoes many loading/unloading cycles which lead to cracks close to the surface. These might result in a breakup of the surface, and lead to the formation of wear particles

A detailed description of wear mechanisms can be found in [104]. From all these forms of wear production, it can be deduced that wear is complicated and can also involve different mechanisms at various stages of the process. Wear in general depends upon the properties of the material surfaces, the surface roughness, the sliding distance, the sliding velocity and the temperature. For processes involved in excavation it is important to reduce the wear mechanisms. Abrasive wear and impact wear are chosen as the most relevant types of wear for the analysis. Impact wear, produced by the collision of bodies, was not defined before, but it is important to be considered in this case.

### 6.3.1 Abrasive wear

Abrasive wear arises when a hard, rough surface slides against a softer surface, digs into it, and plows a series of grooves. The material originally in the grooves is normally removed in the form of loose fragments, or else it forms a pair of mounds along each groove. The material in the mounds is then vulnerable to subsequent complete removal from the surface.

There is also another case of abrasive wear, that arises when hard abrasive particles are introduced between sliding surfaces and abrade material off each. The two forms of wear are generally referred as two-body and three-body abrasive wear process respectively [19].



**Figure 6.2:** A much simplified abrasive wear model showing how a cone removes material from a surface

To quantify abrasive wear through constitutive equations, the properties or effects that play a major role have to be determined. Usually a simple model is considered. In this model, the asperities on the hard surface are conical (see Figure 6.2). A conical asperity, carrying a load  $\Delta F_N$  will penetrate into a softer surface to an extent given by

$$\Delta F_N = H \cdot \Delta A = H \cdot \pi r^2 \quad (6.2)$$

where  $H$  is the indentation of penetration hardness of the softer surface and  $\Delta A$  the penetrating area. The projecting area of the penetrating cone in the vertical plane is  $rh$ . Thus when the cone moves a distance  $dx$ , it will sweep out a volume  $dV$  given by

$$dV = r \cdot h \cdot dx = r^2 \cdot \tan \theta \cdot dx = \frac{\Delta F_N \cdot \tan \theta \cdot dx}{\pi H} \quad (6.3)$$

Then

$$\frac{dV}{dx} = \frac{\Delta F_N \cdot \tan \theta}{\pi H} \quad (6.4)$$

The contribution of all asperities yields

$$V = \frac{F_N x \overline{\tan \theta}}{\pi H} \quad (6.5)$$

where  $\overline{\tan \theta}$  is a weighted average of the  $\tan \theta$  values of all individual cones. Finally it becomes an expression that was proposed for adhesive wear by [7] and [52], and later also applied to abrasive wear [104]. This is called an *Archard* type law:

$$V_{wear} = k_{abr} \frac{F_N g_T}{H} \quad (6.6)$$

where  $g_T := x$  is the relative sliding distance and  $k_{abr}$  is the abrasive wear coefficient that physically represents the average tangent of the roughness angle divided by  $\pi$ . The range of  $k_{abr}$  is between  $10^{-2}$  to  $10^{-5}$ . Typical values can be seen in Table 6.1. In the three-body case, the range for the abrasive wear coefficient is an order of magnitude smaller. It seems that the abrasive grains in the three-body case spend about 90% of the time rolling,



thus not producing abrasive wear particles, and only about 10% of the time sliding and abrading the surfaces.

**Table 6.1:** Abrasive Wear Coefficient Values  $k_{abr}$  [104]

	File	Abrasive Paper (new)	Loose Abrasive Grains	Coarse Polishing
Dry surfaces	$5 \cdot 10^{-2}$	$10^{-2}$	$10^{-3}$	$10^{-4}$
Lubricated	$10^{-1}$	$2 \cdot 10^{-2}$	$2 \cdot 10^{-3}$	$2 \cdot 10^{-4}$

Note: These are maximum rates for sharp fresh abrasive surfaces. After wear and clogging, abrasive wear rates are generally reduced by a factor of about 10

The effect of hardness on the abrasive wear rate is the fact that an abrasive material must be harder than the surface to be abraded, though not enormously harder. This is an important feature, as a no abrasive material will cut anything harder than itself. In that context is helpful to remember that the most common abrasive contaminant is sand ( $\text{SiO}_2$ ) with a hardness of about  $750 \text{ kg/mm}^2$ . The most common engineering material is steel, whose hardness can be varied from 200 to  $1000 \text{ kg/mm}^2$ . Therefore, a very hard steel is not abraded by sand, but softer grades of steel are readily worn away. The penetration hardness coefficient  $H$  is specified by the Brinell test or the Vickers test, the last one is made pressing a square pyramid into a flat surface and the diagonals of the square indentation are measured. The hardness coefficient forms the basis of the Mohs's hardness scale, widely used by mineralogists. A comparison of the Mohs and Vickers hardness scales is presented in Table (6.2).

**Table 6.2:** Mohs's Hardness Values [104]

Mineral	Mohs's Hardness Number	Indentation Hardness, $\text{kg/mm}^2$
Talc	1	23
Gypsum	2	30
Calcite	3	100
Fluorite	4	160
Apatite	5	400
Orthoclase	6	600
Quartz	7	810
Topaz	8	1200
Corundum	9	1860
Diamond	10	7800

A simple expression defines an approximate correspondence between these values:

$$\text{Mohs's hardness} = 4.28 \log H - 5.38 \quad (6.7)$$

### 6.3.2 Impact wear

Impact wear is the wear produced by the collision of two surfaces having large velocities normal to their interface. Cutting tools regularly operate under impact conditions. The literature on impact wear gives several approximations for the cause. The first one says that impact wear of ductile materials is similar to adhesive wear of the same materials during sliding. The second idea is that for many materials impact wear is surface fatigue wear. Another authors say that impact wear is a manifestation of surface fracture wear.

The conciliation of these various views are covered with the following statements:

- When the impact energies are very large, rapid material removal occurs by a manifestation of surface wear.
- When the impact energies are low, wear occurs by surface fatigue wear.

Mild impact wear can be treated as a form of adhesive wear. Then it is not strange to use as a quantitative measure of that wear an *Archard*-type law, like in abrasive and adhesive wear. That assumption yields to:

$$V_{wear} = k_{imp} \frac{F_N gT}{H} \quad (6.8)$$

where  $k_{imp}$  is a non-dimensional wear coefficient and  $H$  the penetration hardness. The load  $F_N$  can be replaced by the friction force divided by the friction coefficient  $\mu$ . Then the friction force multiplied by the sliding distance represents the energy dissipated in sliding. Further a coefficient  $\alpha_s$  is defined as the proportion of total impact energy expended in interfacial slip. The resultant term of the entire transformation  $\left(\alpha_s \frac{k}{\mu}\right)$  is replaced by a parameter  $K_{imp}$ . Finally, if  $I$  is the energy per impact and  $n$  is the total number of impacts, (6.8) we obtain

$$V_{wear} = K_{imp} \frac{I n}{H} \quad (6.9)$$

where  $K_{imp}$  is the non-dimensional impact wear coefficient. The values for  $K_{imp}$  are computed from impact tests described in the literature, see [104]. The range of  $K_{imp}$  is between  $5 \cdot 10^{-4}$  -  $10^{-6}$ . Typical values can be seen in Table 6.3.

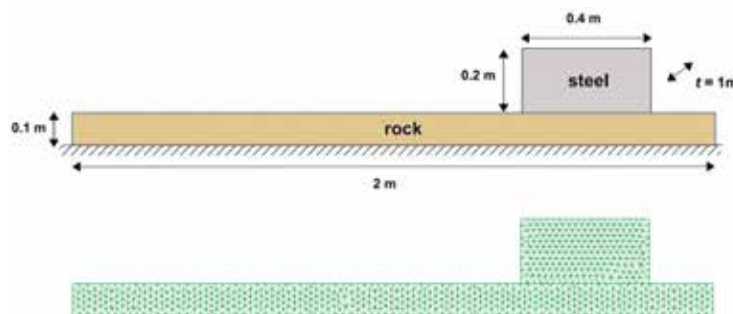
**Table 6.3:** Impact Wear Coefficient Values  $K_{imp}$  [104]

Surface Measured	Energy of Impact (kg-mm)	Number of Impacts	Remarks	Impact Wear Coefficient
Steel	2	1	Transfer	67
Steel	20	1	Lubricated	10
Tungsten	138	$2.4 \cdot 10^3$	Average	130
Hard Steel	160	$5 \cdot 10^5$	Average	28
Hard Steel	0.36	$6.5 \cdot 10^4$	-	14
Hard Steel	0.36	$4.3 \cdot 10^5$	Lubricated	1.2
Soft Steel	0.36	$10^4$	-	247
Soft Steel	110	500	Lubricated	24

### 6.3.3 Wear calibration test

This is a simple example to calibrate the abrasive wear computation. The problem consists in a steel block which is dragged along a geo-material base. For the geo-material different rock properties has been considered. The abrasion on the rock and on the steel can be computed theoretically and compared with the numerical results.

Figure 6.3 shows the considered model and mesh. After an stabilization of the contact forces due the self weight an imposed displacement is applied to the top of the steel block, which is moved on the rock along 6,4 cm distance.



**Figure 6.3:** Problem geometry and mesh discretization for the two blocks of steel and rock. The initial mesh has 1426 triangular elements and 840 nodes.

Table 6.4 shows the material properties of the steel and the used rocks. Three tests are performed considering different rock types for the foundation.

It is important to ensure a good contact force distribution on the base. This use to vary if the problem is considered dynamic due to the wave propagation inside of the solid domains. In fact it is not a dynamic problem but in order to check the accuracy of the method both the static and the dynamic case are considered. Tables 6.6 and 6.5 show the obtained results for wear on steel and rocks respectively.

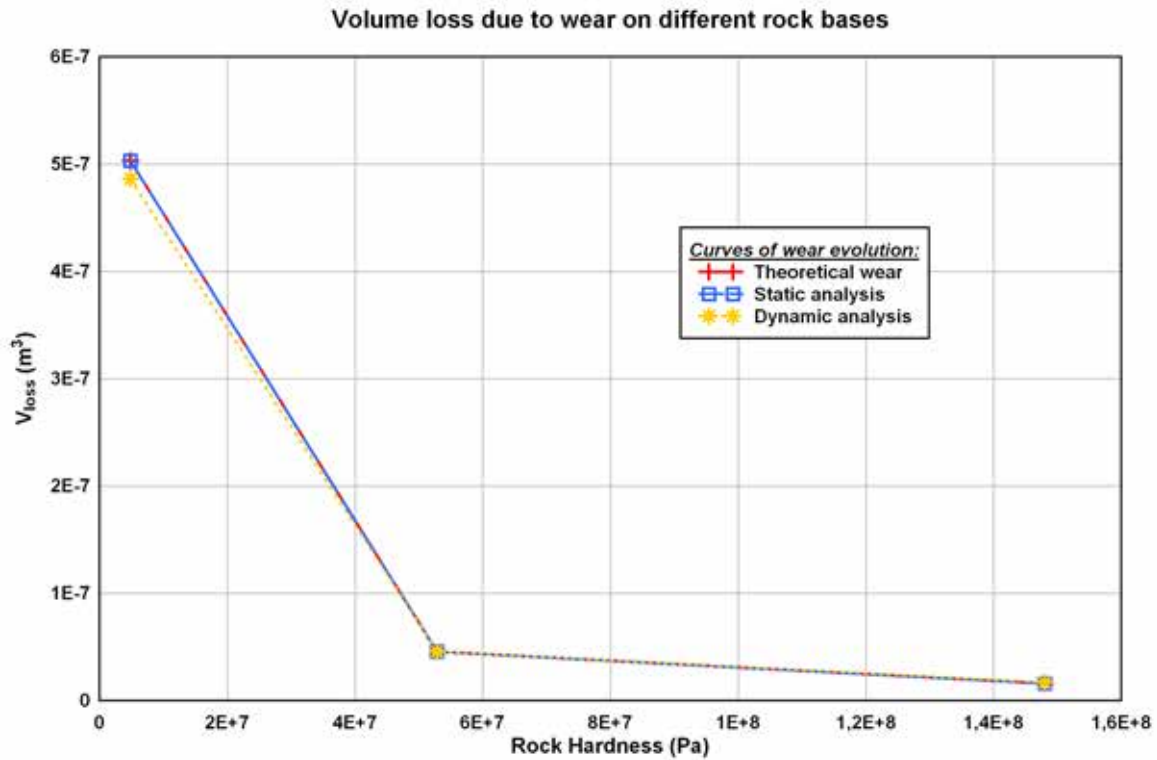
The results of the tables are depicted in Figure 6.4 in order to show the evolution of the volume worn compared with the rock hardness.

**Table 6.4:** Material properties for the wear test.

Materials	Steel	Sandstone Sherwood	Sandstone Fell	Granite Dealbeattie
E (Pa)	$2,0 \cdot 10^{11}$	$6,40 \cdot 10^9$	$3,27 \cdot 10^{10}$	$4,11 \cdot 10^{10}$
$\nu$	0,2	0,3	0,3	0,3
$\delta$ ( $kg/m^3$ )	7850	2680	2690	2670
$H$ (Pa)	$9 \cdot 10^9$	$4,8 \cdot 10^6$	$5,28 \cdot 10^7$	$1,48 \cdot 10^8$
$K_w$	0,06	0,06	0,06	0,06

**Table 6.5:** Volume loss due to wear on steel.

Test	Hardness (Pa)	Theoretical $V_{loss}$ ( $m^3$ )	$V_{loss}$ Static Analysis	$V_{loss}$ Dynamic Analysis
Test 1	$9 \cdot 10^9$	$2,68 \cdot 10^{-9}$	$2,72 \cdot 10^{-9}$	$2,71 \cdot 10^{-9}$
Test 2	$9 \cdot 10^9$	$2,68 \cdot 10^{-9}$	$2,72 \cdot 10^{-9}$	$2,70 \cdot 10^{-9}$
Test 3	$9 \cdot 10^9$	$2,68 \cdot 10^{-9}$	$2,72 \cdot 10^{-9}$	$2,70 \cdot 10^{-9}$



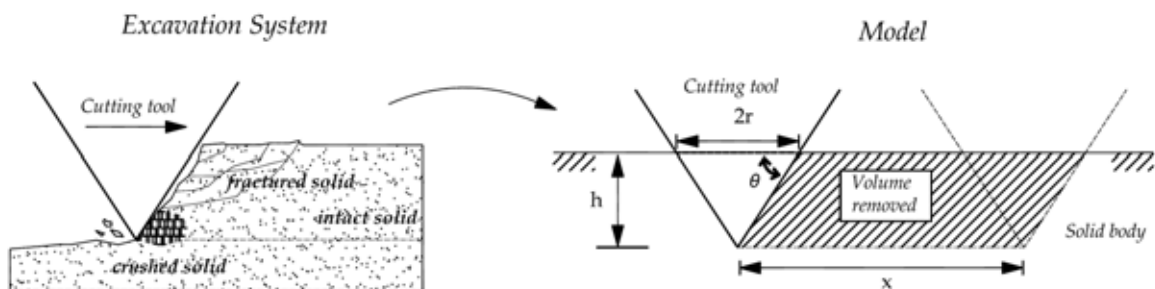
**Figure 6.4:** Volume worn compared with the hardness of the rock surfaces under analysis.

**Table 6.6:** Volume loss due to wear on rocks.

Test	Materials	Hardness (Pa)	Theoretical $V_{loss}$ ( $m^3$ )	$V_{loss}$ Static Analysis	$V_{loss}$ Dynamic Analysis
Test 1	Sandstone Sherwood	$4,8 \cdot 10^6$	$5,03 \cdot 10^{-7}$	$5,03 \cdot 10^{-7}$	$4,86 \cdot 10^{-7}$
Test 2	Sandstone Fell	$5,28 \cdot 10^7$	$4,58 \cdot 10^{-8}$	$4,56 \cdot 10^{-8}$	$4,59 \cdot 10^{-8}$
Test 3	Granite Dealbeattie	$1,48 \cdot 10^8$	$1,63 \cdot 10^{-8}$	$1,63 \cdot 10^{-8}$	$1,70 \cdot 10^{-8}$

## 6.4 Excavation

The mechanism of cutting and digging on the ground is not so much different compared to wear in a solid surface. The most significant difference is the scale where the process takes place. The wear of materials usually is associated with the micro-scale of the solid surfaces. When one refers to the erosion of a fluid on a solid surface the scale starts to change. The same happens when a solid surface digs onto another. Excavation processes can be described with the same physical variables and with similar laws as wear. Figure 6.5 shows how a macro-scale model for excavation is analog to the micro-scale model used for wear (see Section 6.3).



**Figure 6.5:** A simplified excavation model for a removal of a solid material by means of a cutting tool.

The most common mechanical processes in an excavation are the material fracture and crush. By means of these two mechanisms the cutting tool removes the material of the surface. The distribution of forces in the contact surface and the properties of the material govern these processes. With the coupling of the constitutive behavior from crushing and fracturing a rate of removed volume can be defined. The volume loss rate depends on the relative displacement of the cutting tool and the hardness of the material. A rate function is used for the description of the excavation as was done with wear. Similar analogies have been employed by other researchers in order to model fracture of hard cutting indenters in a brittle material, see [25].

There is an inexorable declaration of the theory of dimensions which says that there is no other way in which the key variables of load, distance of sliding and hardness can be arranged [114]. Hence the excavation model is going to be similar to an *Archard*-type equation. Taking as a reference equation (6.5), the evaluation of excavated volume is given by:

$$V_d = K_d \frac{F_N g_T}{H_d} \quad (6.10)$$

where  $V_d$  is the volume loss of material along the contact surface,  $g_T$  is the sliding distance of the cutting tool,  $F_N$  is the normal force to the contact surface and  $H_d$  is the equivalent hardness of the material for an excavation process. The constant  $K_d$  is the

non-dimensional digging coefficient, which depends on the relative contribution of body grooving abrasion and fracture process operating.

Equation 6.10 is a very general unspecific model. For its features it would be useful for indenters and drag bits, but not for other excavation tools like discs. The excavation and wear mechanisms involved in a performance of disc cutters must be studied to formulate a more specific and accurate model. Therefore the most common tool sets used in excavation machines are analyzed. Most of them are composed for picks and discs. A particular model is formulated for each one of them.

#### 6.4.1 Boring machines

The proposed excavation model of equation (6.10) must be adapted to the tunnel boring machine under analysis. The model is specially determined by the sum of cutting tools employed in each case. The critical conditions for wear and excavation will appear in hard materials, where the interaction between the solid ground and the boring machine is harder. For the excavation of these materials (e.g. massive rock) two types of machines are used:

- Tunnel Boring Machines (TBMs), for a more general excavations inside of a certain range of rock hardness, and
- Roadheaders, which permit more punctual attacks usually in softer rocks. A example of that machines is shown in Figure 6.6.

The characteristic cutting tools for TBM's are discs and for Roadheaders are drag picks.

#### 6.4.2 TBMs

Some basic concepts related with the design of TBMs are presented here in order to formulate an excavation model for cutting discs. Further information can be consulted in [38]. The objective is to define a boring law, similar to equation (6.10), to predict the amount of material that is excavated during the interaction of a TBM with a solid ground. Hence, focussing in the TBM analysis, some characterizing parameters must be introduced.

The TBM thrust, or normal force, usually is distributed along the discs of the machine head. For each cutting disc two force variables are defined: the normal force  $F_n$  and the disc rolling force  $F_r$ , see Figure 6.7.

The cutting rate is defined as  $\left(\frac{s}{p}\right)$ , where  $s$  is the disc spacing and  $p$  the penetration of the disc due to a machine head revolution. Usually  $p$  is expressed by the index  $PRev := p$ , which means the penetration per revolution.  $PRev$  is controlled primarily by  $F_n$ . The ratio of  $F_n$  and  $F_r$  is defined as the cutting coefficient  $C_c$ , this is:

$$C_c = \frac{F_r}{F_n} \quad (6.11)$$

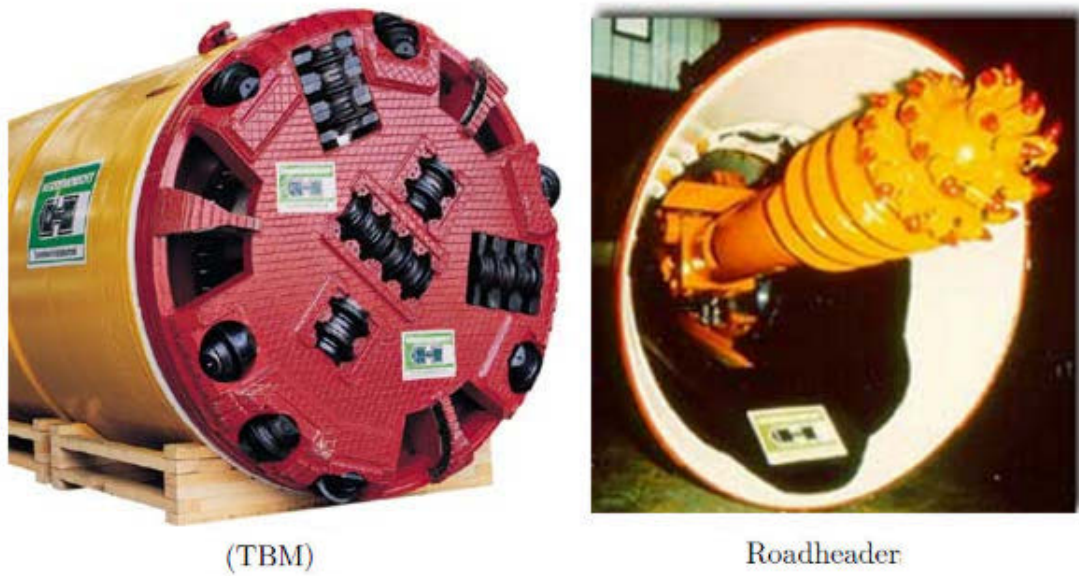


Figure 6.6: Common types of boring machines (Herrenknecht).

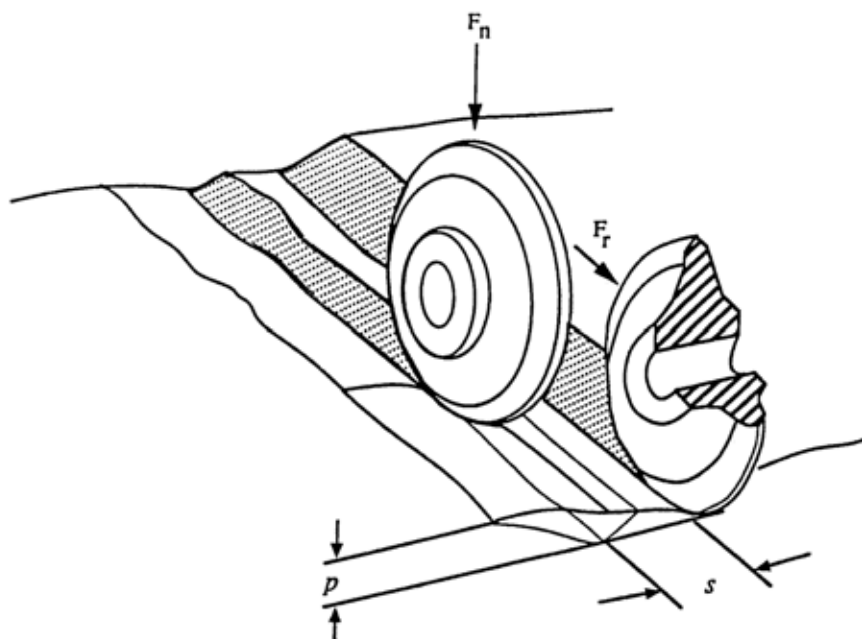
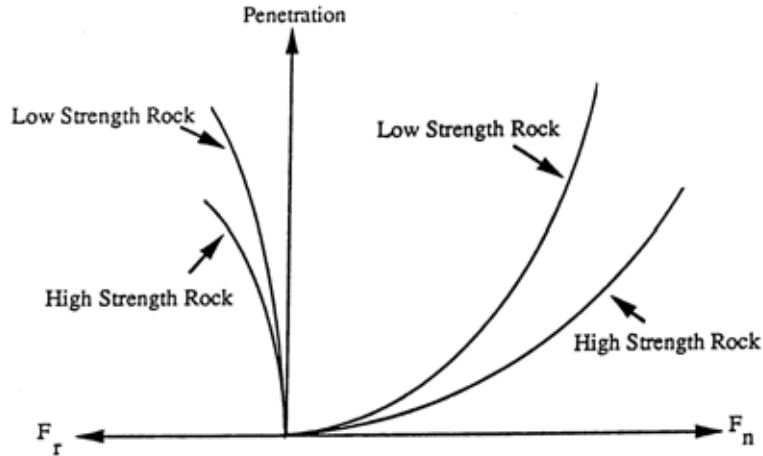


Figure 6.7: Disc force and geometry for kerf cutting



The penetration of the discs in the solid material  $p$ , is a non-linear function of the disc normal forces and rolling forces. The interaction, in the case of rocks, is indicated in Figure 6.8.



**Figure 6.8:** Disc cutter force variation with penetration for high and low strength rocks

The changing slope corresponds to a transition in the dominance between crushing and chip formation and is called the *critical thrust*. Unless a force of this magnitude can be applied, chipping between grooves will not occur. The critical thrust is directly related to rock strength or hardness and increases with cutter spacing  $s$  and disc edge width  $w$ .

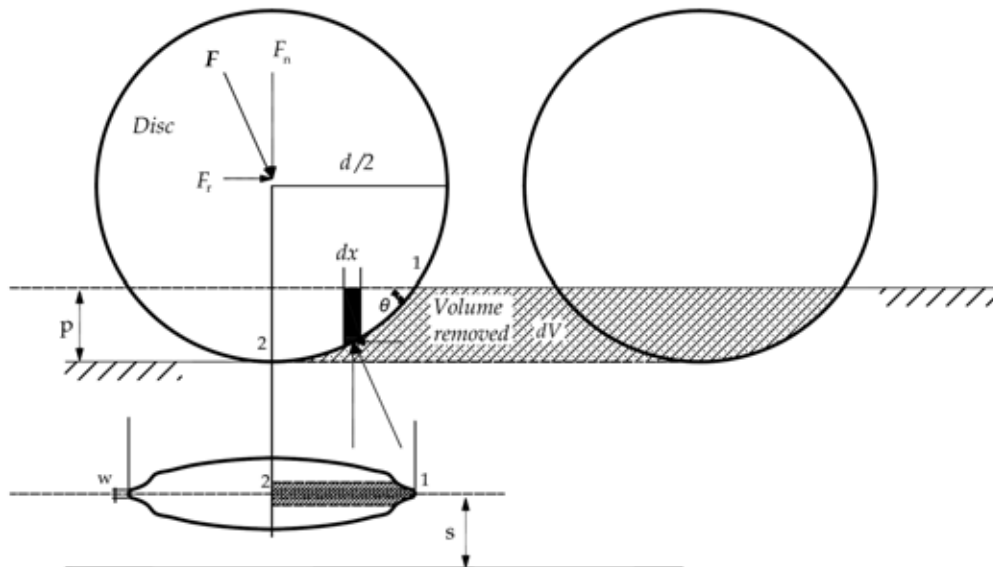
Using that information, the prediction of the excavated volume can follow the simple model of Figure 6.9. The cutting tool has a known section; see Figure 6.10. The part of the tool that will penetrate on the solid domain can be considered that has a sectional area of  $p \cdot w$ , where  $w$  is the disc edge width of the tool. Thus when the disc moves through a distance  $dx$  it will sweep out a volume  $dV$  given by

$$dV = p \cdot w \cdot dx \quad (6.12)$$

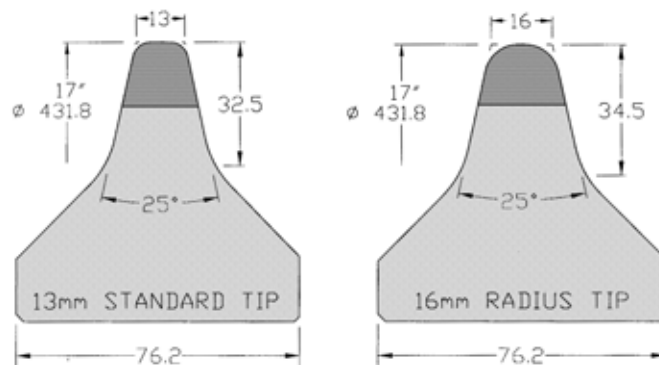
The positioning of discs on a cutting head of a TBM is the factor that performs the excavation. The chips formed between discs must be also considered as removed volume. This gives

$$dV = p \cdot w \cdot dx + p \cdot s \cdot dx = p \cdot (s + w)dx \simeq PRev \cdot (s + w)dx \quad (6.13)$$

where  $s$  is the cutter spacing and  $w$  is the disc edge width. Equation (6.13) accounts the kerf interaction and chip formation and basically depends on the disc geometry and the penetration index. There are some empirical equations developed from data on rock testing that can be used to determine  $PRev = PRev(F_n)$  [38]. The most important are presented next.



**Figure 6.9:** A simplified excavation model for removal of solid material by means of a cutting disc.



**Figure 6.10:** Characteristic section for the cutting ring of a disc (Herrenknecht profiles).

- For sedimentary rocks, *Farmer and Glossop* [42]:

$$PRev = 624 \frac{F_n}{\sigma_{tB}} \quad (6.14)$$

where  $\sigma_{tB}$  is the *Brazilian* tensile strength.

- For hard rocks (UCS 140 to 200 MPa), *Graham* [48]:

$$PRev = 3940 \frac{F_n}{UCS} \quad (6.15)$$

where UCS is the uniaxial compressive strength, the most widely test for rock strength.

- For mining in coal, *Hughes* [54] suggested:

$$PRev = 1.667 \left( \frac{F_n}{UCS} \right)^{\frac{1}{2}} \cdot \left( \frac{2}{D} \right)^{0.6} \quad (6.16)$$

where  $D$  is the disc diameter in millimeter assuming that only one disc tracks in each kerf groove, the normal practice for TBM design.

$PRev$  evaluated in units of millimeters/revolution,  $F_n$  in  $N$ , and the uniaxial compressive strength (UCS) and *Brazilian* tensile ( $\sigma_{tB}$ ) strengths expressed in units of  $kPa$  or  $Pa$ , as noted. The predictive ability of the empirical correlations for  $PRev$  are shown in the table of Figure 6.7.

**Table 6.7:** Comparison of TBM Case Study and Predicted Penetration Rates [38].

Project Information		Rock Strength (MPa)*		TBM Performance		Prediction Method (1)Farmer/ Glossop, (2)Graham, (3)Hughes		
		Brazil		$F_n$ (kN)	$\frac{P}{rev}$ (mm)	(1) $\frac{P}{rev}$	(2) $\frac{P}{rev}$	(3) $\frac{P}{rev}$
Location	Rock Unit	UCS	Tensile					
Buffalo	Falkirk Dolostone	188	13.3	134	7.6	6.3	2.8	2.9
	Oatka Dolostone	139	13.0	108	10.4	5.2	3.1	3.3
Rochester	W./Sodus Shale	80	(8.0)	99	10.0	-	4.9	5.7
	Reynales Limestone	128	15.0	141	6.8	5.9	4.3	5.0
	Maplewood Shale	68	(6.8)	98	10.4	-	5.7	6.8
	Grimsby Sandstone:							
	Wet	130	10.1	112	7.9	6.9	3.4	3.7
	Dry	208	-	-	6.1	11.5	4.1	4.6
Chicago	Romeo Dolostone	237	17.0	145	8.0	5.3	2.4	2.4
	Markgraf Dolostone	168	12.1	137	9.3	7.1	3.2	3.5
Austin	Austin Chalk	10	1.3	33	9.6	15.7	99.1	18.5

\* (8.0) and (6.8) for Brazil tensile strength are estimated as UCS/10.

For instance an equation similar to (6.15) can be selected to define the penetration:

$$PRev = k_p \frac{F_n}{UCS} (mm) = \frac{k_p}{1000} \frac{F_n}{UCS} (m) \quad (6.17)$$

where  $k_p$  is a constant that depends on the material ( $k_p = 3940$  for hard rocks). Using (6.17) in (6.13) yields to

$$\frac{dV}{dx} = PRev(F_n) \cdot (s + a) = \frac{k_p}{10^3} \frac{F_n}{UCS} (s + w) \quad (6.18)$$

Equation (6.13) is the contribution for one disc. From the contribution of all discs we obtain

$$V_d = \sum_{i=1}^{n_d} \left( \frac{k_p}{10^3} \frac{F_{ni}}{UCS} (s + w) \cdot x_i \right) \simeq \frac{k_p}{10^3 UCS} \left( \sum_{i=1}^{n_d} F_{ni} \right) (\bar{s} + \bar{w}) x \quad (6.19)$$

where  $n_d$  is the number of discs,  $F_N$  is the total normal force and  $g_T$  is the relative sliding distance. All variables are expressed in SI units. Finally (6.19) becomes an expression that is written as

$$V_d = K_d \frac{F_N}{UCS} g_T \quad (6.20)$$

where  $K_d$  is the boreability coefficient that physically represents the weighted average of the values of spacing  $\bar{s}$  and the disc width  $\bar{w}$ .

$$K_d = \frac{k_p (\bar{w} + \bar{s})}{10^3} \quad (6.21)$$

Depending on the selected equation for computing  $PRev$ , different expressions can be formulated for  $K_d$ . This means that  $K_d$  is mainly a parameter that has to be calibrated using the properties of the excavated material and the geometrical properties of the TBM cutter head: discs distribution and disc geometry.

Equation (6.19) depends on the normal force applied to the excavation front  $F_N$ . This is only a component of the contact forces that the head of the TBM receives. There are also the rolling forces on the discs of the TBM that have tangential direction. From equation (6.11) the relation between rolling forces and normal forces is defined by the cutting coefficient  $C_c$ . The inverse of  $C_c$  can be seen as an equivalent friction coefficient for the macroscopic scale  $\mu_{exc} := \frac{1}{C_c}$ . Therefore the tangential forces on the cutting head of the TBM are defined as

$$F_T = \mu F_N \quad \text{with} \quad \mu = \frac{\sum_{i=1}^{n_d} \frac{1}{C_c}}{n_d} \quad (6.22)$$

Usually  $F_T$  is defined as the cutting force and  $F_N$  as the thrust force of the tunneling machine.

The cutting coefficient  $C_c$  can be predicted as a function of  $PRev$  and the disc diameter only, with the influence of rock strength implicit in the computed value of  $PRev$ . Some empirical equations for  $C_c$  are presented next:

- Assuming  $PRev$  equal to the depth of indentation or cut, *Roxborough and Phillips* [112]:

$$C_c = \frac{F_n}{F_r} = \sqrt{\frac{PRev}{(D - PRev)}} \quad (6.23)$$

where  $D$  is the disc diameter.

- The equation from Colorado School of Mines [100]:

$$C_c = \tan \frac{\phi}{2}; \quad \phi = \cos^{-1} \left[ \frac{(R - PRev)}{R} \right] \quad (6.24)$$

where  $R$  is the disc radius.

- Expression proposal by *Hughes*[54]:

$$C_c = 0.65 \sqrt{\frac{PRev}{0.5 D}} \quad (6.25)$$

The units for  $PRev$ ,  $D$  and  $R$  are mm. Table 6.8 shows the results for  $C_c$  in comparison for 432 mm diameter cutters. Other examples for the determination of  $C_c$  can be found in [30].

**Table 6.8:** Results of  $C_c$  prediction equations for a 432-mm-diam cutters [38].

PRev(mm)	Roxborough and Phillips/CSM	Hughes
4	0.10	0.09
8	0.14	0.13
12	0.17	0.15

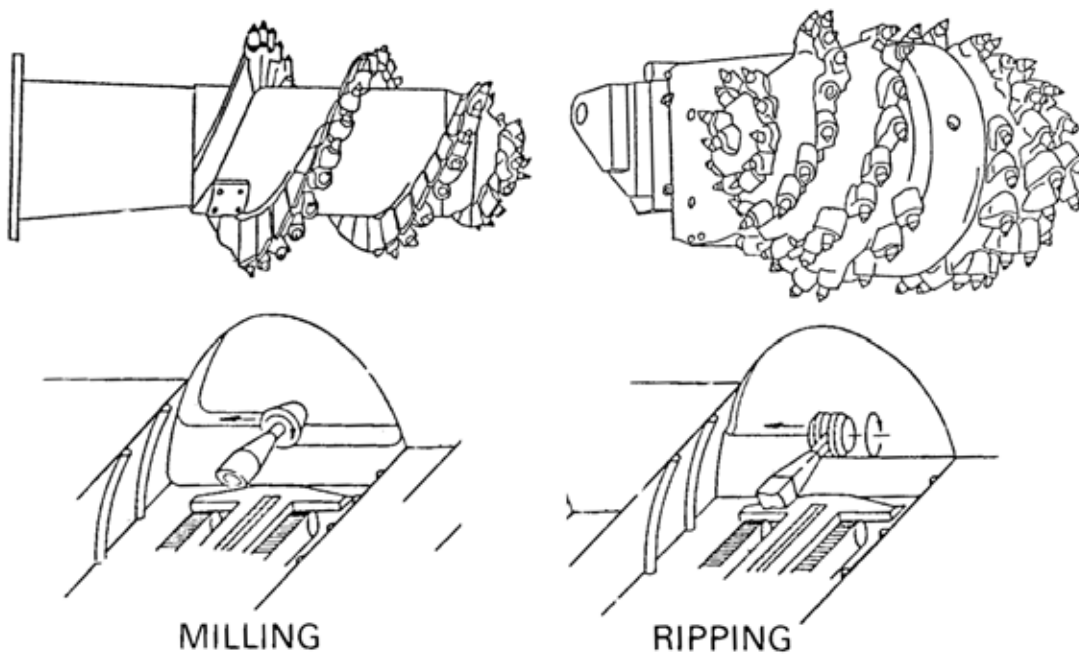
Equation (6.20) is only a simple excavation law for the performance of TBM machines. The accuracy must be tested in future with a calibration and a disciplined research on the field.

There are other geomechanical characteristics with big influence in the solid (rock) properties for excavation: the rock quality design (classified by the index RQD), the spacing of the most influential joints of the rock masses (classified by the Join Spacing index), the groundwater situation, the orientation of the joint sets respect to the engineering work. These parameters are used to classify the Rock Mass Rating (RMR) which gives a description of the rock mass. This information could be used and included in a more

precise excavation model. Also, in order to develop a better model in the future, some characteristics of the NTH prediction methodology and the CSM methodology would be also a good reference to consider [80], [99], [110].

### 6.4.3 Roadheaders

Roadheaders are another type of boring machines used for ripping and milling rocks; see Figure 6.11. Usually they are composed by drag picks installed on a rotation head at the end of a strong mechanical arm. The excavation system is performed by the pick cutting sequence. The first pick that points out will let a free face for the successive picks. There are two main types of picks: the conic bit and the chisel; see Figure 6.12.



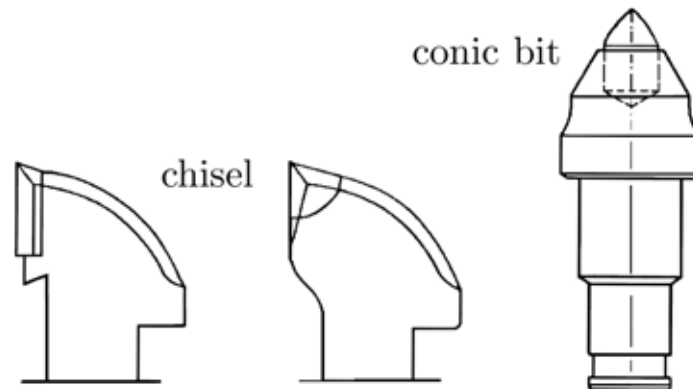
**Figure 6.11:** Ripping and milling cutting systems.

Usually chisels are more efficient than other types of picks in soft rocks where the conic bit is the least efficient because it concentrates smaller forces during the cut. For abrasive rocks and hard rocks conic bits will be more resistant and robust. An extended analysis on the excavation with drag picks can be found in [30] and [113].

The cutting scheme for each individual pick corresponds to the basic model presented in Figure 6.5. Hence the estimation of the volume loss of material is given by equation (6.10), which for the particular case of drag picks is:

$$V_{dp} = K_{dp} \frac{F_N gT}{H_d} \quad (6.26)$$

where  $K_{dp}$  is a measure of the boreability or excavability of the pick which depends



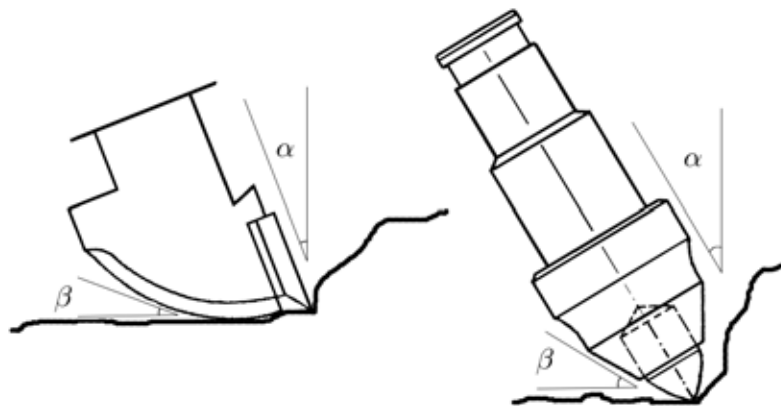
**Figure 6.12:** Types of picks

on extra variables like the clearance angle in the cutting direction  $\beta$  and the rake angle of the bit  $\alpha$ ; see Figure 6.13.  $H_d$  is the hardness of the material.

The derivation of (6.26) is analogue to the one performed for abrasive wear, see Section 6.3.1. From equation (6.5) a physical definition of  $K_{dp}$  can be obtained. It is given by

$$K_{dp} = \frac{\tan \theta}{\pi} \quad (6.27)$$

where  $\theta = \bar{\beta} + \bar{\alpha}$  is the sum of the average clearance angle and the average rake angle for the Roadheader bits.



**Figure 6.13:** Drag tools parallel to the cutting direction.  $\alpha$  : rake angle;  $\beta$  : clearance angle.

## 6.5 Modeling excavation with the PFEM

The goal of using a particle based method is to have the capacity of modelling rapidly changing geometries. An example of that is an excavating process. When a hard solid domain digs on a softer one the geometry of the surface continuously changes. Some part of the domain is removed from the excavated solid. This is how the phenomena occurs and how it is modelled with the PFEM.

Equation (6.10) is translated to the discrete dynamics of particles. In the PFEM every particle represents a part of the domain volume. The surface of a body is composed for particles that represent the volume of the same body associated to them. By means of the contact model (Section 5) the interaction between two solid domains can be quantified. The result is the normal force  $F_N$  associated to the surface particles, and the relative velocities or the sliding distances. Therefore, for each particle laying on the surface, the volume loss of material can be estimated in time by the excavation model. Applied to the most general form, equation (6.10) yields:

$$V_d^{t+\Delta t} = V_d^t + K_d \frac{\|\mathbf{F}_N\|}{H_d} (\|\mathbf{v}_t\| \cdot \Delta t) \quad (6.28)$$

where  $\mathbf{v}_t$  is the relative tangent velocity between the contact surfaces and  $\Delta t$  is the time step. The volume loss of material  $V_d$  can be compared with the volume associated to each contact particle. From this comparison different strategies for geometry shaping can be formulated. However some initial assumptions must be considered before applying any geometrical change.

When the amount of excavated material is not small, the force used within the analysis can not be kept constant. This is due to the change of mass, or to the change on the geometry which takes place at the same time contact interaction occurs. To take into account geometrical changes, a coupling with the displacement field is necessary. To avoid the coupling, the variation of the normal pressure due to the material loss and the geometrical changes has to be considered.

In the PFEM the readaptation of the geometry is performed at each time step. That means that the force is not constant during all the analysis, but it remains the same within each time step. If the time step is too large and produces large changes on contact surfaces, the consistence of the physical model is endangered. However, that is not certainly true. In fact, when a body is digging on another, the external force applied in one time step is kept constant. That force is transmitted by the contact surface, which will withstand the same pressure either if it is degraded or not. Therefore, if the global contact pressure on a given surface is the characterizing parameter, the rate of the material volume loss is correctly computed. Consequently, the method is consistent with the physics. This does not penalize the percentage of volume loss, which can be applied lately in terms of the force distribution.

With these assumptions there are two different ways to apply the volume loss prediction  $V_d$  to the geometry shaping. They are presented next.

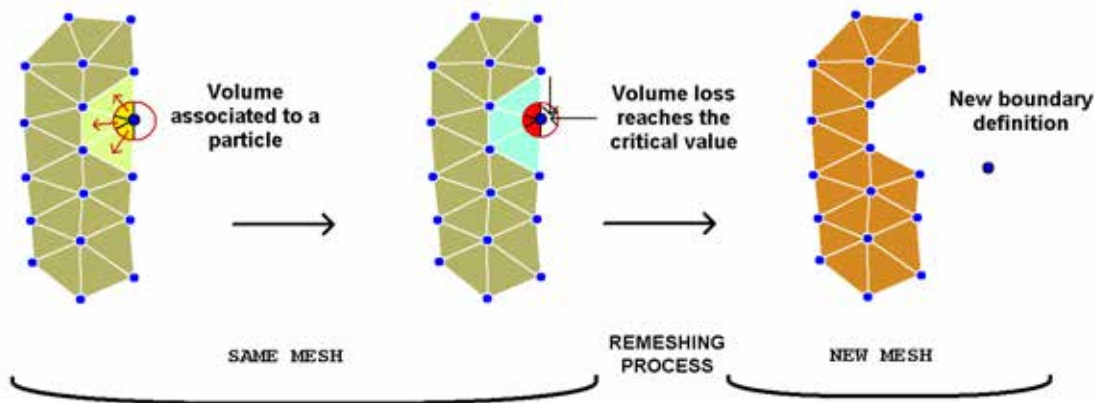


### 6.5.1 Particle release

The first scheme for excavation with the PFEM is the *particle release*. Firstly, each particle of the domain has an associated volume  $V_i$ . During the contact interaction the excavation model (6.28) is evaluated and the loss of volume for each particle is computed  $V_d^t$ . That volume is subtracted from the volume of the particle  $V_i$ . The particle will be released from the body domain when its associated volume is null.

The readaptation of the geometry is done by means of the domain remeshing, the released particles are not included anymore in the body description. This yields an excavated profile on the contact surfaces. When the  $\alpha$ -shape method is used in the remeshing step, a new geometry for the bodies is defined (Section 4.3.1). The geometrical changes are applied intrinsically with the method and they do not affect the future contact interactions.

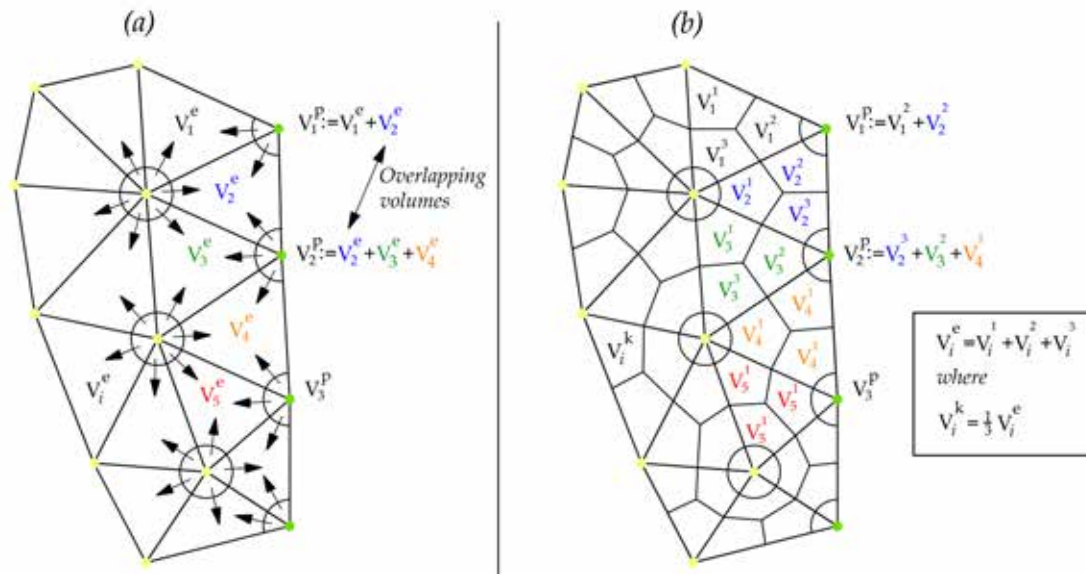
A simple visual scheme of this process is shown in Figure 6.14.



**Figure 6.14:** *Particle release* excavation strategy. When the associated volume of a boundary particle reaches the critical null value the particle is set free.

This approach has some controversial factors. The first one is the assignation of the volume to each particle. There are several possibilities for this. The two most common ones are depicted in Figure 6.15. Volume assignation does not affect to the finite element computations, as the mass of the domain is computed using the elements of the mesh and the densities stored on the particles (as a material property). The volume on particles is only used to model wear and excavation. For that reason the volume assignation will be selected in terms of the excavation strategy.

Figure 6.14 shows that the patch of elements associated to the particle are also shared with other particles. Thus when a particle is released, it takes with it a volume larger than the one which geometrically corresponds to it. In order to be consistent to this fact, there is an overlapping in the volume assignation to particles (Figure 6.15). When a particle is released the volume of the body domain is recomputed and reassigned to



**Figure 6.15:** Assignment of the volume from elements to particles: (a) With overlapping (b) Without overlapping.

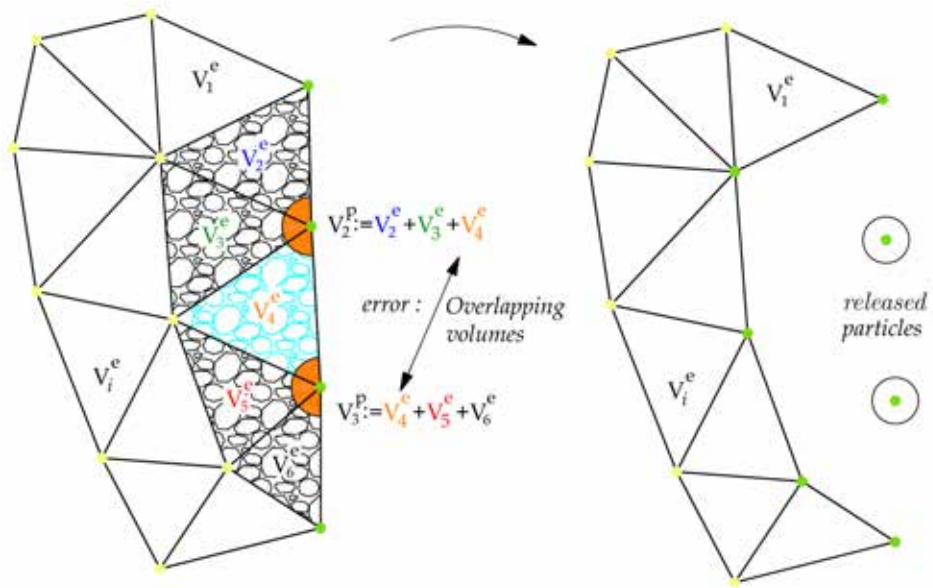
the rest of particles. However, this modification in the volume assignment does not solve the problem entirely because there is an overlapping on neighbor boundary particles when they are taken away. When two neighbor boundary particles are released at the same time, there is a part of the volume that is computed twice. Hence the released volume does not correspond exactly to the computed volume loss. Figure 6.16 shows this situation.

Nevertheless, this is not a big deal and it just has to be taken in account during the calibration of the model parameters. To improve the accuracy of the method another strategy has been developed. It is based in the comparison between the volume of the particles and the volume loss, but at this time the boundary is shaped gradually.

### 6.5.2 Surface shaping

The second excavation scheme developed for the PFEM is the *surface shaping*. It is a more precise alternative than the *particle release* when the geometrical changes due to volume loss have to be applied. It consists in a reduction of the domain volume by means of moving the surface particles inwards the domain. This movement changes the volume of the body coinciding with the computed volume loss. This operation also shapes the boundaries and improves the geometrical adaptation to the excavation. When a gradual loss of volume is performed the ill-adjustment produced by volume overlapping is minimized.

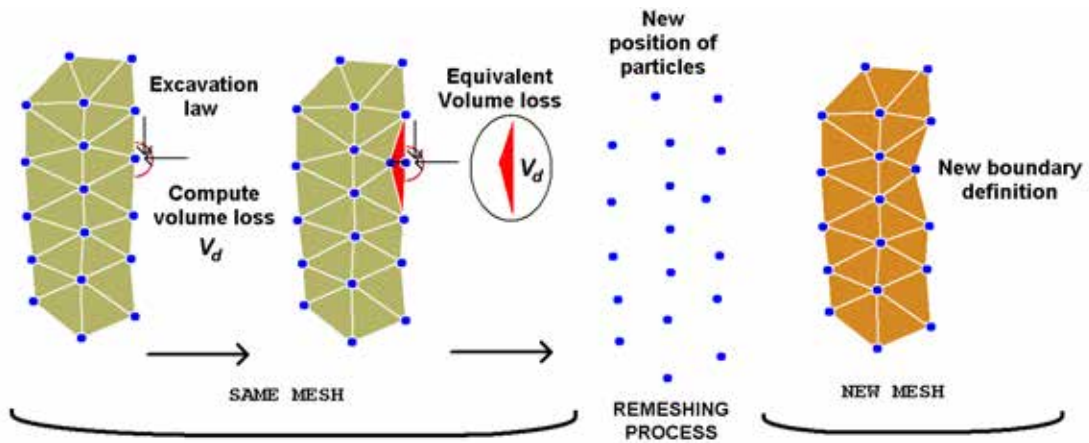
The scheme to shape the surface is not based in the comparison between volume loss and volume assigned to particles. It is based in the volume reduction from the current mesh. When a particle loses volume is moved inwards to the solid domain using the direction of the surface normal and the direction of the main stresses. The proper distance to apply in the shaping is computed by comparing the volume loss  $V_d$  and the volume that



**Figure 6.16:** Error induced in the volume reduction due to the release of consecutive particles.

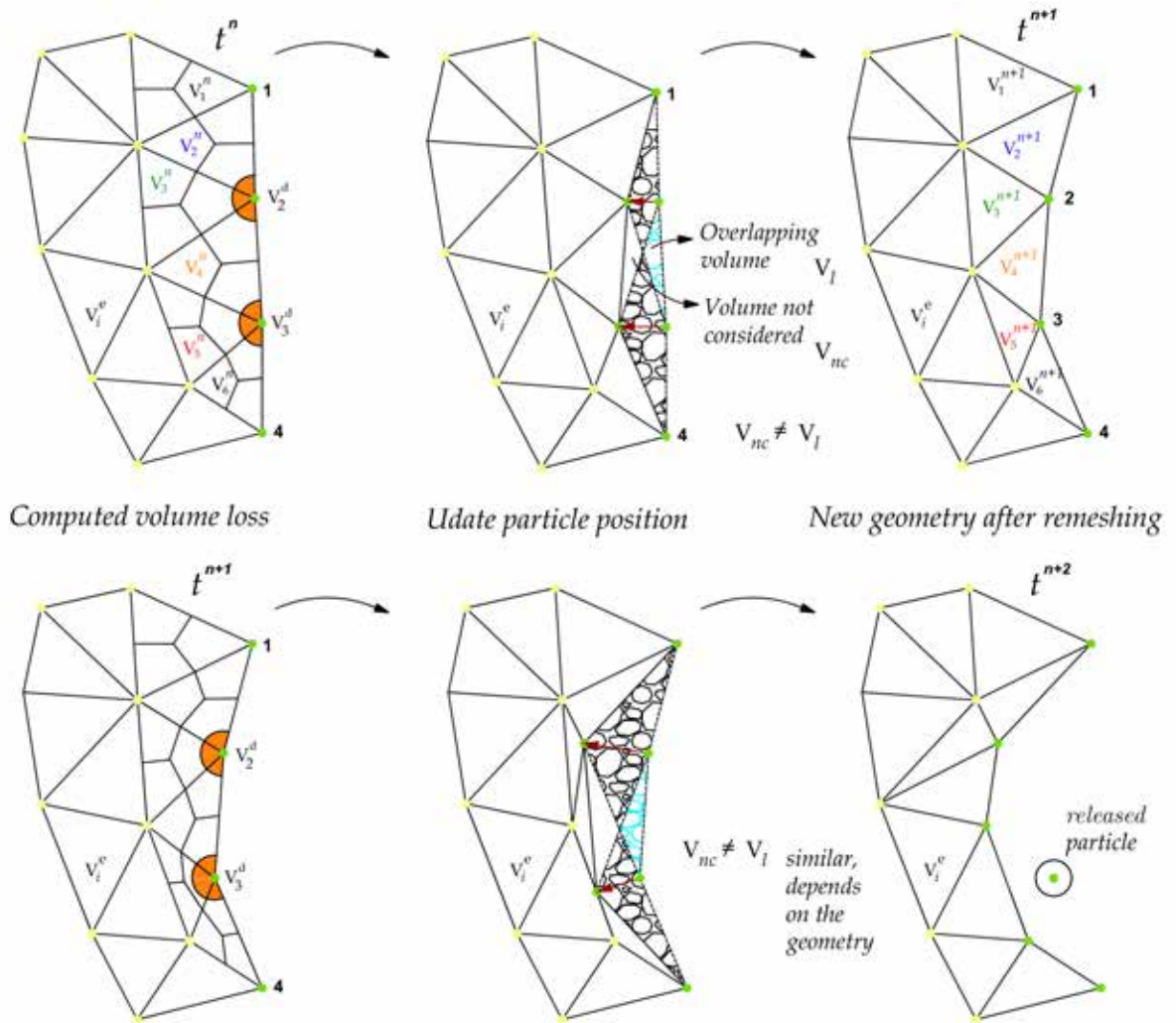
needs to be reduced. In 2D this volume corresponds to a triangle formed with the neighbor boundary particles. In 3D it fits to the pyramid formed with the neighbor boundary particles.

The process is carried on within the remeshing scheme of the PFEM. The new position of the particles defines the new boundaries automatically by means of the  $\alpha$ -shapes method, see Section 4.3.1. An intuitive scheme for surface shaping is presented in Figure 6.17.



**Figure 6.17:** Excavation strategy consisting in a *Shaping the Surface* using the computed volume loss

In this case the volume associated to the particles has not major influence. However, the scheme is not flawless in the volume adjustment. There are still some ill-adjustments which are depicted in Figure 6.18.



**Figure 6.18:** Error induced in the volume reduction due to the inwards movement of consecutive particles.

The strategy for shaping the solid surface also release particles when these particles are going to overlap the interior domain. In this case the *particle release* scheme is applied. Hence a combined method is implemented to update the geometry with the changes in volume produced by excavation.

The above explained scheme holds for wear and excavation. Usually, during excavation the cutting tools are worn out due to the operation. The surface that plays the role of

cutting, digs into the other surface. At the same time the excavating surface of the cutting tools is worn out by the abrasive excavated ground. Therefore, an excavation model and a wear model together must be included in the modeling of the problem.

All the figures shown are depicted in two-dimensions. The same shaping strategies are applied to two and three-dimensions. The integrated algorithm for the excavation in the PFEM is presented in Section 6.5.5.

### 6.5.3 Justification for the conservation of mass

When the particles of the domain are set free due to the wear and excavation they are released from the model. Hence the global volume of the problem changes as the number of particles is reduced. The historical values of the variables in these particles are lost as these particles do not contribute to the system anymore. The mass of the domain decreases with the volume reduction. This is a real consequence of the studied phenomenon, but can be considered as an inconsistency with the physics of the continuum formulation; see Section 2.3.1.

In fact the mass conservation is fulfilled. During the mechanical computation, within of each time step, all balance laws are preserved. Hence, the equilibrium of the physical system is reached satisfying all fundamental laws. Remember that the mechanical behavior of the continuum arises from these laws. Suppose now that there is an independent body in the domain, which has no mechanical influence on the rest of the bodies. The analysis can neglect the computation of that body assuming that it does not affect the mechanical behavior of the physical system. Then the continuum can be split to be computed separately and the parts which has no influence with others can be released. This is the same as is done in the PFEM for modeling excavation processes. The released particles are split from the main problem, because their kinematics are not useful and they are not computed anymore. Hence, to take away the excavated particles, and consequently some mass, is not an inconsistency as it has not any mechanical effect.

### 6.5.4 Constitutive models and excavation

Rocks and hard soils represent the most common sort of materials encountered in excavation problems. The standard behavior for these materials is a brittle constitutive behavior, hence they damage and fracture abruptly. In order to model this behavior a *damage model* is a good option. The constitutive laws related to damage models were introduced in Section 2.4.4. Tension and compression are treated differently by the model which describes the constitutive behavior of the ground.

Damage which fractures the ground due to excavation is considered within the excavation model. The boreability coefficient  $K_d$  includes this phenomenon (see equation (6.21)). This coefficient defines the material properties that quantify the damage produced on the surface of the geomaterial, but only in the cutting region. Note that the damage from the stresses generated in the excavation can be extended inwards and affect other parts of the excavated material. The damage not directly produced by the excavation crush is described by the constitutive model, but not by the excavation model.

When a region of the ground is damaged, automatically with a modification of the

boreability coefficient  $K_d$  is considered for this zone. Material properties change and is easier to dig on it. Therefore the damage model affects the excavation model by means of this coupling. The modification on the boreability coefficient is applied using the damage variable  $d$  as follows:

$$\hat{K}_d = \frac{K_d}{1-d} \quad \text{with} \quad d \in [0, 1) \quad (6.29)$$

where  $\hat{K}_d$  includes the damage influence in the volume loss rate. Using the general form of equation (6.10) and introducing the damage influence yields

$$V_d = \hat{K}_d \frac{F_N g_T}{H_d} = \left( \frac{1}{1-d} \right) K_d \frac{F_N g_T}{H_d} \quad (6.30)$$

The layer of elements in the contact surface is not described by the geomaterial constitutive law. Instead of that, an elastic layer is used. This is applied in order to avoid the local effects that contact forces produce in contact surfaces. Damage produced in the surface area is computed by the excavation model. Once an internal damaged element is reached by the excavation front, the amount of accumulated damage is accounted for in the model by using equation (6.29).

Similarly as for worn particles, totally damaged particles do not contribute to the mechanical behavior of the domain. Therefore when particles on the body surface reach the maximum damage level ( $d = 1$ ) they are removed from the analysis domain.

### 6.5.5 Algorithm for excavation with the PFEM

**Wear and Excavation Algorithm:**  
**"Surface shaping" including "Particle release"**

1. Volume loss distribution:

Compare volume loss of a particle  $p$  :  $(V_p^{wear} + V_p^{dig})$  (wearing+digging) with the volume associated to this particle:  $V_p$

$$V_p^{rest} = V_p - (V_p^{wear} + V_p^{dig})$$

the rest of volume loss  $V_p^{rest}$  is distributed to neighbor particles.

2. Compute direction for geometry reduction  $\mathbf{n}_{shape}$ :

$$\mathbf{n}_{shape} = \mathbf{n}_c + \mathbf{v}_\sigma$$

where  $\mathbf{n}_c$  is the particle contraction vector and  $\mathbf{v}_\sigma$  is the direction of the particle main stresses.

3. Compute the volume reduction due to particle the new particle position:

**2D:**

Reduced volume from the triangles formed with neighbor particles  $V_p^{geo}$

**3D:**

Reduced volume from the pyramids formed with neighbor particles  $V_p^{geo}$

4. Determine the distance reduction:

Comparing  $V_p^{geo}$  and  $V_p^{loss}$  the new position of the particle is determined.

$$\mathbf{p}_p = d_s \cdot \|\mathbf{n}_{shape}\|$$

where  $d_s$  is the distance reduction and  $\mathbf{p}_p$  the new particle position.

5. Check the volume of neighbor elements:

- If any element which contains that particle has a less than 10% of its volume consumed, the particle  $p$  is released. Then neighbor particles are marked as a new boundary.
- In the contrary the particle is moved to  $\mathbf{p}_p$ , any new particle is marked as boundary.

Box 6.1: Flowchart for shaping the worn and excavated surfaces.

## 6.6 Excavation scheme

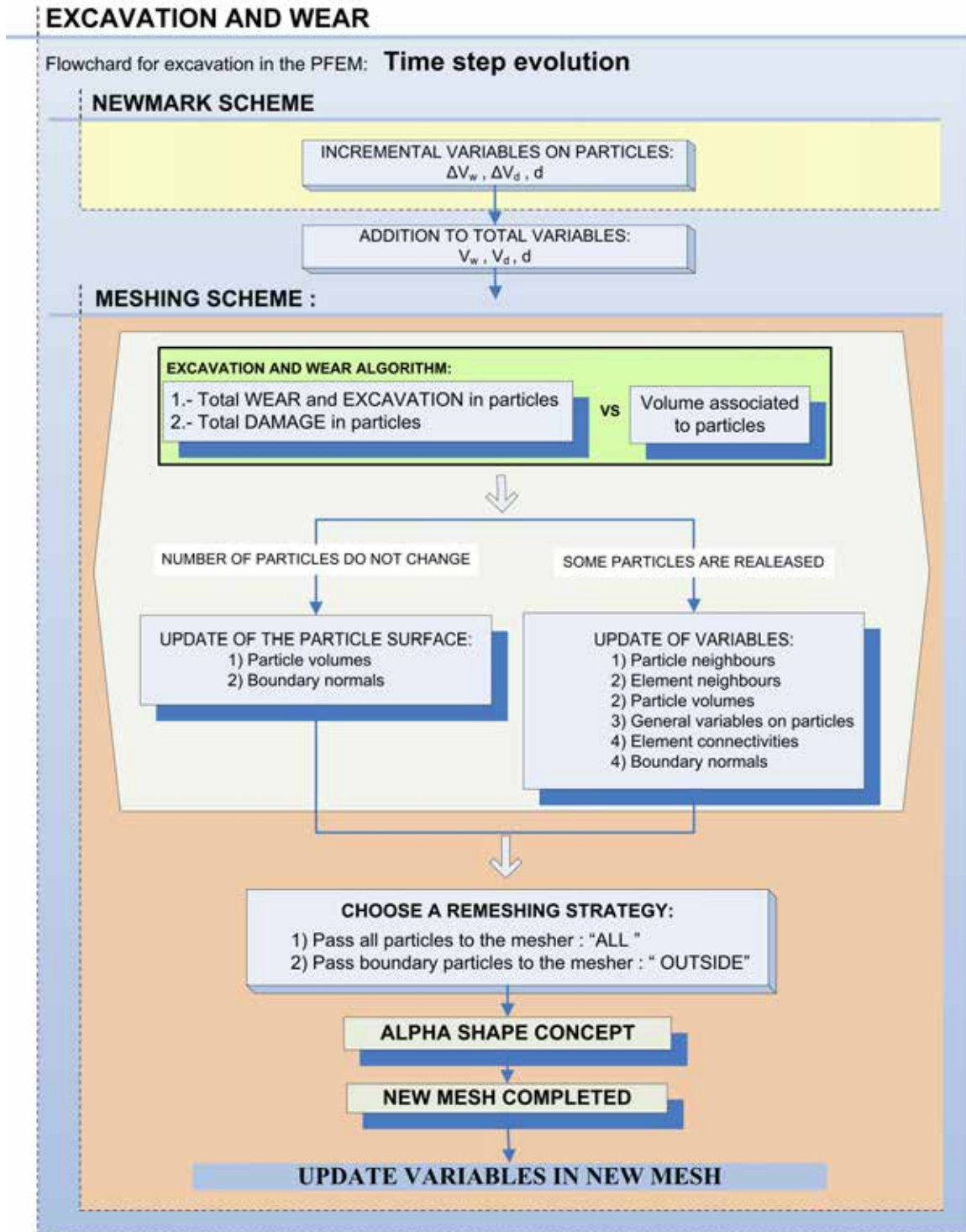


Figure 6.19: Excavation scheme for the PFEM



## 6.7 Examples

Some qualitative examples are presented next. The objective is to show the functionality of the method for modeling ground excavation processes. The application of the PFEM to excavation is the result of the research and new developments presented in an integrated solution scheme. All these new features are implemented in a single code. This code has also been tested in previous chapters with contrasted numerical examples.

The conjunction of all of these developments creates a new challenge in terms of calibration and testing. At this time, there is not a background of benchmarks for excavation and wear processes. This is a new task that needs the support of experimental testing. This is the next step after the numerical modeling research in order to make the method reliable for engineering applications.

For the above reasons, the examples presented are more qualitative than quantitative. However, the performed analyses tried to be as quantitative as possible, although in most of them it is not possible to compare with experimental data. Therefore, examples shown next are designed mainly with the purpose of showing the functionality and the capabilities of the PFEM for modeling excavation problems.

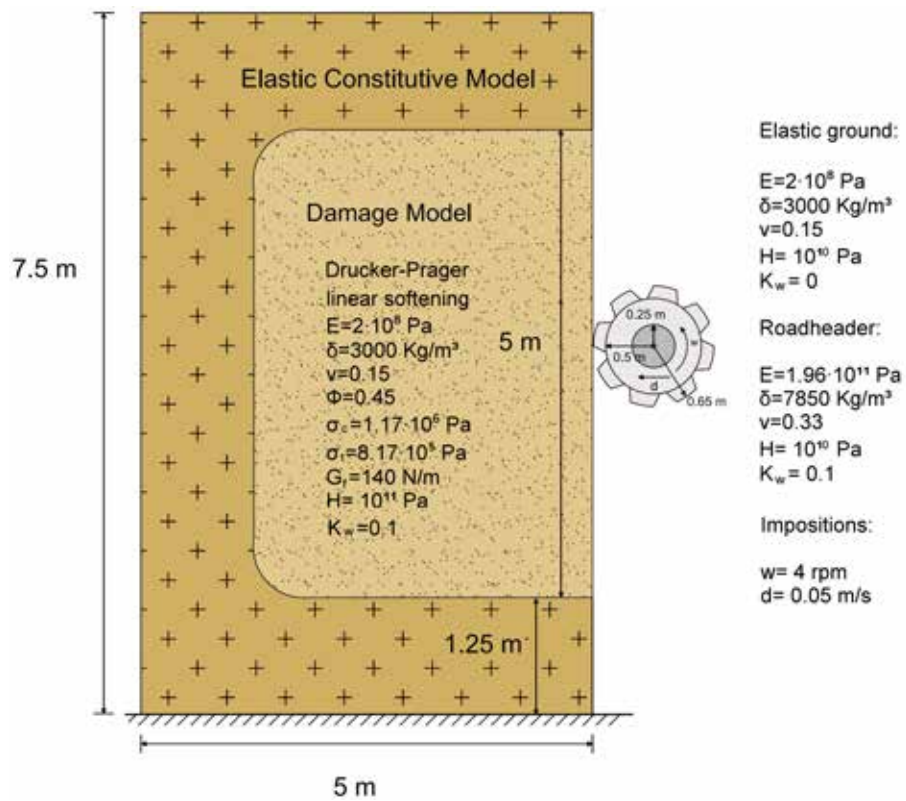
### 6.7.1 2D Excavation with a roadheader

The first example of application of the PFEM is an elastic roadheader in 2D. Usually excavation problems are fully 3D. However, some simplifications can be made to model the problem as 2D. The roadheader is composed by a circular center which has an imposed rotation and displacement. This transfers the rotation to the dentated ring of the roadheader which generates friction when contacting with the ground. Figure 6.20 shows the 2D model with the material properties and boundary conditions.

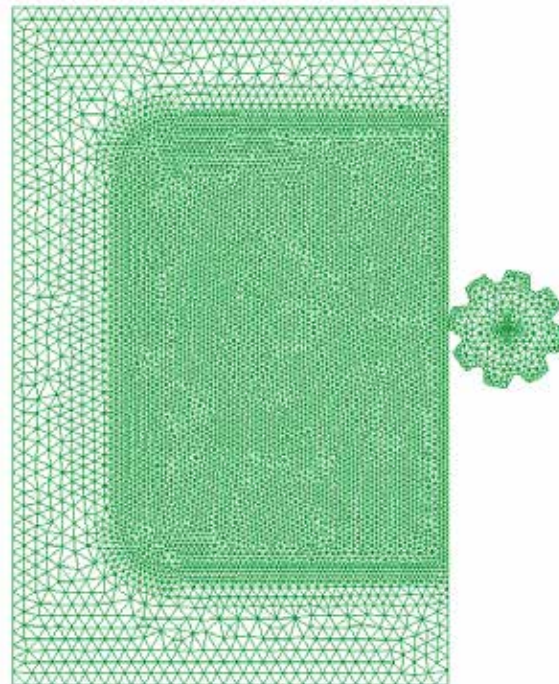
The problem is modeled as a dynamic interaction between two continuum domains. A small *Rayleigh* damping has been considered in order to reduce the frequencies induced for the excavation impacts. For the elastic ground material *Rayleigh* parameters are  $a_1 = 0$  and  $a_2 = 0.05$  and for the roadheader  $a_1 = 0$  and  $a_2 = 0.01$  a (see equation (3.40)). The geomaterial is modelled with a damage constitutive law using a Drucker-Prager damage surface and linear softening. The *Rayleigh* damping considered for this material is  $a_1 = 0$  and  $a_2 = 0.01$ .

The wear coefficient  $K_w$  and the material hardness  $H$  are defined in the properties of each material. A model of cutting picks has been considered on the surface of the roadheader (see equation (6.26)). The hardness of the geo-material is  $H_{dp} = 10^7 Pa$  and the boreability coefficient  $K_{dp} = 2500$ . The friction coefficients are  $\mu_s = 0.3$  and  $\mu_d = 0.25$ . The thickness of the model is  $t = 1 m$ .

Figure 6.21 shows the initial mesh used in the model. It has 15498 triangles and 7919 nodes. The time step defined for the implicit integration is  $\Delta t = 0.005 s$ .



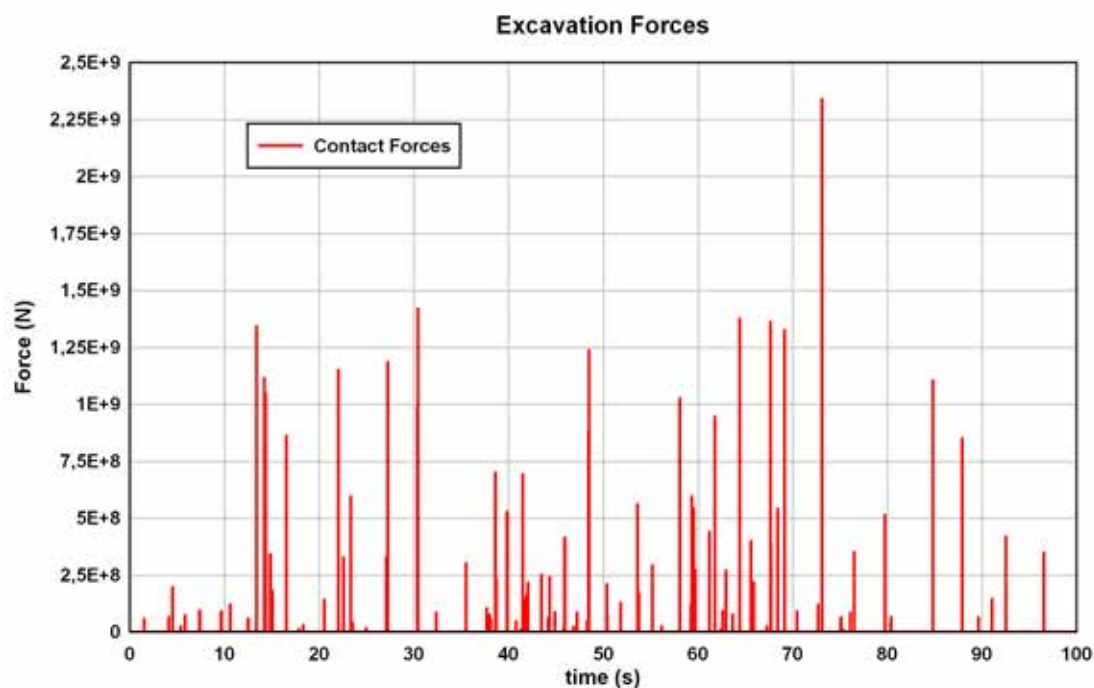
**Figure 6.20:** 2D model of an excavation problem with a roadheader.



**Figure 6.21:** Initial mesh for the model of a 2D roadheader and the ground. The mesh has 15498 triangular elements and 7919 nodes.

Contact occurs when the disc edges comes near the ground wall. An interface mesh of contact elements is generated and it anticipates the contact area. The contacting forces are transmitted through the contact elements to each domain. This interaction damages the solid material and digs in it. With the normal contact force and the relative sliding velocity an excavated volume is computed for each surface particle. The geometry of the ground is shaped at the same time the excavation moves forward.

Contact forces are the result of several impacts on the ground surface. From that, a very irregular force diagram is obtained. High force picks represent the sudden impacts on the ground surface. Figure 6.22 shows the computed forces.

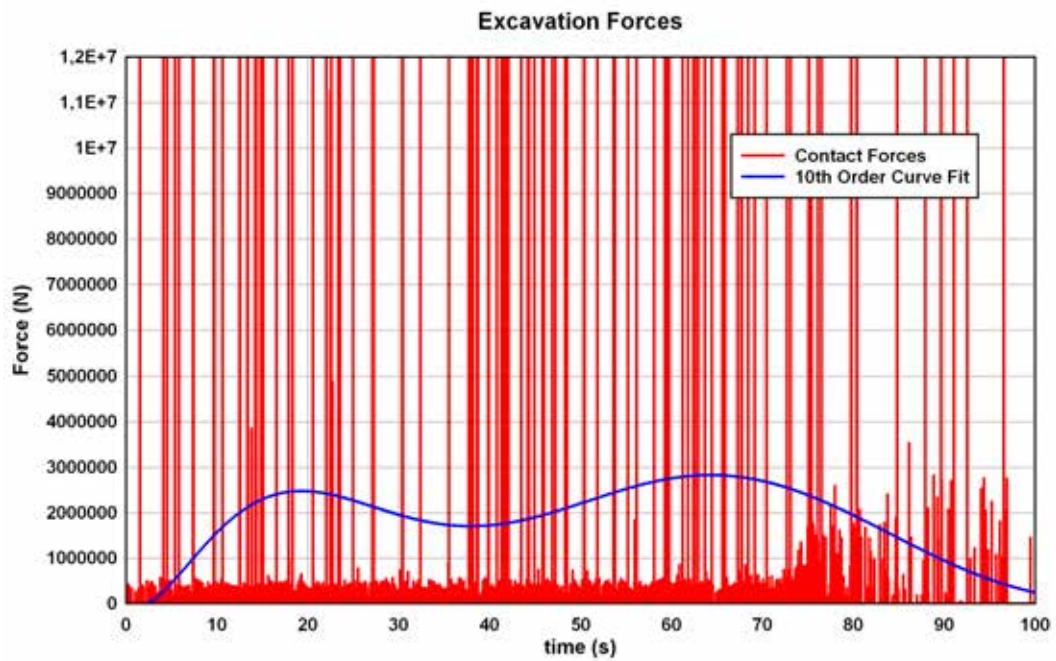


**Figure 6.22:** Computed contact forces on the 2D roadheader.

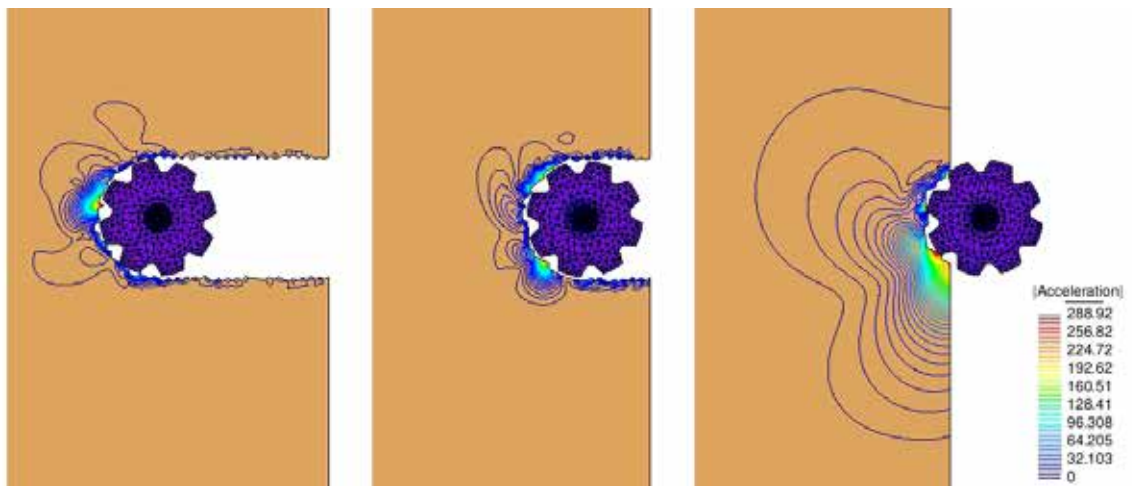
It can be observed that the magnitude of some of that forces is extremely high. This is because this is not a real excavation. In the model, the roadheader excavates 5 meters deep in less than 2 minutes. To achieve that production large forces are needed. Another handicap is the point-wise and sudden nature of the contacts, that makes difficult the estimation of the correct force. These large values can be globally averaged and an evolution of the contact interaction obtained; see Figure 6.23.

The resultant of the surface forces can be computed in the axis of the roadheader in order to yield force and momentum reactions. This gives a practical information of the power and torque needed for the excavation.

Figure 6.24 shows the excavation in initial, intermediate and final states. The impacting forces are depicted using the resulting accelerations in the excavation front.

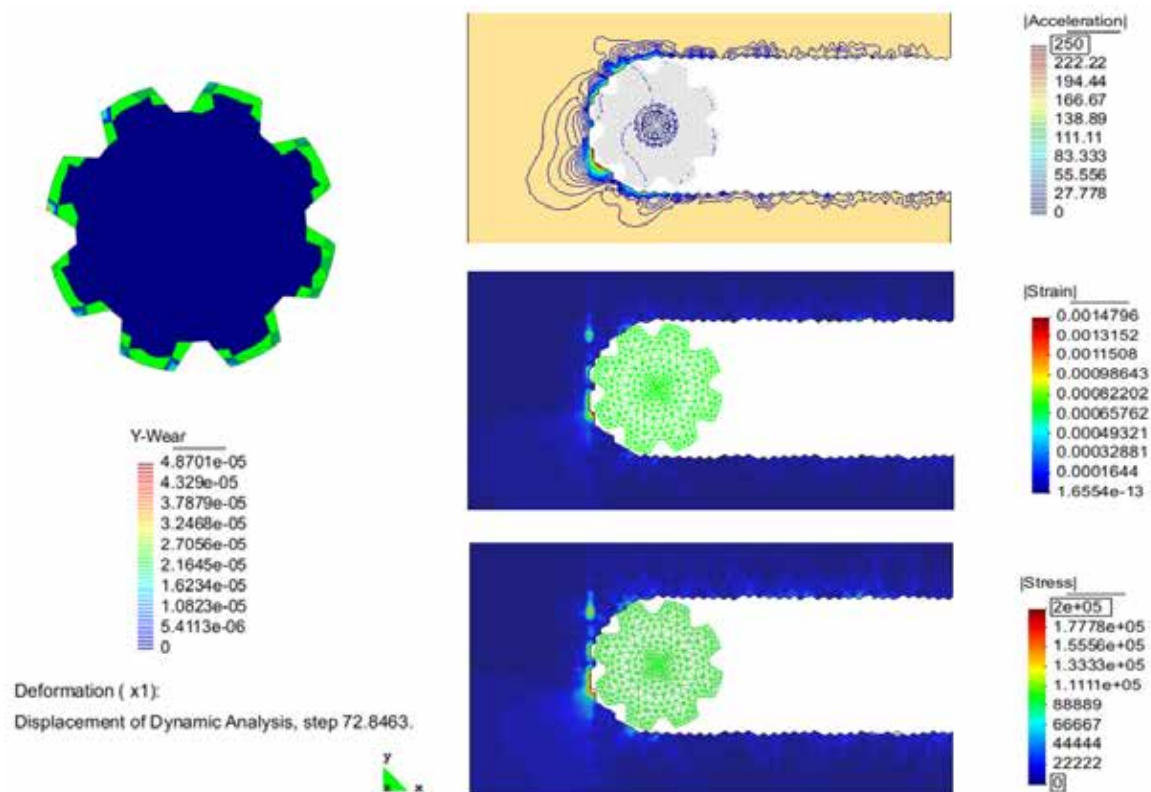


**Figure 6.23:** Computed contact forces on the 2D roadheader and averaged contact interaction curve.



**Figure 6.24:** 2D excavation with a roadheader.

Figure 6.25 shows the results after approximately 72 seconds of analysis. Stresses, strains and accelerations in the ground are depicted as well as wear produced on the surface of the roadheader after 4 revolutions of the tunneling machine.

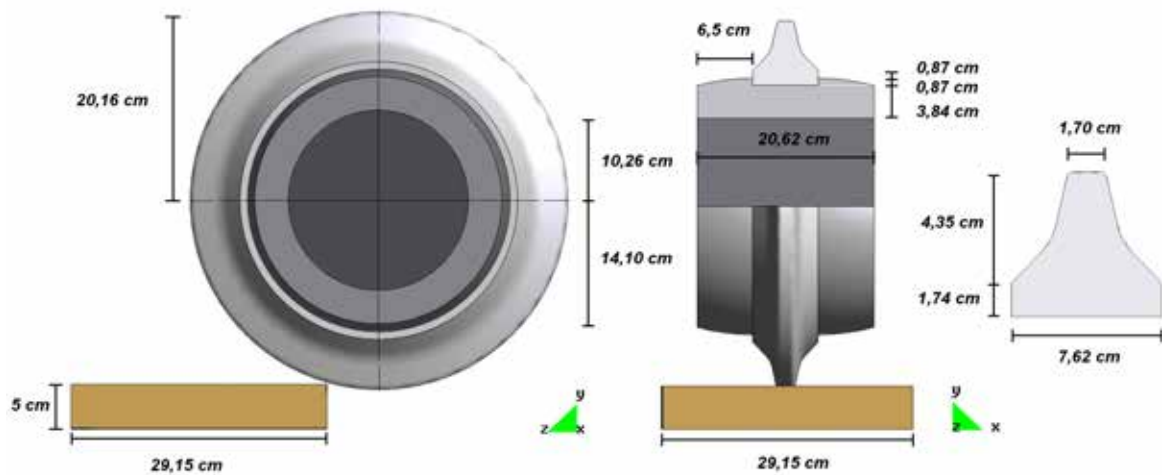


**Figure 6.25:** Stresses, strains and accelerations on the ground due to excavation. Accumulated wear on the roadheader surface at the end of the simulation.

### 6.7.2 Disc linear cutting test

This is an example of a standard linear cutting test for a disc of a TBM machine. The geometry of the disc and the modeled ground is shown in Figure 6.26. The geo-material is fixed on the base and the disc has an imposed velocity in the rotation axis of  $\mathbf{v} = (0, 0, 2) \text{ m/s}$ . The imposed linear velocity coincides with the standard value for rotational velocity of a tunneling machine.

In this example the geo-material considered has been modeled using two different constitutive laws: an elastic model and a damage model. The purpose is to compare the obtained results and see the differences obtained using a different constitutive behavior for the continuum. The material properties are defined in Table 6.9.



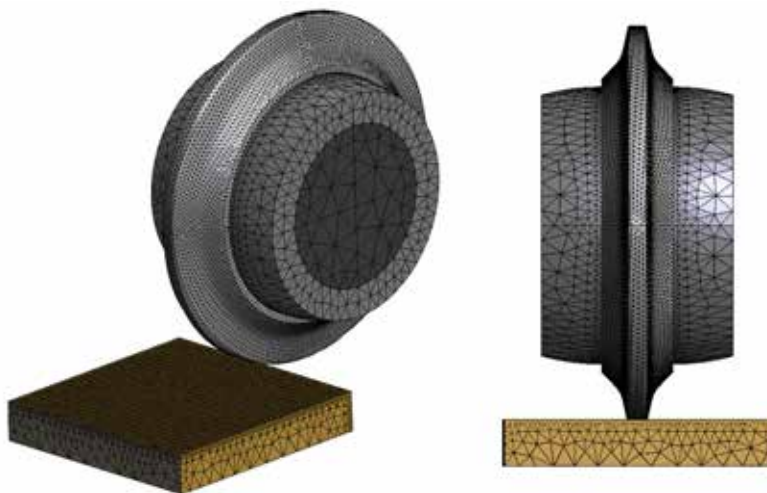
**Figure 6.26:** Geometry of the cutting disc.

The model is discretized using a unstructured mesh of tetrahedra. The initial mesh of model is shown in Figure 6.27. The ring of the disc and the ground surface are discretized with finer meshes compared with other elements of the model. The total mesh has 243835 tetrahedra and 51109 nodes. These nodes are the particles used along the analysis as a reference for the geometrical and mechanical definition of the problem. The time step for the implicit integration is  $\Delta t = 0.001 \text{ s}$ .

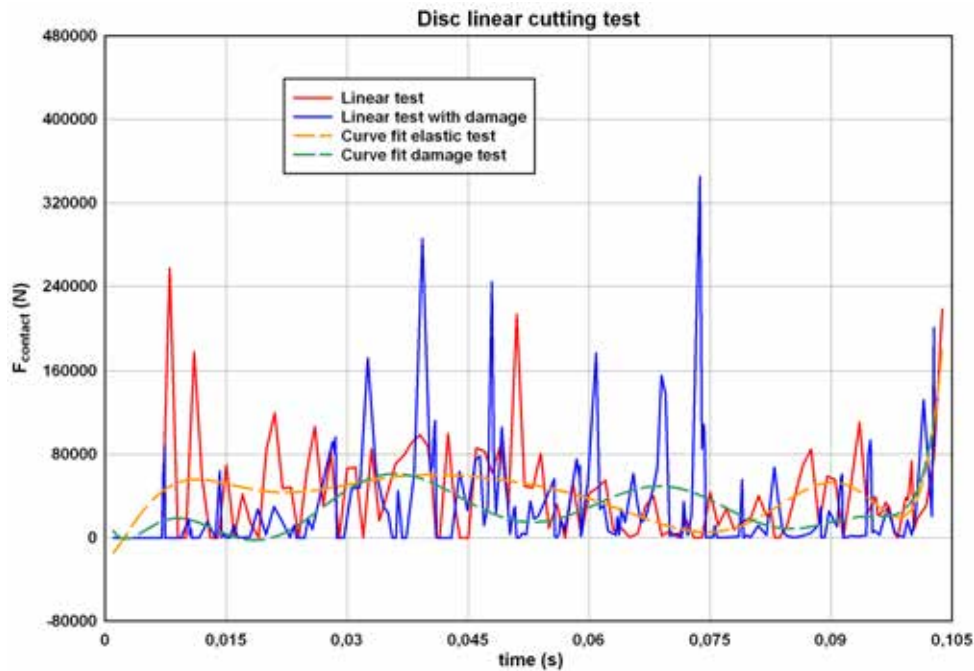
Some results are presented here as a comparison of the cutting test after 10 cm cutting. Firstly the contact forces are displayed in Figure 6.28. An irregular contact behavior is obtained in both analyzed cases. The force diagram shows that the excavation is a consequence of a group of impacts between the disc and the ground. The force distribution is different considering damage in the ground and not considering it. Comparing with experimental results the obtained average contact force is larger than the one computed by the example. This can be an incidence of the bad contact force distribution which can be improved with a finer discretization of the model.

**Table 6.9:** Material properties for the linear cutting test example

Materials	Ground Damage	Ground Elastic	External Ring	Internal Ring	Disc Center
E (Pa)	$28,4 \cdot 10^9$	$28,4 \cdot 10^9$	$4,96 \cdot 10^{11}$	$1,96 \cdot 10^{11}$	$4,96 \cdot 10^{11}$
$\nu$	0,29	0,29	0,33	0,33	0,33
$\delta$ ( $kg/m^3$ )	3500	3500	7850	7850	100
H (Pa)	$10^{10}$	$10^{10}$	$10^{10}$	$10^{10}$	$10^{10}$
$K_w$	2	2	0,5	0,5	0,5
Damage Surface	Mohr-Coulomb	-	-	-	-
Softening	linear	-	-	-	-
$\phi$ (Pa)	0,30	-	-	-	-
$\sigma_c$ (Pa)	$6,17 \cdot 10^7$	-	-	-	-
$\sigma_t$ (Pa)	$1,17 \cdot 10^6$	-	-	-	-
$G_f$ (Pa)	250	-	-	-	-

**Figure 6.27:** Mesh discretization of the disc and the ground. Initial mesh of 243835 elements and 51109 nodes, including both domains.

Excavating forces are a direct consequence of the contact set of active elements. In Figure 6.29 the number of active contact elements of the analysis is depicted in function of time. It can be observed also a very irregular oscillating curves. This tells us that the contact area is not homogeneous and it changes a lot during the excavation. The cutting forces are applied to singular points on the interacting zone. It is important to notice



**Figure 6.28:** Contact force comparison for the linear cutting test with and without ground damage.

that a larger number of contact elements does not mean bigger contact forces. There is no correspondence between Figures 6.28 and 6.29.

Critical points occur when the active set is zero. This occurs as a consequence of the particle release and produces detachment between surfaces which is not realistic. Usually some crashed material remains on the cutting zone keeping the contact active. It seems that when damage is active the active set remains more stable. This can be a consequence of the loss of resistance on the ground contact zone that allows a larger deformation.

One of the major interests of the analysis is to get the correct wear on the cutting disc. This has a direct relationship with the force computation and the relative velocity. Figure 6.30 depicts the instantaneous volume loss due to wear on the disc. It can be observed, comparing Figures 6.29 and 6.30, that a large amount of wear is produced after losing contact in between disc and ground. This situation produces an increase of the relative velocity between contact domains, as they are set free. At the same time, when they contact again, a large contact force is produced. Joining these two phenomena the volume loss due to wear increases.

The sum of instantaneous wear on the disc surface is depicted in Figure 6.31. The linear cutting test considering an elastic ground has a linear regular increase of the wear on the disc surface. In the case of ground damage the wear is larger and has a more irregular slope.

A comparison of the stresses and strains on the ground is presented in Figure 6.32. Similar values are obtained in both cases. Some different distributions can be observed on



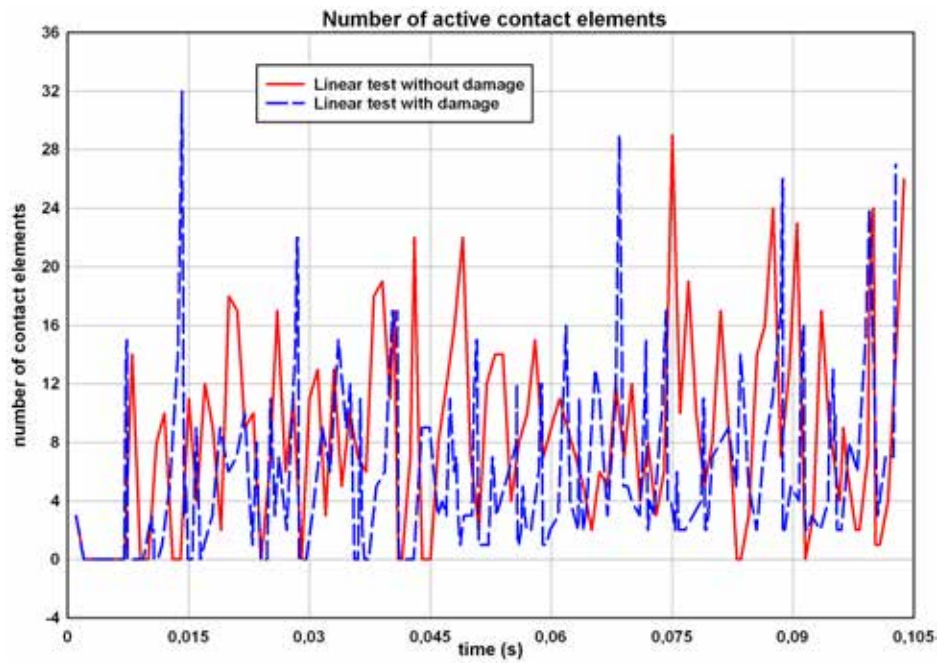


Figure 6.29: Number of active contact elements along the analysis.

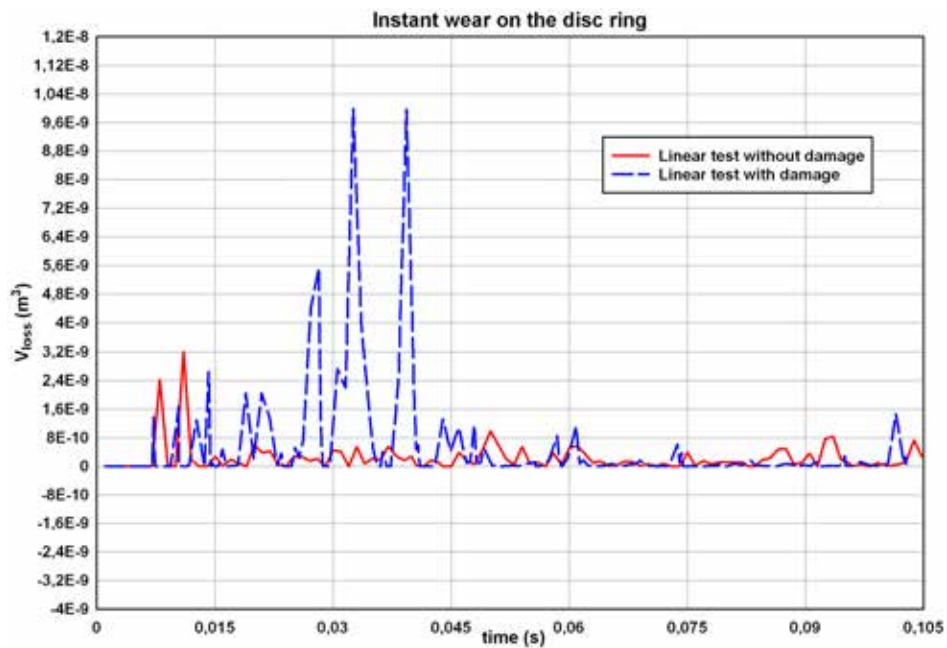
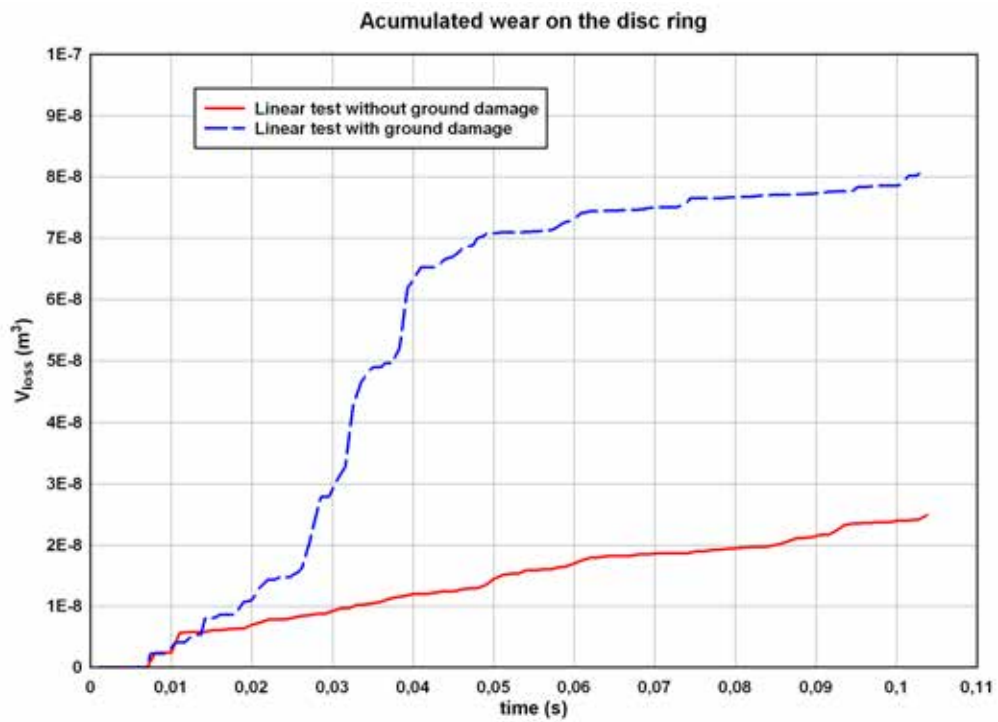


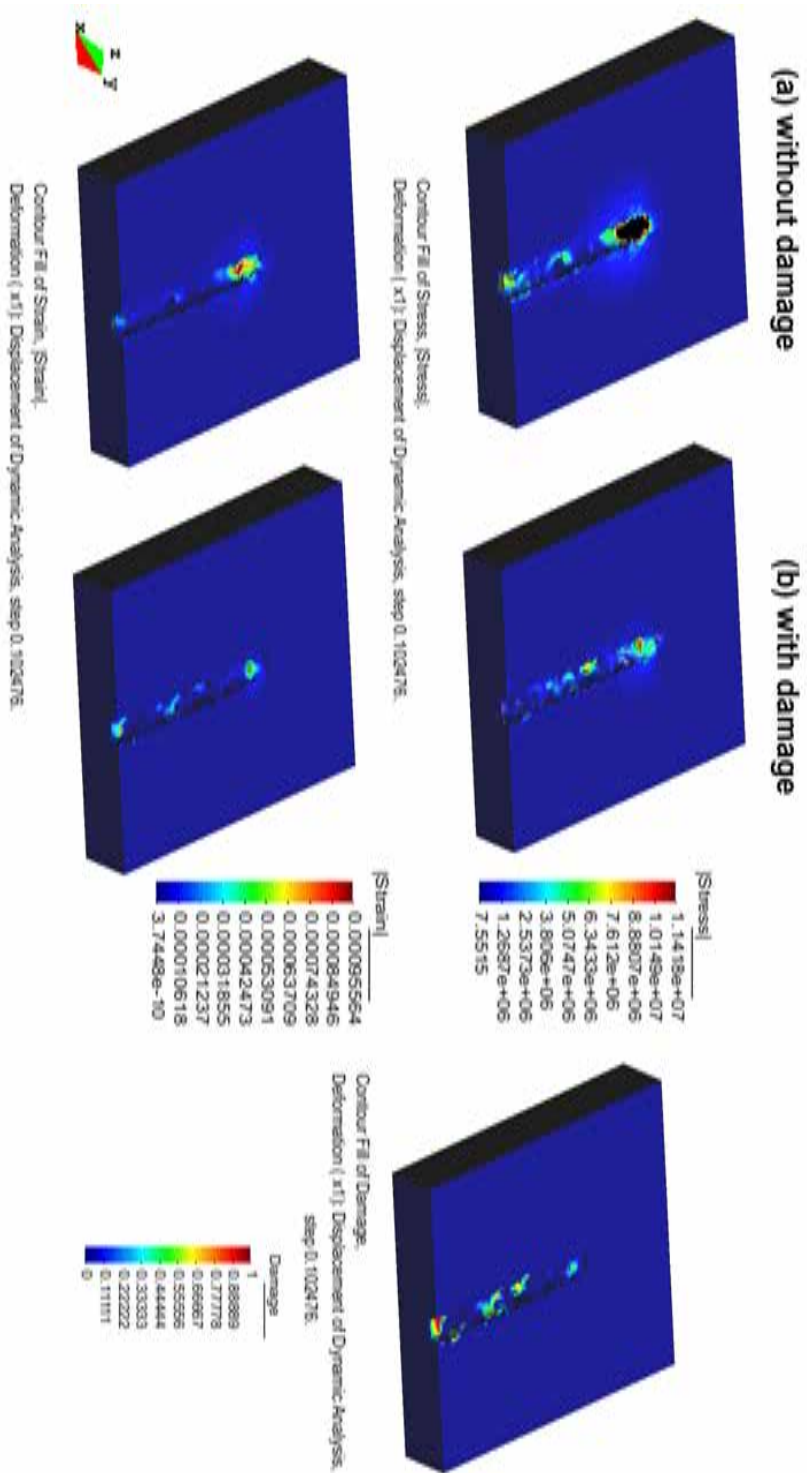
Figure 6.30: Instantaneous wear computed on the surface of the cutting disc.



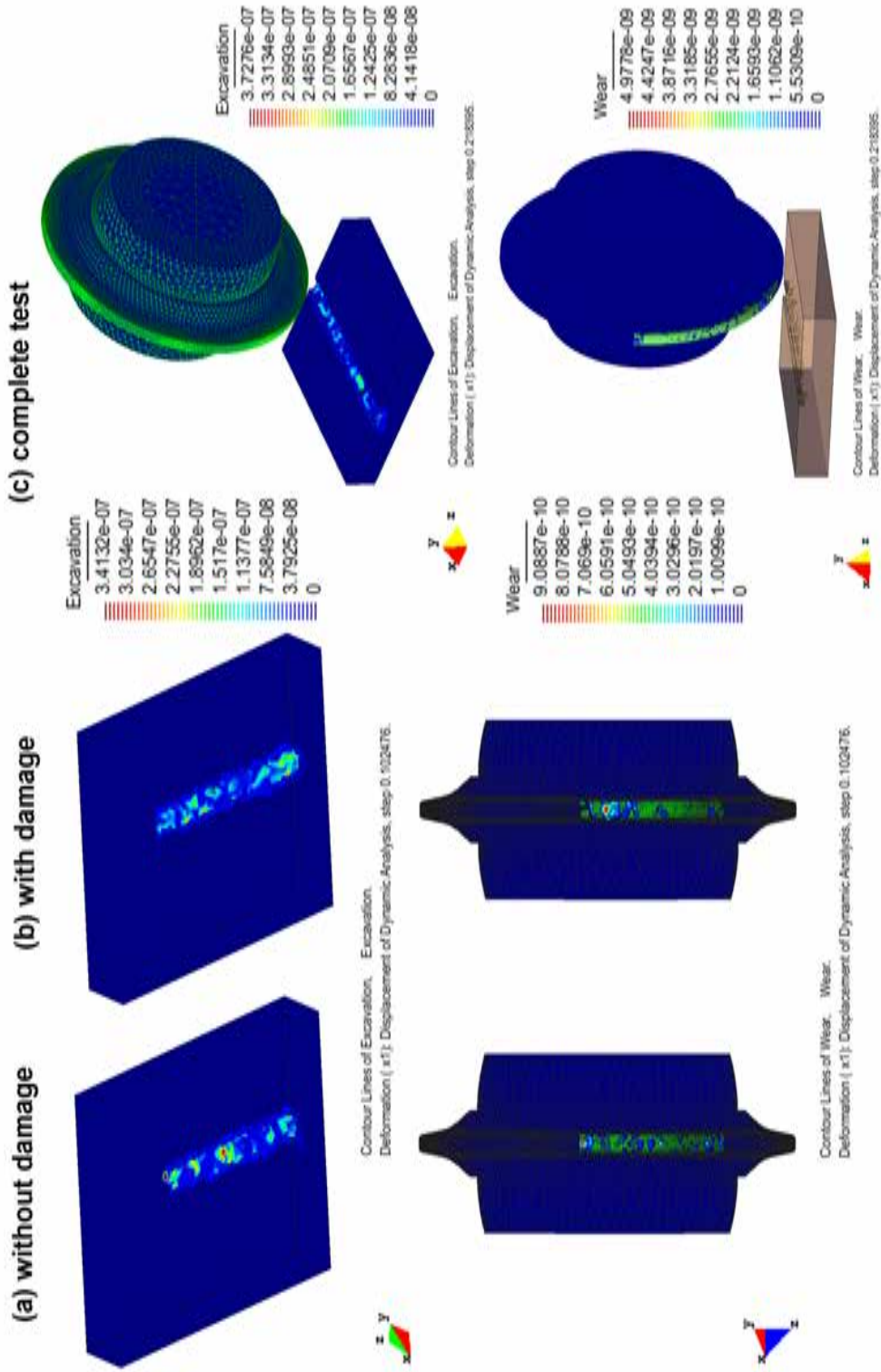
**Figure 6.31:** Acumulated wear computed on the surface of the cutting disc.

the ground incision.

Figure 6.33 shows the wear on the disc and the excavation rate on the ground. The complete test with the cut piece of geo-material is also presented in this figure.



**Figure 6.32:** Stresses, strains and damage on the ground after 10 cm cutting. The disc is hidden but in fact is pushing the last part of the cutting channel.



**Figure 6.33:** Wear on the ring of the disc and rate of excavated volume on the ground with (b) and without (a) a damage constitutive law for the ground modeling. (c) Final state after cutting the whole piece of geo-material.

### 6.7.3 3D Excavation with a roadheader

In this example an excavation with a roadheader is modeled in 3D. The simulation considers a ripping process. Firstly the ripping head is pushed towards the rock-solid with a constant rotation. After 18 seconds, when it has excavated a considerable portion of the rock, it moves over a lateral part. To reproduce the work an imposed rotation of 2 rpm is imposed at the center of the roadheader during the entire analysis. An imposed velocity of  $\mathbf{v} = (0, 0, -0.01) \text{ m/s}$  is applied at the first 18 seconds and a velocity of  $\mathbf{v} = (0.01, 0, 0) \text{ m/s}$  in the following time steps. All impositions are applied in the cylindrical axis of the machine only. The external part of the roadheader is set free. The geometry of the model is displayed in Figure 6.34. The rock-solid is fixed on the base. Points in the lateral walls are allowed to move vertically.

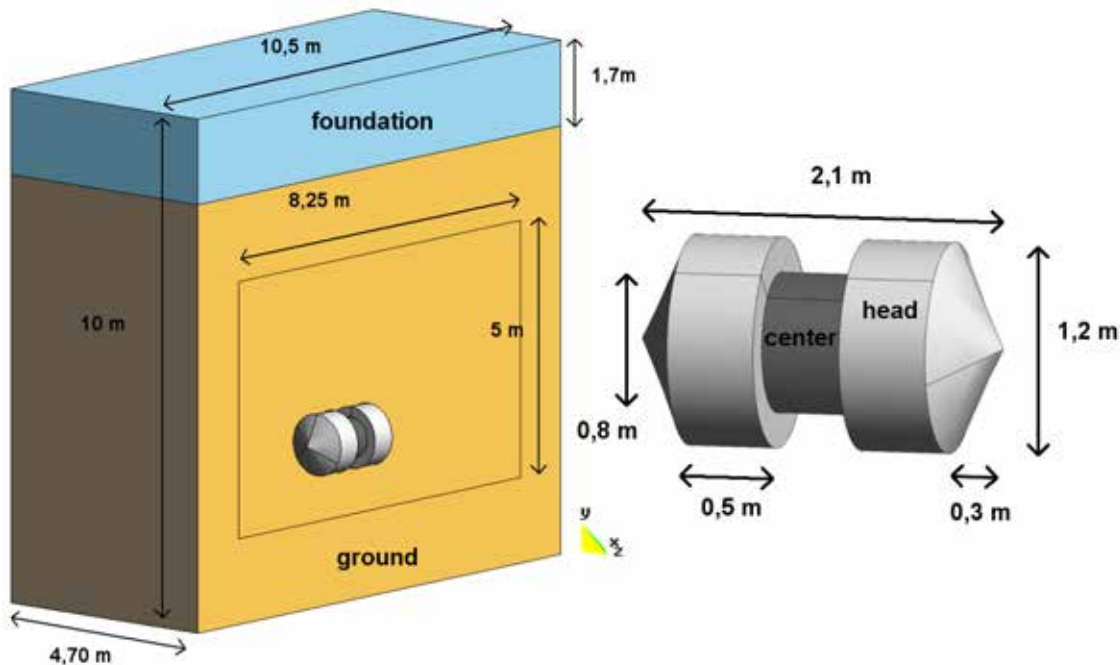


Figure 6.34: Model of a roadheader cutting a ground.

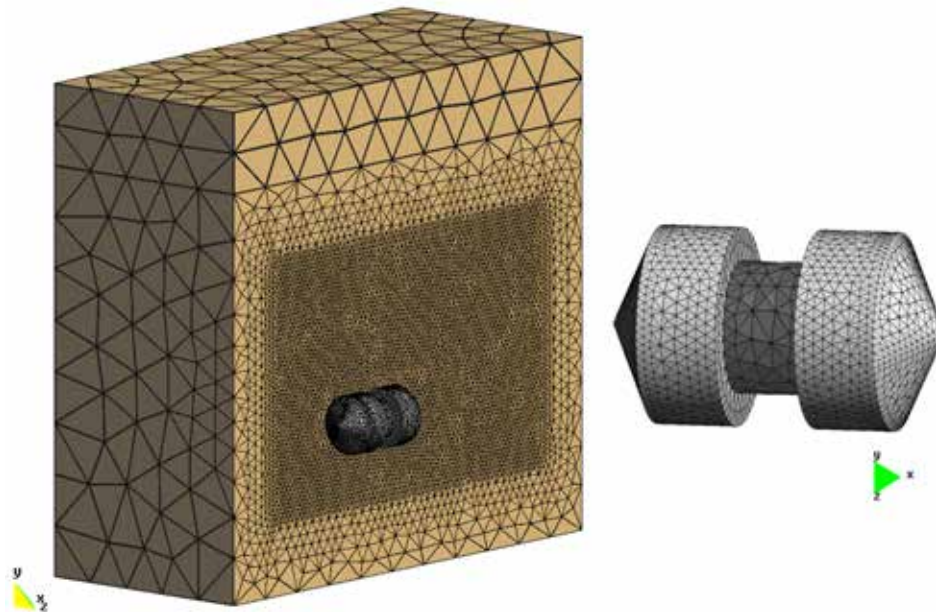
Table 6.10 contain all material properties of the model. The wear coefficient  $K_w$  and the material hardness  $H$  are defined as properties of each material (see Table 6.10). A model of cutting picks has been considered on the surface of the roadheader (see equation (6.26)). The hardness of the geo-material is  $H_{dp} = 0.5 \cdot 10^7 \text{ Pa}$  and the boreability coefficient  $K_{dp} = 100$ . These properties are assigned to the surface of the roadheader head considering 50 picks on it.

Figure 6.35 shows the discretization chosen. The initial mesh has 270463 tetrahedra and 52551 nodes. The time step used in the analysis is  $\Delta t = 0.04 \text{ s}$ .

The simulation permits to characterize some important excavation variables. Firstly,

**Table 6.10:** Material properties for the roadheader excavation example

Materials	Rock	Elastic Foundation	Roadheader Head	Roadheader Center
$E$ (Pa)	$20,5 \cdot 10^9$	$1,96 \cdot 10^{10}$	$1,96 \cdot 10^{12}$	$1,96 \cdot 10^{11}$
$\nu$	0,15	0,33	0,33	0,33
$\delta$ ( $kg/m^3$ )	2500	4500	7850	7850
$H$ (Pa)	$10^{10}$	$10^{10}$	$10^{10}$	$10^{10}$
$K_w$	0,5	0,5	0,1	0,0
$\mu_d$	0,2	0,2	0,2	0,2
$\mu_s$	0,3	0,3	0,3	0,25

**Figure 6.35:** Discretization of the roadheader excavation model. The initial mesh has 270463 tetrahedra and 52551 nodes. The used time step is  $\Delta t = 0.04$  s

the most important unknowns are the contact forces. Figure 6.36 shows the evolution of the normal and tangential forces. It can be observed that the force prediction is very irregular and represents the response of several discrete impacts. An smoothed evolution of the forces is depicted for each case in Figure 6.36.

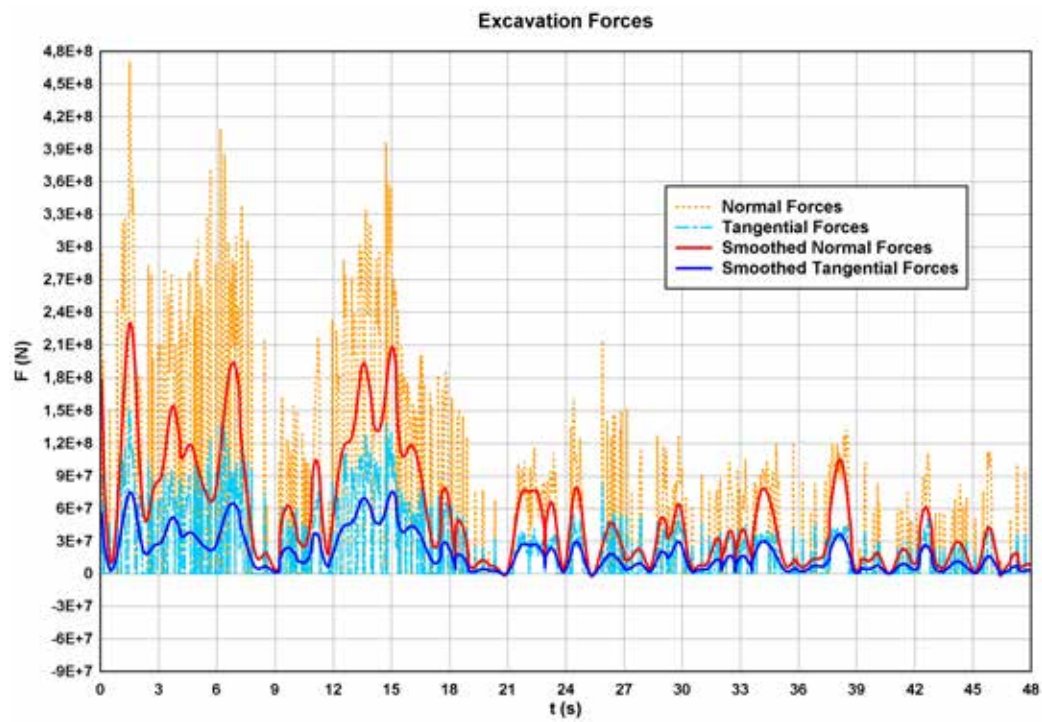


Figure 6.36: Evolution of the contact forces at the excavation front.

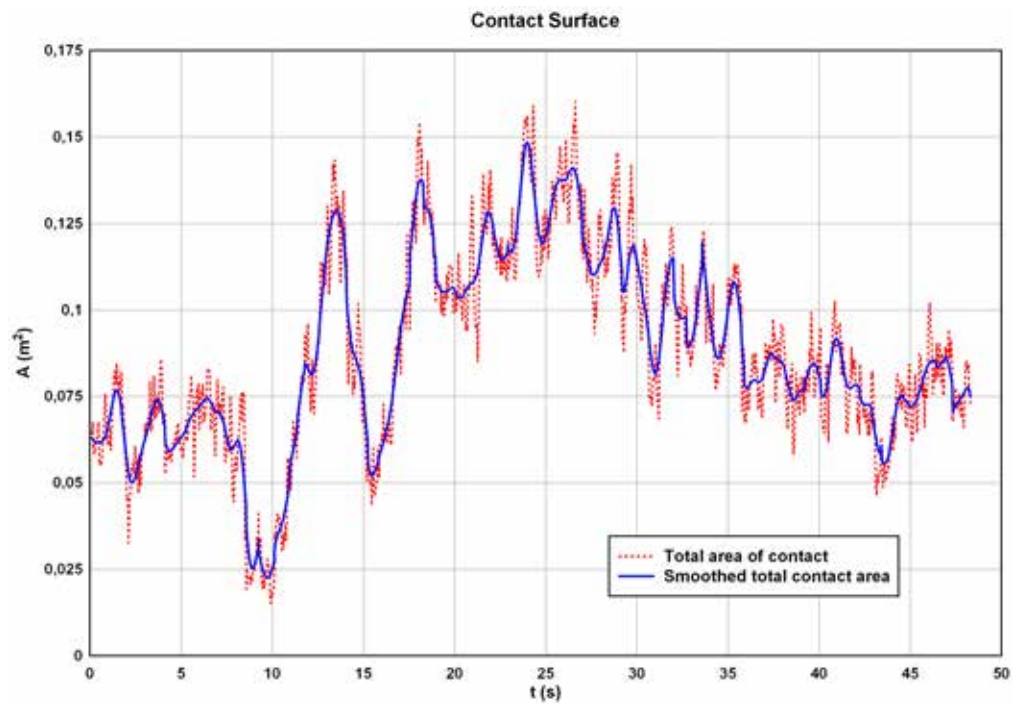
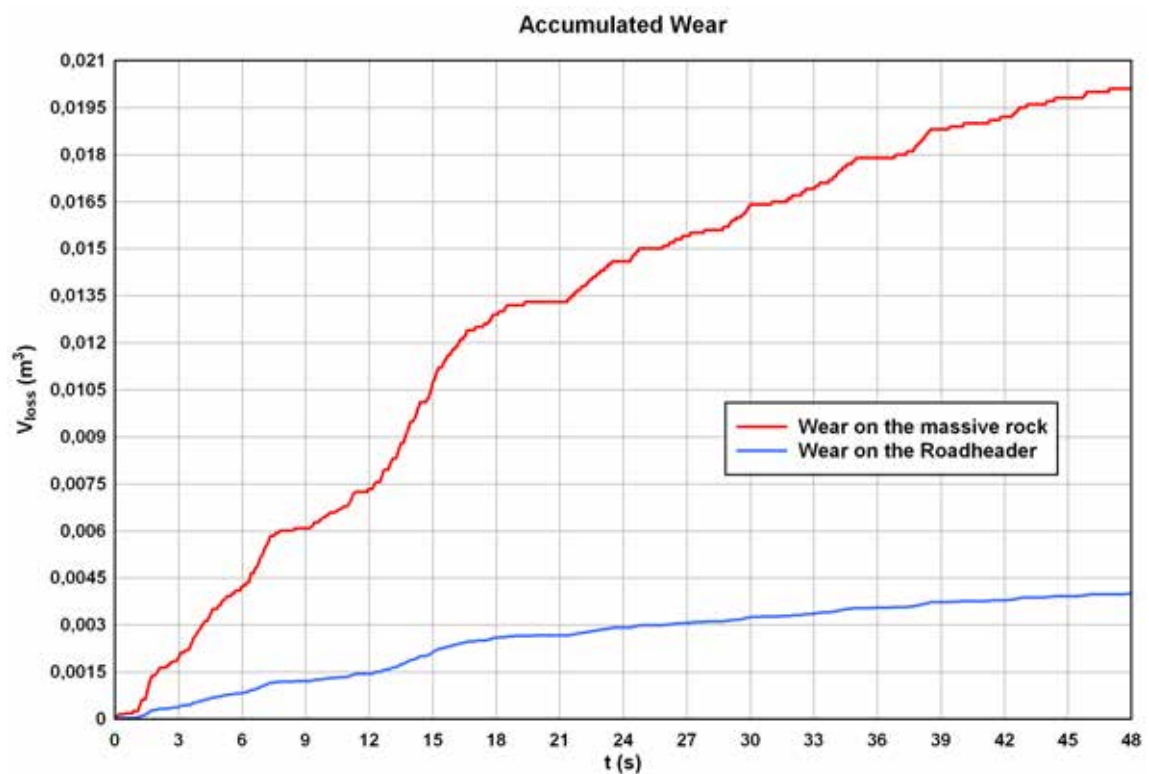


Figure 6.37: Evolution of the contact surface.

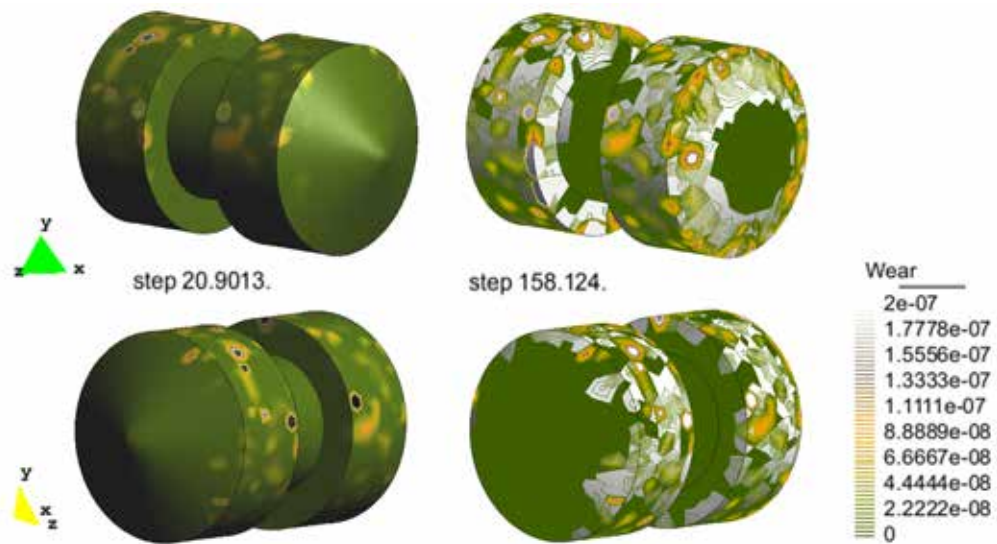
The contact area between the roadheader and the massive rock is depicted in Figure 6.37. The evolution of wear in terms of volume loss has the progression shown in Figure 6.38. Figure 6.39 shows the distribution of wear on the head of the roadheader after the initial penetration on the massive rock and some seconds later when it rips onto the left side.



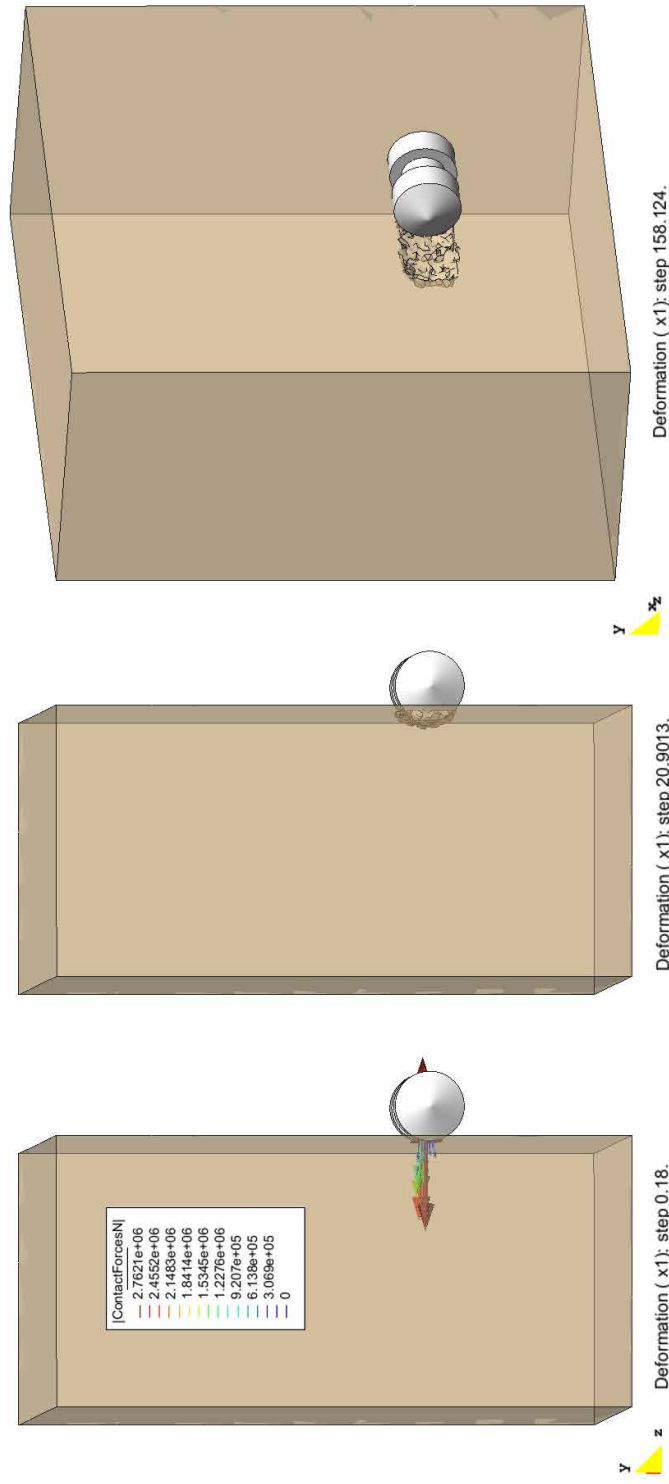
**Figure 6.38:** Wear progression in time due to the contact interaction machine-ground.

The initial contact forces and the geometric profile that results from the excavation are presented in Figure 6.40. Excavation forces generate accelerations on the rock which are transmitted to the surface. The foundation located on the top of the model vibrates due to the excavation. Figure 6.41 shows the produced acceleration for different stages of the excavation. This information is very useful in order to predict possible damages in the buildings of the surface.





**Figure 6.39:** Wear on the cutting tool after 20.9 seconds penetrating on the massive rock and at 158.12 seconds, when the roadheader is moving towards the side.



**Figure 6.40:** Normal forces in the initial contact. Excavation profiles after the roadheader penetration and after ripping along the rock front.

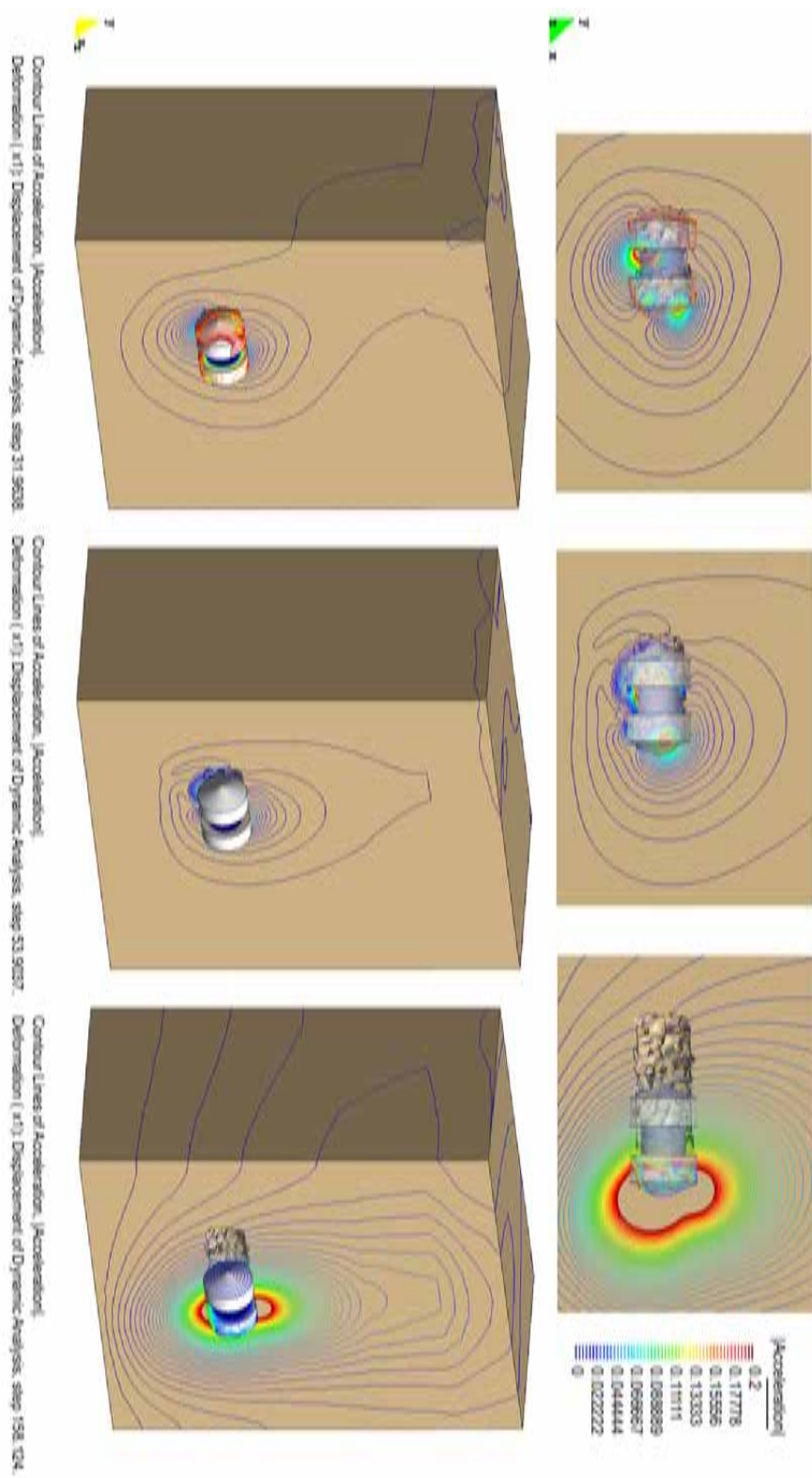
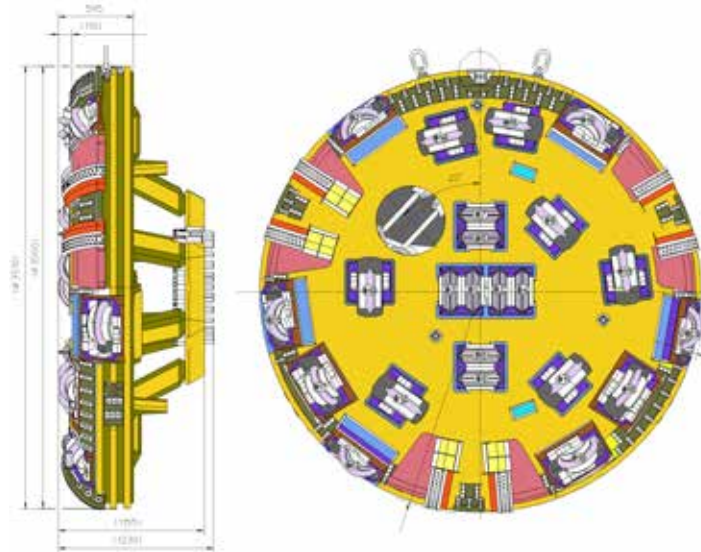


Figure 6.41: Contour lines of accelerations produced by the excavation forces.

#### 6.7.4 TBM excavation

The simulation of the actual functioning of a Tunnel Boring Machine (TBM) is performed in this example. Figure 6.42 shows the real geometry of the TMB head. It is a rock cutting head of small size. This machine is usually used when large water canalizations must go underground. That happens when the water piping crosses a highway or a populated area. This type of machines are used only for excavating hard rock. For soft rock grounds discs are not used and are generally replaced by picks.

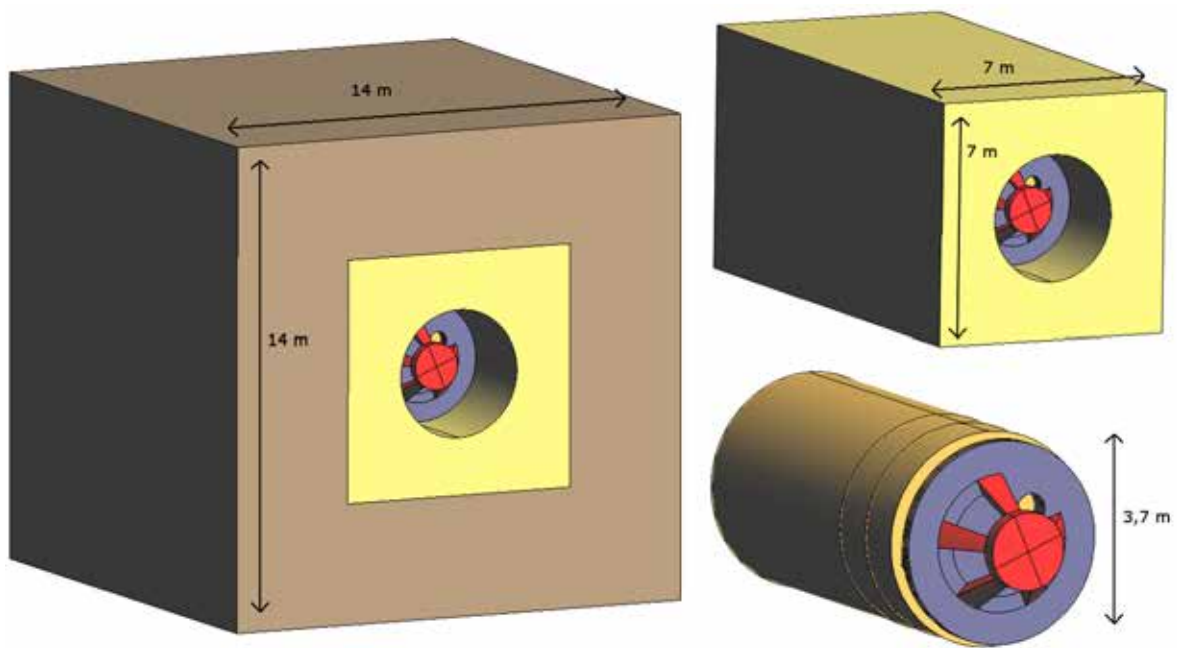


**Figure 6.42:** Geometry of a real rock cutting head, measures in mm (Herrenknecht).

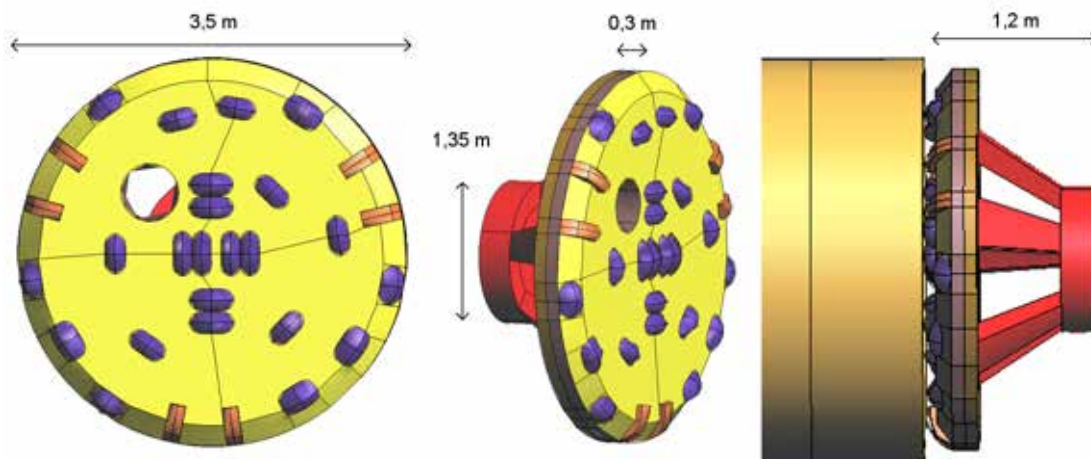
The geometry of a TBM is very complex. Therefore, a simplified model is created from the machine depicted in Figure 6.42. Figure 6.44 shows the TMB cutting head model and Figure 6.43 the piece of rock-solid where the machine is introduced. In Table 6.11 the material properties for the rock-solid and the TMB head are presented. To model the excavation, the characteristics of the ground and the boreability of the cutting parts have to be defined. These properties are assigned to the TBM head surface. Figure 6.45 shows the assignation area, with different colors, and the parameters used for each case.

The TBM kinematics are the head rotation and the forward movement. In order to reproduce them, an imposed velocity of  $\mathbf{v} = (0.01, 0, 0) \text{ m/s}$  and an imposed rotation of 4 rpm are assigned to the rear support of the TBM head. These movements are kept constant during the entire analysis. The rock-solid is fixed at the base.

The problem is analyzed using two different unstructured meshes. The meshes differ in the number of elements and nodes. In both cases the discretization is finer in the excavation zone. The finer mesh has initially 704475 tetrahedra and 127299 nodes; it is shown in Figure 6.46. The coarse mesh has initially 252928 tetrahedra and 48970 nodes. The time step used in both analysis is  $\Delta t = 0.01 \text{ s}$ .



**Figure 6.43:** Ground model in the TBM excavation problem.

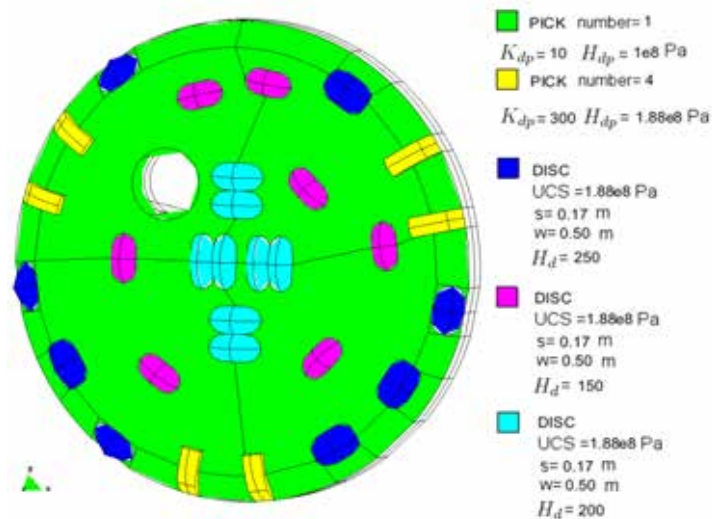


**Figure 6.44:** Model of the real rock cutting head of the TBM.

The initial simulation period is selected in order to study the computed forces and wear in detail. Figure 6.47 shows the total force computed on the TMB cutting head. The evolution is very irregular and tends to increase. From that results the needed power for the TBM can be estimated. Figures 6.48 and 6.49 show the instantaneous wear and the accumulated wear respectively. With these values we have a reference about the number

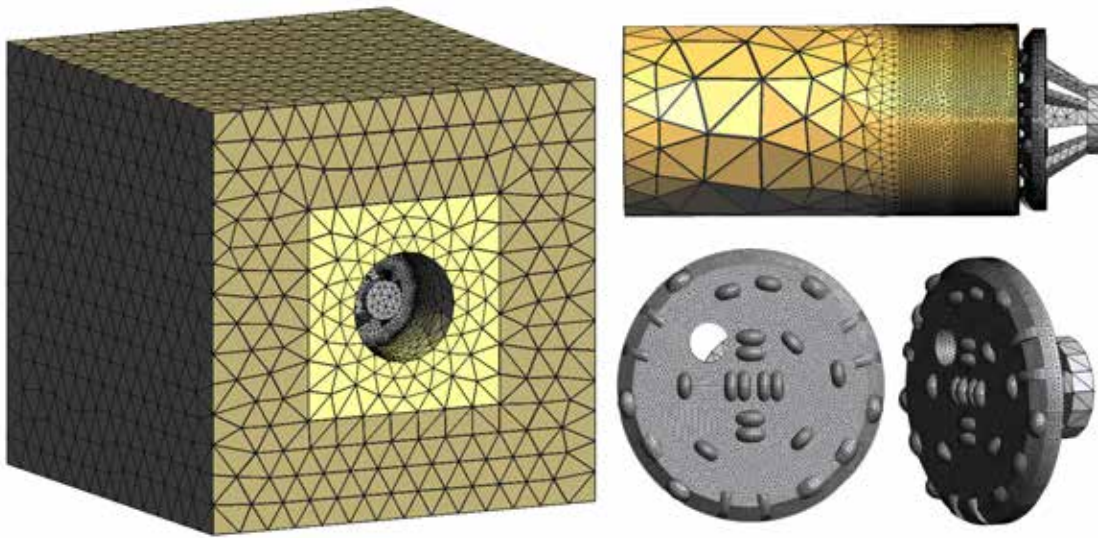
**Table 6.11:** Material properties for the TBM excavation example

Materials	Rock	TBM rear support	TBM Cutting Head
$E$ (Pa)	$4 \cdot 10^9$	$1,96 \cdot 10^{12}$	$1,96 \cdot 10^{11}$
$\nu$	0,15	0,33	0,33
$\delta$ ( $kg/m^3$ )	2500	7850	7850
$H$ (Pa)	$2 \cdot 10^9$	$10^{10}$	$10^{10}$
$K_w$	1	0,5	0,5
$\mu_d$	0,2	0,2	0,2
$\mu_s$	0,3	0,3	0,3

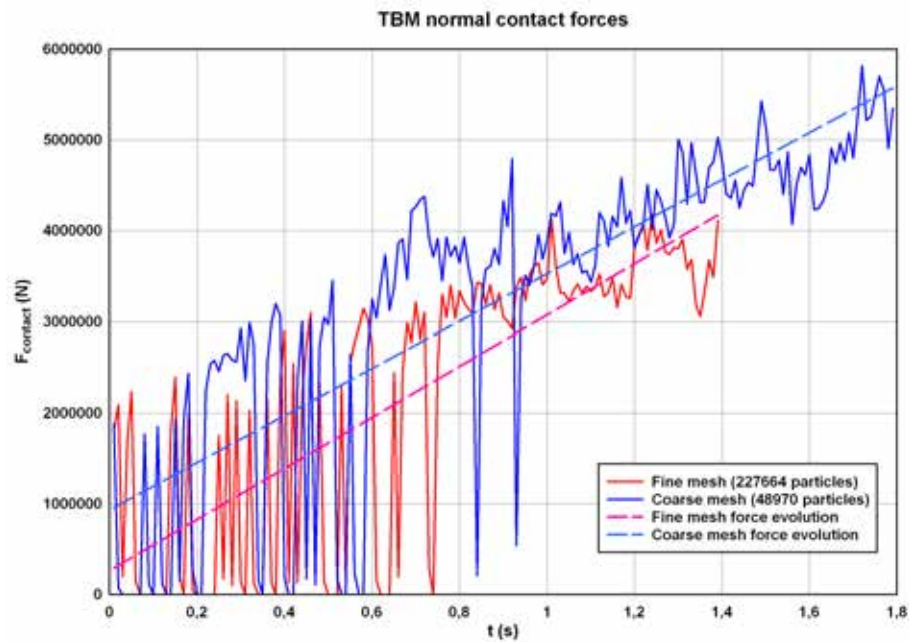
**Figure 6.45:** Excavation parameters for the TBM cutting head.

of times the discs must be changed. The most outworn tools can be determined using the wear distribution on the TMB head. This is shown in Figure 6.50 which depicts the wear produced on the TBM discs after some rotation and pressure over the ground surface.

Figures 6.51 and 6.52 display some results in the initial interaction between the TBM and the ground. The model reproduces the volume loss due to excavation. Some volume on the surface of the ground is taken away automatically when discs dig on it. The prior geometry is shaped to the new geometry at each time step. Figure 6.52 shows the instantaneous forces of the TBM discs against the massive ground. These forces are used



**Figure 6.46:** Discretization of the model. The initial mesh has 704475 tetrahedra and 127299 nodes.



**Figure 6.47:** Normal contact forces produced during the TBM excavation.

to estimate the needed power and torque. The results also provide a reference for the TMB excavation advance and prediction of the wear on the machine cutting tools (Figure 6.50).

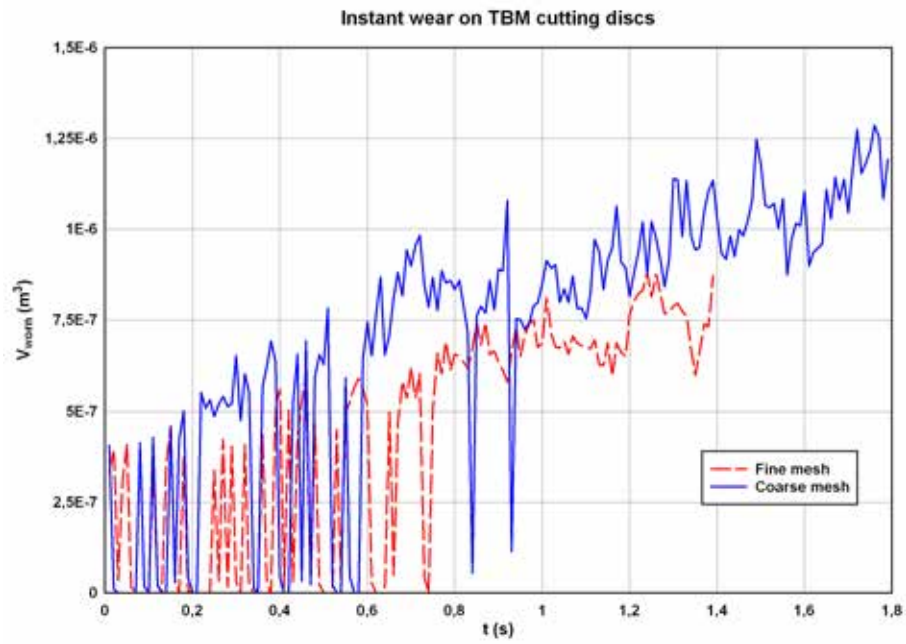


Figure 6.48: Instantaneous wear computed on the surface of the TBM cutting discs.

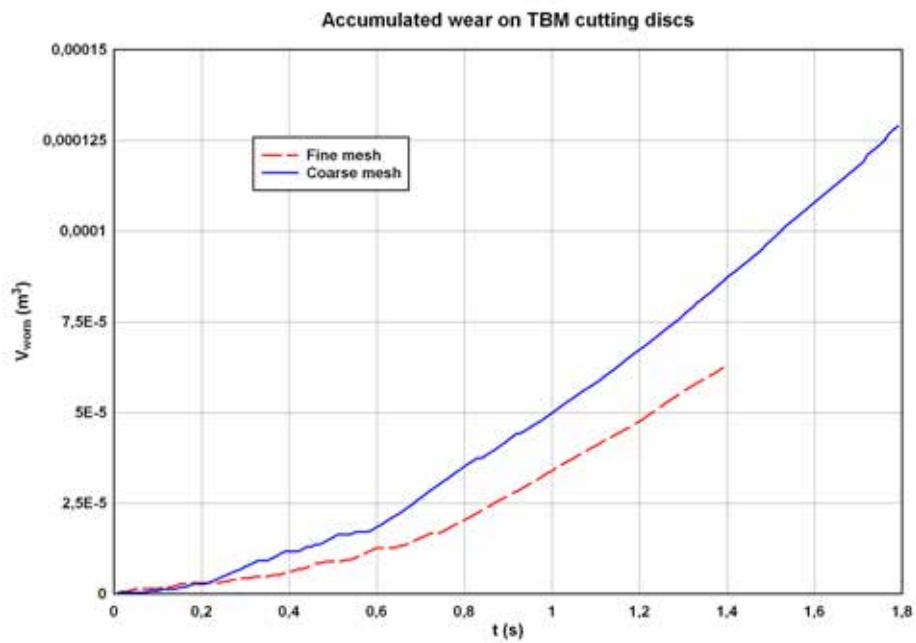


Figure 6.49: Accumulated wear computed on the surface of the TBM cutting discs.



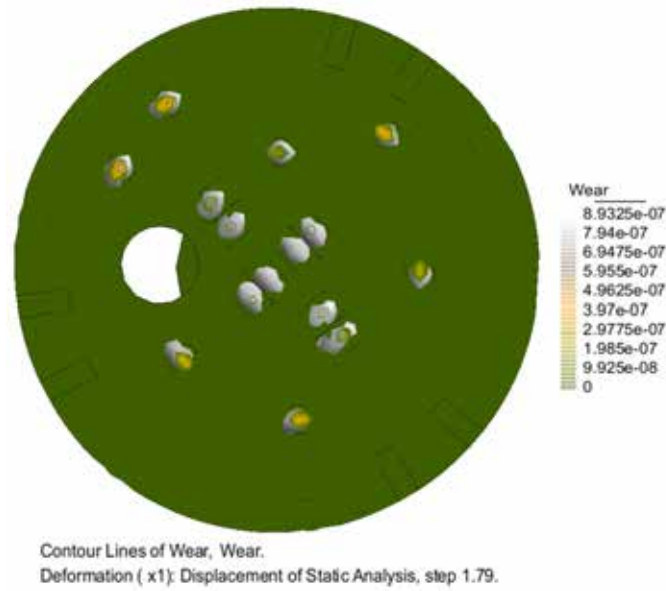


Figure 6.50: Volume loss due to wear on the TBM discs at time  $t = 1.79s$ .

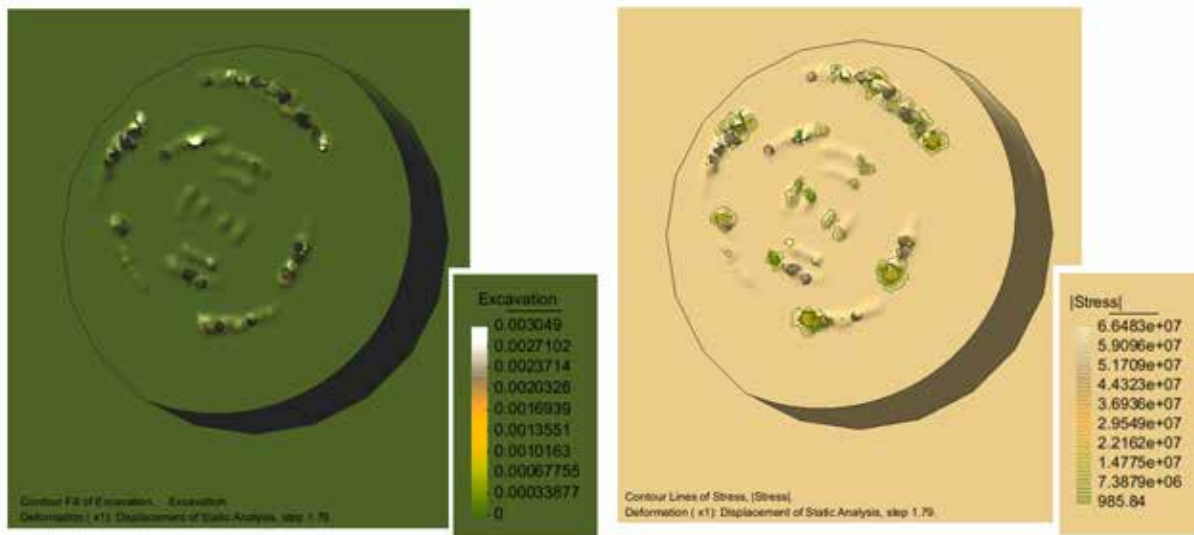
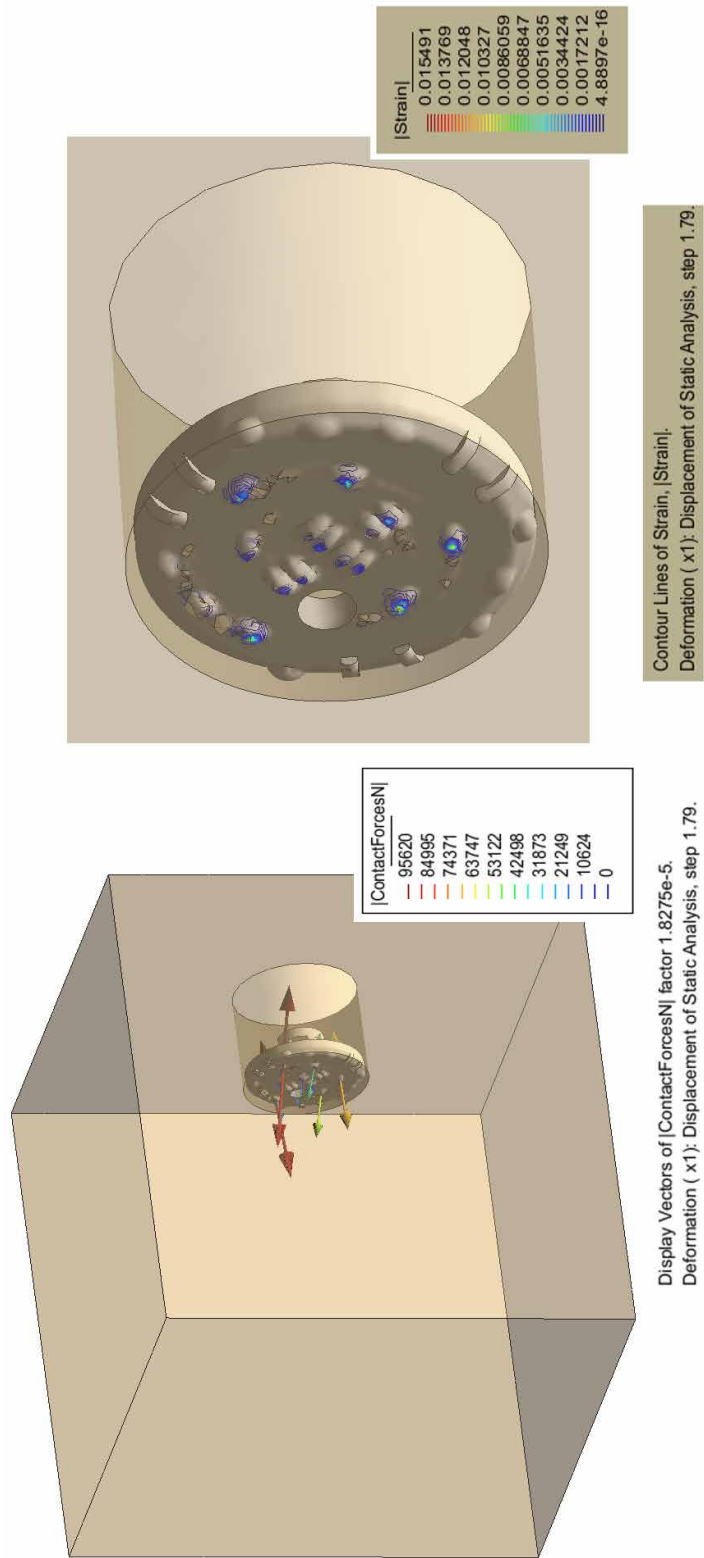


Figure 6.51: Excavation rate and stresses on the massive ground in the initial interaction.



**Figure 6.52:** Normal forces and strains produced due to the excavation in the initial interaction.



*Truth is what stands the test of experience.*

Albert Einstein (1879-1955)

# 7

## Conclusions

This monograph presents the advances in the numerical modeling of ground excavation using a new method, the PFEM, which is an alternative to the discrete element method (DEM) for the simulation of these complex problems.

The accomplished research can be seen as an incremental improvement of existing particle based methods. Innovative solutions are found to solve the new problems encountered in the PFEM formulation. Much has been done to extend the PFEM to non-linear solid mechanics problems. Furthermore, several new features are implemented extending the method capabilities. The overall objectives have been achieved. We have done our bit in the study of underground construction. The examples presented show that the PFEM has an excellent performance for the modeling of ground excavation processes.

### 7.1 Final Remarks

#### 1. The major contributions of the work are:

- The development of PFEM in the field of non linear solid mechanics and the adaptation for modeling ground excavation. New schemes for material definition are introduced inside a particle-based structure. An integral solution for transfer of variables and the adaptation of the constitutive modeling in the PFEM has been presented.
- The development of new techniques within the PFEM for meshing and boundary recognition. New algorithms for the geometry refinement and for the contact detection have been developed.
- The development of contact mechanics in the PFEM. A new treatment of the contact problem is developed in this work. This includes an innovative strategy for the contact detection and computation. The *Continuum Constraint Method* has been developed as a relevant feature of the PFEM.
- The classical material wear theory has been adapted to model numerically tool wear and excavation, via coupling the FEM with automatic geometry shaping.
- A global PFEM solution scheme is created by assembling all the single contributions presented in this work. They set a powerful tool for simulating a wide range of excavation processes, from a full tunneling machine to a single cutting tool.

- A new C++ code for the application of the PFEM in solid mechanics, contact mechanics and excavation has been programmed and tested with several examples.

## 2. The remarkable advantages of the PFEM are:

The method is based on the standard FEM, which permits the use of well-known constitutive equations and all the existing background knowledge in the method. This is an advantage compared with the DEM, which lacks this fundamental base in the constitutive modeling.

The PFEM structure allows the modeling of rapid changes in the domain geometries. The flexibility offered by the particle settings makes it very suitable for a wide range of problems.

Obtained results show that the PFEM is a competitive alternative versus DEM to model overall excavation problems. This is due to its greater accuracy and computational efficiency for solving large scale 3D problems of practical interest in civil engineering.

## 7.2 Further lines of research

Several lines of research open from the results of the present work. They can be grouped in three different parts:

### 1. Future research in the PFEM:

There is an important research on the PFEM in many fields. Future extensions of this work will focus in coupling fluid mechanics and solid mechanics in unified scheme for the PFEM. A general method for fluid structure interaction with deformable solids is one of the important future goals. For that, several aspects have to be improved, from the remeshing strategy to the solution method.

The use of a mixed formulation can be the starting point for the treatment of FSI problems combined with the present formulation. At present, there are some approaches of this type. The new objective is to use these approaches as a bridge between the fluid and solid formulations, but not focussing the solution as a single formulation.

The use of a mixed formulation for the treatment of incompressibility is also an objective from the point of view of contact mechanics. There are many aspects that can be adapted to the modeling of the contact constraints. The future research on the field must aim for more stable and accurate contact methods.

### 2. Future research in excavation:

Firstly, the developed models for the volume loss prediction due to excavation have to be properly calibrated. Some study cases must be taken as a reference to determine

the correct characterization parameters. This information is essential for future research work in the field.

Excavation problems are very complex and are coupled with several other phenomena which has not been treated yet. New solutions for the removing of material have to be investigated. The research must be extended to the coupling of excavation models with fracture models. The particular contact elements used in the PFEM can be improved and used for the discretization and modeling of the crushed material.

Other physical phenomena coming from excavation has to be included in the analysis. Heat and water are two important factors that have a lot of influence in an excavation. Further research must include the thermo-mechanical effects from the contact interaction. Once water can be modeled in the same analysis the mechanical effects and the refrigeration of water also have to be included.

The interface mesh used for contact modeling can be employed for the treatment of fluid mechanics into the excavation process.

### 3. Future work in programming:

The computational cost of the method can be improved with a better programming and extending the code to parallel computing. Several new features can be added into the programmed code. They consist in the application and adaptation of existent methodologies or schemes. Some ideas are proposed next:

- Program an explicit formulation to be used instead of the implicit solution scheme for the PFEM. A combined implicit-explicit solution can be programmed as well as other technics for the control of forces and displacements. Line-search and arch-length technics have to be added to the solution methods.
- Program a mixed formulation to enhance the future treatment of FSI problems and in order to develop a new contact method with incompressible contact elements.
- Include other element types to discretize solid domains and use higher interpolation order. The FEM can be combined easily with the PFEM in order to obtain solutions with higher accuracy.
- Apply new contact formulations for the contact treatment into the PFEM. The standard contact methods can be adapted to the PFEM characteristics.
- Include a progressive mesh refinement with an automatic adaptation to the problem. New criteria for the addition of particles and for the domain discretization have to be implemented in the code.
- Program new adaptative features on the imposed boundary conditions for the excavation problems. Dynamic boundary conditions have to reproduce the torque and power of the machine that have be controlled in function of the computed contact forces.



*Knowledge is power.*

Sir Francis Bacon (1561-1626)



## Circumradius

This appendix presents the form to find the circumcircle of a triangle and the circumsphere of a tetrahedron. In a finite element mesh these geometric figures are defined for the coordinates of its vertexes. The radius of the circumcircle or the circumsphere is the variable to compare with the  $\alpha$ -shapes in order to define the shape of the domains, see section 4.3.1. The following explanation can be found in [125] and [126].

### A.0.1 Circumcircle

The equation for the circumcircle of the triangle with polygon vertices  $(x_i, y_i)$  for  $i = 1, 2, 3$  is

$$\begin{vmatrix} x^2 + y^2 & x & y & 1 \\ x_1^2 + y_1^2 & x_1 & y_1 & 1 \\ x_2^2 + y_2^2 & x_2 & y_2 & 1 \\ x_3^2 + y_3^2 & x_3 & y_3 & 1 \end{vmatrix} = 0 \quad (\text{A.1})$$

Expanding the determinant yields

$$a(x^2 + y^2) + b_x x + b_y y + c = 0 \quad (\text{A.2})$$

where  $a$ ,  $b_x$ ,  $b_y$  and  $c$  are

$$a \equiv \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \quad (\text{A.3})$$

$b_x$  is the determinant obtained from the matrix

$$b = \begin{pmatrix} x_1^2 + y_1^2 & x_1 & y_1 & 1 \\ x_2^2 + y_2^2 & x_2 & y_2 & 1 \\ x_3^2 + y_3^2 & x_3 & y_3 & 1 \end{pmatrix} \quad (\text{A.4})$$

by discarding the  $x_i$  column and taking a minus sign. This is similar for  $b_y$  but this time taking the plus sign and discarding  $y_i$ . That gives:

$$b_x = - \begin{vmatrix} x_1^2 + y_1^2 & y_1 & 1 \\ x_2^2 + y_2^2 & y_2 & 1 \\ x_3^2 + y_3^2 & y_3 & 1 \end{vmatrix} \quad (\text{A.5})$$



and

$$b_y = + \begin{vmatrix} x_1^2 + y_1^2 & x_1 & 1 \\ x_2^2 + y_2^2 & x_2 & 1 \\ x_3^2 + y_3^2 & x_3 & 1 \end{vmatrix} \quad (\text{A.6})$$

Finally  $c$  is given by

$$c \equiv - \begin{vmatrix} x_1^2 + y_1^2 & x_1 & y_1 \\ x_2^2 + y_2^2 & x_2 & y_2 \\ x_3^2 + y_3^2 & x_3 & y_3 \end{vmatrix} \quad (\text{A.7})$$

Using last definitions in (A.2) and completing the square gives

$$a \left( x + \frac{b_x}{2a} \right)^2 + a \left( y + \frac{b_y}{2a} \right)^2 - \frac{b_x^2}{4a} - \frac{b_y^2}{4a} + c = 0 \quad (\text{A.8})$$

which is a circle with the form

$$(x - x_0)^2 + (y - y_0)^2 = r^2 \quad (\text{A.9})$$

with the circumcenter in

$$x_0 = -\frac{b_x}{2a} \quad (\text{A.10})$$

$$y_0 = -\frac{b_y}{2a} \quad (\text{A.11})$$

and the circumradius as

$$R = \frac{\sqrt{b_x^2 + b_y^2 - 4ac}}{2|a|} \quad (\text{A.12})$$

### A.0.2 Circumsphere

The circumsphere is the sphere circumscribed in a given solid and its radius is called the circumradius. By analogy with the equation of the circumcircle, the equation for the circumsphere of the tetrahedron with polygon vertices  $(x_i, y_i, z_i)$  for  $i = 1, 2, 3, 4$  is

$$\begin{vmatrix} x^2 + y^2 & x & y & z & 1 \\ x_1^2 + y_1^2 & x_1 & y_1 & z_1 & 1 \\ x_2^2 + y_2^2 & x_2 & y_2 & z_2 & 1 \\ x_3^2 + y_3^2 & x_3 & y_3 & z_3 & 1 \\ x_4^2 + y_4^2 & x_4 & y_4 & z_4 & 1 \end{vmatrix} = 0 \quad (\text{A.13})$$

Expanding the determinant yields

$$a(x^2 + y^2 + z^2) + (D_x x + D_y y + D_z z) + c = 0 \quad (\text{A.14})$$

where  $a$ ,  $b_x$ ,  $b_y$  and  $c$  are the following

$$a \equiv \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \\ x_4 & y_4 & 1 \end{vmatrix} \quad (\text{A.15})$$

$D_x$  is the determinant obtained from the matrix

$$D = \begin{pmatrix} x_1^2 + y_1^2 & x_1 & y_1 & z_1 & 1 \\ x_2^2 + y_2^2 & x_2 & y_2 & z_2 & 1 \\ x_3^2 + y_3^2 & x_3 & y_3 & z_3 & 1 \\ x_4^2 + y_4^2 & x_4 & y_4 & z_4 & 1 \end{pmatrix} \quad (\text{A.16})$$

by discarding the  $x_i$  column and taking a plus sign. This is similar for  $D_y$  but this time taking the minus sign and discarding  $y_i$  and also for  $D_z$  discarding  $y_i$  and taking a plus sign. This gives

$$D_x = + \begin{vmatrix} x_1^2 + y_1^2 & y_1 & z_1 & 1 \\ x_2^2 + y_2^2 & y_2 & z_2 & 1 \\ x_3^2 + y_3^2 & y_3 & z_3 & 1 \\ x_4^2 + y_4^2 & y_4 & z_4 & 1 \end{vmatrix} \quad (\text{A.17})$$

$$D_y = - \begin{vmatrix} x_1^2 + y_1^2 & x_1 & z_1 & 1 \\ x_2^2 + y_2^2 & x_2 & z_2 & 1 \\ x_3^2 + y_3^2 & x_3 & z_3 & 1 \\ x_4^2 + y_4^2 & x_4 & z_4 & 1 \end{vmatrix} \quad (\text{A.18})$$

$$D_z = + \begin{vmatrix} x_1^2 + y_1^2 & x_1 & y_1 & 1 \\ x_2^2 + y_2^2 & x_2 & y_2 & 1 \\ x_3^2 + y_3^2 & x_3 & y_3 & 1 \\ x_4^2 + y_4^2 & x_4 & y_4 & 1 \end{vmatrix} \quad (\text{A.19})$$

Finally  $c$  is given by

$$c \equiv - \begin{vmatrix} x_1^2 + y_1^2 & x_1 & y_1 & z_1 \\ x_2^2 + y_2^2 & x_2 & y_2 & z_2 \\ x_3^2 + y_3^2 & x_3 & y_3 & z_3 \\ x_4^2 + y_4^2 & x_4 & y_4 & z_4 \end{vmatrix} \quad (\text{A.20})$$

Using last definitions in (A.14) and completing the square gives

$$a \left( x + \frac{D_x}{2a} \right)^2 + a \left( y + \frac{D_y}{2a} \right)^2 + a \left( z + \frac{D_z}{2a} \right)^2 - \frac{D_x^2 + D_y^2 + D_z^2}{4a} + c = 0 \quad (\text{A.21})$$

which is a circle with the form

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2 \quad (\text{A.22})$$

with the circumcenter in

$$x_0 = \frac{D_x}{2a} \quad (\text{A.23})$$

$$y_0 = \frac{D_y}{2a} \quad (\text{A.24})$$

$$z_0 = \frac{D_z}{2a} \quad (\text{A.25})$$

and the circumradius as

$$R = \frac{\sqrt{D_x^2 + D_y^2 + D_z^2 - 4ac}}{2|a|} \quad (\text{A.26})$$

*In a few minutes a computer  
can make a mistake so great  
that it would have taken  
many men many months to equal it.*

Unknown

# B

## Developed Code

Several developments in the PFEM are presented in this work. These developments are based in particle description of the continuum and extended to the field of non-linear solid mechanics. The new approach for the continuum medium description and operation needs particular techniques and algorithms for the programming of the method. Standard code structures used in the classical FEM have to be readapted to include a particle based arrangement.

At the beginning of this work no code existed yet for the study of solid mechanics problems with the PFEM. First advances were focussed in the field of Computational Fluid Dynamics (CFD). When this work started there was only a single PFEM code, programmed in C++ language, applicable to CFD problems. Nowadays there are several extensions of that code and even new codes that use the PFEM for fluid dynamic calculations.

The programming done in this work started almost from zero, with some small references from fluid codes and fully thinking on a particle-based method structure. It is not the adaptation of other finite element code. However, some general libraries for the mesh generation and for solving linear systems of equations have been used.

It is important to remark that the programming is a task that requires a lot of time. This increases if there is no previous background and everything has to be build from the beginning. That has also other consequences, it is difficult to find the best alternative in the first attempt to solve the problem.

### B.1 Description of the code

In this work the PFEM has been programmed using C++. This programming language is object-oriented. Each entity of the code has their own group of variables and functions. These objects have also particular relationships with other objects and can be organized following a specific hierarchy. Some references can be found in [43], [46], [102] and [35].

The object-oriented language is very appropriate for defining a clear structure of the code. It is also very suitable for an easy update of the program features. The objects are defined by class types or by class templates. Using properly templates for the class definition a single code for 2D and 3D is programmed for the PFEM.

At this time the program runs in Linux. It is a sequential program and it has to be compiled for each computer. It has to be mentioned that it is an original work but includes

some important external libraries:

1. Standard C++ class library.
2. Boost library. It uses *ublas* classes for containers: matrices and vectors [1].
3. PETSC: Portable, Extensible Toolkit for Scientific Computation [3].
4. TetGen: tetrahedron mesh generator [4].
5. TRIANGLE: triangle mesh generator [5].
6. GIDPost: library for the post-processing in GID [2].

The features of the two first libraries are used for building the variables of the code. PETSC is used as a library for linear solvers. TetGen and Triangle are the meshing libraries included in the code. The particles or points in the space are passed to that classes and a mesh is returned. Finally the GIDPost library is used to write the results according to GID standards in order to be visualized with the GID postprocess.

PETSC must be compiled with a unix environment, this is the reason to use a Linux platform only. The used libraries can be replaced for similar ones if is necessary for further applications.

## B.2 General scheme of the program

In order to offer a general point of view of the structure of the code, the program scheme is shown in Figure B.1. Two strategies are present, the 2D strategy with the linear triangle as a finite element and the 3D strategy, with a linear tetrahedron as a finite element. *Tetrahedra* and *Triangles* are derived classes from a general *Element* class. New derived elements can be added in the future. Within these strategies the mesher is automatically selected depending on the space dimension and the element type.

The kernel of the program is shared for all strategies. It starts for the input file reading. The reading functions are located in classes *Point* and *Node*, in class *Element* and in the class *Domains*. Class *Domains* establish a bridge between conditions of materials assigned to elements that have to be transferred to points. Once the initial conditions and boundary conditions are kept in the random access memory. The class *Assembly* manages the assignation to the global variables. The functions located in the object *Assembly* are called from the general main class *Solve*. Within *Solve* the solution scheme is located, the functions for the variable transfer are called, the system of equations is build and the implicit (or explicit) solution scheme is performed. Within this solution flowchart the mesher is also called.

The initial chosen strategy defines which mesher is used. The mesher manage the meshing, the geometry shaping due to excavation and wear, the alpha-shapes and the variables update (Section 4.3.1). An output file is written with the results for every time step. The program keeps on running the solution loop until the end of the computation.

### Program Structure

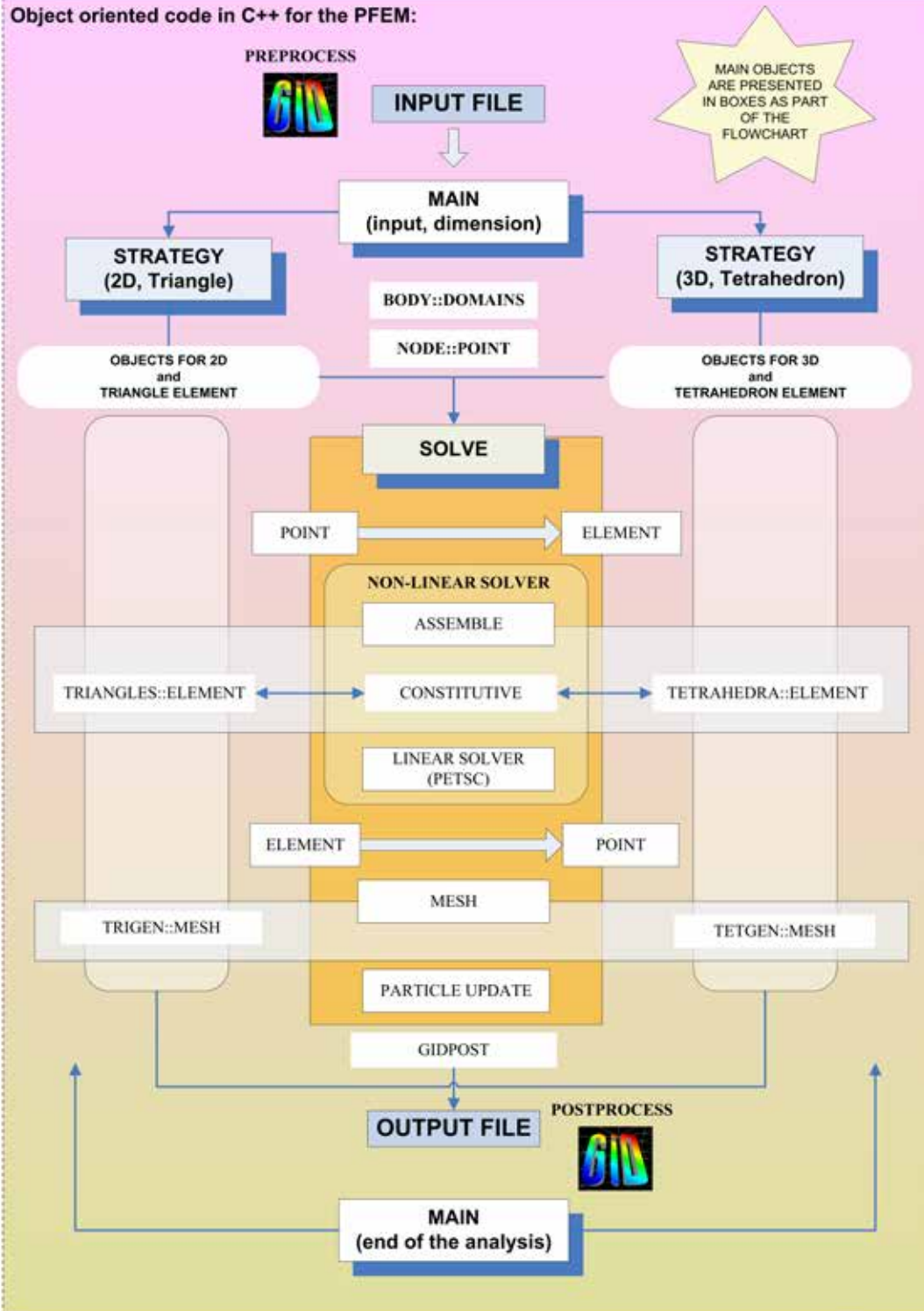


Figure B.1: Program scheme for the PFEM

This is a general explanation, there are many details in every class and every function of the code. This explanation is just a general introduction to the programming of the PFEM carried out in this work. For a complete understanding one must mindfully study the code written, which is not included here. However, a short description of the main classes is given next in order to have a closer approach on their functioning.

### B.3 Description of the main classes

The main classes of the code are listed and explained briefly in the following:

**Point** contains the main geometrical functions for point arrays and single point variables. It has the enumeration functions for the assembly of the global variables.

**Node::Point** is a class derived from *Point*, it contains the reading functions for the input file, with the arrays for the boundary conditions and the arrays for the initial conditions. The main variables for storing particle information are defined in this class.

**Element** is the base class for a single element as well as for the element arrays. It contains the functions for single element computation and the global functions for the information transfer from points to elements. It also assigns the flags to element domain and element material and their properties.

**Triangles::Element** is a class derived from *Element*. It computes the elemental triangle variables. It also contains the triangular contact element and the search of the contact active set.

**Tetrahedra::Element** is a class derived from *Element*. It computes the elemental tetrahedron variables. It also contains the tetrahedral contact element and the search of the contact active set.

**Assembly** assembles the elemental matrices from *Triangles* or *Tetrahedra* and returns the global variables for the system of equations. It contains the functions for the total to global variables transfer. It also contains the functions for boundary and initial conditions imposition. The update of the point variables from elemental values is performed by functions of this class.

**Mesh** assigns geometrical order to *Element* variables and *Point* variables. It contains functions for the boundary identification and mesh distribution: points surrounding points, elements surrounding points, elements surrounding elements, boundary surface identification, normals to the boundary surface. Functions for the geometry shrinkage of the body domains are also defined here. The meshes from the whole domain, contact and body meshes are treated here as a single mesh.

**Trigen::Mesh** is a class derived from *Mesh*. A selection of the particles is performed and the spatial position modified for the excavation and wear schemes. After that, a new mesh of triangles is generated from a cloud of points. With the

alpha-shape concept, the boundaries are defined. The point values are updated to the new size and number. All these processes are performed by the functions of this class.

**Tetgen::Mesh** is a class derived from *Mesh*, a selection of the particles is performed and the spatial position modified for the excavation and wear schemes. After that, a new mesh of tetrahedra is generated from a cloud of points. With the alpha-shape concept, the boundaries are defined. The point values are updated to the new size and number.

**Constitutive** contains the constitutive laws for the finite elements. They are linear elasticity, non-linear elasticity, damage models and plasticity. It is a class that is called by *Triangles* or *Tetrahedra* during the computation of the elemental variables.

**Petscsol** contains the solvers and pre-conditioners for the linear system of equations.

**Domains** has the containers for the material definition for the particles and the elements of the domain. It has the functions to read the material assignment and the material transfer scheme.

**GidPost** the nodal and elemental variables are written to a data file to be read by the GID post-processor.

**Solve** contains the general solution scheme. It has the functions for implicit and explicit solvers. It asks to the *Assembly* class all global variables for the system building. It calls the linear solver, in the *Petscsol* class, for each iteration and the mesher, in the *Mesh* class, for each time step. It also calls the *GidPost* class in order to write the step results. It is the main engine of the code.

**Strategy** is the class that sets the problem characteristics, dimension, type of element and mesh. It also sets the particular preferences for an implicit or explicit integration of the *Solve* class.

## B.4 Computational cost and time consuming

Some references to the computational cost and computational time have to be mentioned. This only pretends to give an order of magnitude of the speed and the cost of the programmed code for the PFEM. It is very difficult to compare computational information with other codes if there is not a complete knowledge of their internal features. Time and memory consuming is a sum of many aspects. There include the most basic variable definition as well as the largest and more complex algorithms coded. In addition, the compiler and the hardware can change radically the speed of the same computation.

However, an objective analysis of the obtained data can be done. The computational times for some examples are presented in Table B.1. 3D Problems with 1 million elements need around 4 Gb of RAM, with 700 thousand elements around 3 Gb of RAM and with 400 elements around 1 GB of RAM. Taking into account that they are linear elements,



triangles(2D) or tetrahedra(3D), and contact elements are also present in the analysis. The building of the system is more complex as it comes from the particle basis and has the contact features included.

With this data everyone can make a comparison with the capabilities of other codes. It gives an idea about how fast and computationally cheap the PFEM could be with an optimal programming. Improvements can be achieved using a different structure for the code, using parallel computing or even more a different programming language.

**Table B.1:** Time consuming with the programmed code for the PFEM (2,4 GHz processor)

Problem	Elements	Nodes	$\Delta t$ (s)	Average iterations	Building time (s)	Solver time (s)	Mesh&Update time (s)	Total Time for $\Delta t$ (s)
2D Elastic Contact	1986	1271	$5 \cdot 10^{-6}$	3	0.03	0.07	0.03	0.6
2D Roadheader	15488	7919	0.005	3	0.3	0.2	0.18	3.18
3D Contact Wear	28988	6155	0.001	1	2	1.4	1.3	10
3D Linear test (LT)	243838	51109	0.001	3	10	20-30	9	150
3D Damage LT	243651	51092	0.001	2-5	12	30-60	9	250
3D Linear test 2	560323	121043	0.001	7	28	70	25	1000
3D RoadHeader 1	427450	76792	0.01	2-6	27	15-30	15	200
3D RoadHeader 2	1317240	227664	0.02	3	70	80	42	800
3D Static TBM 1	255425	48970	0.01	3	13	7	6	110
3D Static TBM 2	706876	127299	0.01	3	43	47	21	410
3D Dynamic TBM 1	255425	48970	0.01	3-7	13	8	7	150
3D Dynamic TBM 2	706876	127299	0.01	6	41	43	21	500
3D Dynamic TBM 3	1065945	185191	0.005	2-4	67	100	48	1200

## References

- [1] BOOST C++ libraries. <http://www.boost.org/>.
- [2] GID. <http://gid.cimne.upc.es/>.
- [3] *PETSC*. <http://www.mcs.anl.gov/petsc/petsc-as/>.
- [4] Tetrahedral mesh generator. <http://tetgen.berlios.de>.
- [5] Triangle mesh generator. <http://www.triangle.com>.
- [6] G. Amontons. On the resistance originating in machines (in french). *Mem. Acad. Roy.*, pages 206–222, 1699.
- [7] J.F. Archard. Contact and rubbing of flat surfaces. *Journal of Applied Physics*, 24:981–988, 1953.
- [8] M. Arrea and A. R. Ingraffea. Mixed-mode crack propagation in mortar and concrete. Report 81-13, Department of Structural Engineering, Cornell University, Ithaca, New York, 1982.
- [9] A. Barbat and J. Miquel. *Estructuras sometidas a acciones sismicas*. CIMNE, 2 edition, 1994.
- [10] C.B. Barber, D.P. Dobkin, and H.T. Huhdanpaa. The Quickhull algorithm for convex hulls. *ACM Trans. on Mathematical Software*, 44(4):469–483, December 1934. <http://www.qhull.org>.
- [11] K. J. Bathe. Conserving Energy and Momentum in Nonlinear Dynamics: A simple implicit time integration scheme. *Computers and structures, in press*, 2007.
- [12] K.J. Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice Hall, 1982.
- [13] T. Belytschko, W.K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2000.
- [14] D.P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, New York, 1984.
- [15] D. Besdo and E. Stein, editors. *On the treatment of contact constraints within coupled thermomechanical analysis*, Finite Inelastic Deformations, Berlin, 1992. Proc. of EUROMECH, Springer.
- [16] J. Bonet and R. D. Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press, 1997.
- [17] B. Boroomand and O. Zienkiewicz. Recovery by Equilibrium in patches (REP). *Int. J. Num. Meths. Engrg.*, 40(1):137–164, 1997.

- [18] R.M. Brannon. *Functional and structured tensor analysis for engineers*. University of New Mexico, Albuquerque, 2003.
- [19] J.T. Burwell. Survey of possible wear mechanisms. *Wear*, 1:119–141, 1957.
- [20] J.T. Burwell and E. Rabinowicz. The nature of the coefficient of friction. *J. Appl. Phys.*, 24:136–139, 1953.
- [21] N. Calvo, S.R. Idelsohn, and E. Oñate. The extended Delaunay tessellation. *Engineering Computations*, 20:583–600, 2003.
- [22] E. J. Car. *Modelo constitutivo continuo para el estudio del comportamiento mecánico de los materiales compuestos*. PhD thesis, Departament de Resistència de Materials i Estructures a l'Enginyeria (RMEE) UPC, March 2000.
- [23] S.W. Cheng, T.K. Dey, H. Edelsbrunner, M.A. Facello, and S.H. Teng. Sliver Exudation. *Proc. 15th Ann. Sympos. Comput. Geom.*, 1999.
- [24] S.W. Cheng, T.K. Dey, and T. Ray. Weighted Delaunay Refinement for Polyhedra with Small Angles. *Proceedings 14th International Meshing Roundtable*, September 2005.
- [25] B. Chiara. Fracture mechanisms induced in a brittle material by a hard cutting indenter. *International Journal of Solids and Structures*, 38:7747–7768, 2001.
- [26] J. Chung and G.M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized alpha-method. *Journal of Applied Mechanics*, 60:371–375, 1993.
- [27] K. L. Clarkson, K. Mehlhorn, and R. Seidel. Four Results on Randomized Incremental Constructions. *In Symposium on Theoretical Aspects of Computer Science*, 1992.
- [28] T.F. Conry and A. Seireg. A mathematical programming method for design of elastic bodies in contact. *Journal of Applied Mechanics*, 38:1293–1307, 1971.
- [29] L. Cornejo. Las Máquinas Rozadoras en Túneles y Minas. *Revista de Obras Públicas*, 1:177–191, 1985.
- [30] L. Cornejo. La excavación mecánica de rocas. *Revista de Obras Públicas*, 1:649–664, 1987.
- [31] C.A. Coulomb. The theory of simple machines (in french). *Mem. Math. Phys. Acad. Sci.*, 10:161–331, 1785.
- [32] R. Courant, K.O. Friedrichs, and H. Lewy. Über die partiellen differenzgleichungen der mathematischen physik. *Math. Ann.*, 100:32, 1928.

- [33] P.A. Cundall and O.D.L. Strack. A discrete numerical method for granular assemblies. *Geotechnique*, 29:47–65, 1979.
- [34] A. Curnier and P. Alart. A generalized newton method for contact problems with friction. *J. Mec. Theor. Appl.*, 7:67–82, 1988.
- [35] Pooyan Dadvand. *A framework for developing finite element codes for multi-disciplinary applications*. PhD thesis, Universitat Politècnica de Catalunya (UPC), 2007.
- [36] F. Del Pin. *The Meshless Finite Element Method Applied to a Lagrangian Particle Formulation of Fluid Flows*. PhD thesis, Facultad de Ingeniería y Ciencias Hídricas (FICH), November 2003.
- [37] B.N. Delaunay. Sur la Sphère Vide, A la memoire de Georges Voronoi. *Otdelenie Matematicheskii i Estestvennyka Nauk*, 7:793–800, 1934.
- [38] Department of the Army U.S. Army Corps of Engineers, Washington, DC 20314-1000. *Engineering and Design TUNNELS AND SHAFTS IN ROCK*, 1997.
- [39] S.J. Dokos. Sliding friction under extreme pressures. *J. Appl. Mechanics*, 13A:148–156, 1946.
- [40] G. Duvaut and J.L. Lions. *Inequalities in Mechanics and Physics*. Springer, 1976.
- [41] H. Edelsbrunner and E.P. Mucke. Three dimensional alpha shapes. *ACM Transaction on Graphics*, 13:43–72, 1994.
- [42] I. W. Farmer and N. H. Glossop. Mechanics of Disc Cutter Penetration. *Tunnels and Tunneling*, 12(6):22–25, 1980.
- [43] J. Ferguson, B. Patterson, J. Beres, P. Boutquin, and M. Gupta. *La biblia de C#*. ANAYA, 2003.
- [44] K.A. Fischer. *Mortar Type Methods Applied to Nonlinear Contact Mechanics*. PhD thesis, Institute für Baumechanik un Numerische Mechanik, 2005.
- [45] Michel Geradin. General methods for flexible multibody systems. *2nd International Congress: 'Apport du calcul scientifique en automobile'*, February 1993.
- [46] L. Gil and M. Sánchez. *El C++ por la pràctica. Introducció al language y su filosofía*. Edicions Universitat Politécnica de Catalunya, S.L., 1999.
- [47] W. Goldsmith. *IMPACT: The Theory and Physical Behavior of Colliding Solids*. Dover, 1960.
- [48] P. C. Graham. Rock Exploration for Machine Manufacturers in Exploration for Rock Engineering. *Proceedings of the Symposium, Johannesburg, Z. T. Bieniawski, cd., Balkema*, 1:173–180, 1976.

- [49] C. Hahn. *Models, Algorithms and Software Concepts for Contact Fragmentation in Computational Solid Mechanics*. PhD thesis, Institute für Baumechanik un Numerische Mechanik, 2006.
- [50] E. Hinton and J.S. Campbell. Local and global smoothing of discontinuous finite element functions using a least squares method. *Int. J. Num. Meth. Engng.*, 8:461480, 1974.
- [51] J.D. Hoffman. *Numerical methods for engineers and scientists*. Marcel Dekker, Inc., 2001.
- [52] R. Holm. *Electric Contacts*. Almquist and Wiksells, Stockholm, 1946.
- [53] M. Hooda and M. Alehosseinb. A development in rock cutting technology. *International Journal of Rock Mechanics and Mining Sciences*, 37:297–305, 2000.
- [54] H. M. Hughes. The Relative Cuttability of Coal Measures Rock. *Mining Science and Technology*, 3:95–109, 1986.
- [55] S. Idelsohn, N. Calvo, and E. Oñate. Polyhedrization of an arbitrary 3d point set. *Computer Method in Applied Mechanics and Engineering*, 192:2649–2667, 2003.
- [56] S.R. Idelsohn, E. Oñate, N. Calvo, and F. Del Pin. The meshless Finite element method. *International Journal for Numerical Methods in Engineering*, 58(4), 2003.
- [57] S.R. Idelsohn, E. Oñate, and F. Del Pin. The particle finite element method a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *Int. J. Num. Meth. Engng.*, 61:267–307, 2004.
- [58] U. Israel. Implementation of an algorithm for data transfer on the fluid-structure interface between non-matching meshes. Master’s thesis, Technische Universität München, 2006.
- [59] B. Joe. Construction of Three-dimensional Delaunay Triangulations from local transformations. *Computer Aided Geometric Design*, 8:123–142, 1991.
- [60] K.L. Johnson. *Contact Mechanics*. Cambridge University Press, 1985.
- [61] L.M. Kachanov. Time of rupture process under creep conditions. *Otd Tech Nauk*, 8:26–31, 1958.
- [62] A.R. Khoei and S.A. Gharehbaghi. The superconvergence patch recovery technique and data transfer operators in 3D plasticity problems. *Finite Elements in Analysis and Design*, 43:630–648, 2007.
- [63] N. Kikuchi and J.T. Oden. *Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods*. SIAM, 1988.

- [64] A. Klarbring. A mathematical programming approach to three-dimensional contact problems with friction. *Computer Methods in Applied Mechanics and Engineering*, 58:175–200, 1986.
- [65] R. Klaus. Computation of Lie Derivatives of Tensor Fields Required for Nonlinear Controller and Observer Design Employing Automatic Differentiation. *Proc. Appl. Math. Mech.*, 5:181–184, 2005.
- [66] J. Kolář, Michor. P W. & Slovák. *Natural operations in differential geometry*. Springer-Verlag, 1993.
- [67] D. Kuhl and M.A. Crisfield. Energy-Conserving and Decaying algorithms in nonlinear structural dynamics. *Int. J. Num. Meth. Engng.*, 45:569–599, 1999.
- [68] D. Kuhl and E. Ramm. Constraint energy momentum algorithm and its application to nonlinear dynamics of shells. *Computer Methods in Applied Mechanics and Engineering*, 136:293–315, 1996.
- [69] C. Labra, J. Rojek, E. Oñate, and F. Zárate. Advances in discrete element modelling of underground excavations. *Acta Geotechnica*, 3:317–322, 2008.
- [70] C. Labra, J. Rojek, E. Oñate, and F. Zárate. Advances in discrete element modelling of underground excavations. *Acta Geotechnica*, 3:317–322, 2008.
- [71] T.A. Laursen and J.C. Simo. A continuum-based finite element formulation for the implicit solution of multibody, large deformation frictional contact problems. *International Journal for Numerical Methods in Engineering*, 36:3451–3485, 1993.
- [72] C.L. Lawson. Software for C1 surface interpolation. In *Mathematical Software III*, pages 161–194, 1977.
- [73] J.A. Loera, J. Rambau, and F. Santos. Triangulations of point sets: Applications, Structures, Algorithms. Textbook manuscript.
- [74] R. Lohner. *Applied CFD Techniques: An introduction based on finite element methods*. Wiley, first edition, 2001.
- [75] D.G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, 1984.
- [76] L.E. Malvern. *Introduction to the mechanics of a continuous medium*. Prentice-Hall, Inc., 1969.
- [77] J.M. Marti. *El Método de Elementos Finitos de Partículas en Interacción Fluido-Estructura*. PhD thesis, Universidad Nacional del Litoral, 2008.
- [78] X. Martínez. *Micro-Mechanical Simulation of Composite Materials Using the Serial/Parallel Mixing Theory*. PhD thesis, Departament de Resistència de Materials i Estructures a l'Enginyeria (RMEE) -UPC, June 2008.

- [79] A. Morin. New friction experiments carried out at Metz in 1831-1833 (in french). *Mem. Acad. Sci.*, 4:1–128, 591–696, 1833.
- [80] NTH. Hard rock tunnel boring. Project Report 183:1-88, Norwegian Institute of Technology, Trondheim, Norway, 1988.
- [81] E. Oñate. *Cálculo de Estructuras por el Método de los Elementos Finitos*, volume 2. CIMNE, 5 edition, 2000. in Spanish.
- [82] E. Oñate. Desarrollos y aplicaciones de modelos de fractura en la Escuela de Ingenieros de Caminos de Barcelona. *Anales de Mecánica de Fractura*, 17:127–154, 2000.
- [83] E. Oñate, M.A. Celigueta, and S.R. Idelsohn. Modelling bed erosion in free surface flows by the particle finite element method. *Acta Geotecnica*, 1(4):237–252, 2006.
- [84] E. Oñate, S.R. Idelsohn, and M.A. Celigueta. Lagrangian formulation for fluid-structure interaction problems using the particle finite element method. *Computer Methods in Applied Mechanics and Engineering*, 2006. Barcelona.
- [85] E. Oñate, S.R. Idelsohn, M.A. Celigueta, and R. Rossi. Advances in the particle finite element method for the analysis of fluid-multibody interaction and bed erosion in free surface flows. *Computer Methods in Applied Mechanics and Engineering*, 197(19-20):1777–1800, 2008.
- [86] E. Oñate, S.R. Idelsohn, F. Del Pin, and F. Aubry. The particle finite element method. an overview. *Int. J. Num. Meth. Engng.*, 1(2):964–989, 2004.
- [87] E. Oñate and J. Rojek. Combination of discrete element and finite element methods for dynamic analysis of geomechanics problems. *Comput. Methods Appl. Mech. Engng.*, 193:3087–3128, 2004.
- [88] E. Oñate, J. Rojek, R.L. Taylor, and O.C. Zienkiewicz. Finite calculus formulation for incompressible solids using linear triangles and tetrahedra. *International Journal for Numerical Methods in Engineering*, 59:1473–1500, 2004.
- [89] E. Oñate, H. Tschöpe, and P. Wriggers. Combination of the critical displacement method with a damage model for structural instability analysis. *Engineering Computations*, 18:642–662, 2001.
- [90] J.T. Oden and J.A.C. Martins. Models and computational methods for dynamic friction phenomena. *Computer Methods in Applied Mechanics and Engineering*, 52:527634, 1986.
- [91] J.T. Oden and E.B. Pires. Nonlocal and nonlinear friction laws and variational principles for contact problems in elasticity. *Journal of Applied Mechanics*, 50:67–76, 1983.

- [92] X. Oliver. A Consistent Characteristic Length for Smeared Cracking Models. *International Journal for Numerical Methods in Engineering*, 218:461–474, 1989.
- [93] X. Oliver and C. Agelet. *Mecánica de medios continuos para ingenieros*. Edicions UPC, 2000.
- [94] X. Oliver, J.C. Cante, R. Weyler, C. González, and J. Hernández. Particle finite element methods in solid mechanics problems. In Eugenio O nate and Roger Owen, editors, *Computational Plasticity*, pages 87–103. Springer, Netherlands, 2007.
- [95] X. Oliver, M. Cervera, S. Oller, and J. Lubliner. Isotropic damage models and smeared crack analysis of concrete. *Second international conference on Computer Aided Analysis and Design of Concrete Structures*, 1990.
- [96] S. Oller. *Un modelo de Daño continuo para materiales friccionales*. PhD thesis, Universitat Politècnica. Escola d’Enginyers de Camins, Canals i Ports, 1988.
- [97] S. Oller. *Dinámica no lineal*. CIMNE, 2001.
- [98] S. Oller. *Fractura Mecánica. Un enfoque global*. Edicions UPC, CIMNE, 2001.
- [99] L. Ozdemir. CSM computer model for TBM performance prediction. Technical report, Earth Mechanics Institute (Colorado School of Mines), 1993.
- [100] L. Ozdemir and F. Wang. Mechanical Tunnel Boring, Prediction, and Machine Design. Final Report NSF/RA-790161: 204, Colorado School of Mines, 1979.
- [101] P. E. Petersson. Crack growth and development of fracture zones in plain concrete and similar materials. Report TVBM-1006, Division of Building Materials, University of Lund, Sweden, 1981.
- [102] W.H. Press, S.A. Teukolsky, W.A. Vetterling, and B.P. Flannery. *Numerical recipes in C*. Cambridge University Press, 2 edition, 1992.
- [103] E. Rabinowicz. Direction of the friction force. *Nature*, 179:1073, 1957.
- [104] E. Rabinowicz. *Friction and Wear of materials*. Wiley, 1995.
- [105] V.T. Rajan. Optimality of the Delaunay Triangulation. *Rd. Discrete Comput. Geom.*, 12:189–202, 1994.
- [106] M. Raous. Quasistatic signorini problem with Coulomb friction and coupling to adhesion. In P. Wriggers and P. Panagiotopoulos, editors, *CISM Courses and Lectures*, number 384, page 101178. Springer, Wien, 1999.
- [107] M. Raous, M. Jean, and J. Moreau, editors. *Continuum mechanics modelling of large deformation contact with friction*, number 145-158 in Contact Mechanics. Plenum Press, 1995.



- [108] S.H. Reese. *Adaptive Methods for Continuous and Discontinuous Damage Modeling in Fracturing Solids*. PhD thesis, Der Fakultät für Bauingenieurwesen und Geodäsie der Leibniz Universität Hannover, 2007.
- [109] A. Rieger. *Adaptative Algorithmen für thermomechanisch gekoppelte Kontaktprobleme*. PhD thesis, Institute für Baumechanik un Numerische Mechanik, 2002.
- [110] J. Rostami, L. Ozdemir, and B. Nilson. Comparison between CSM and NTH hard rock TBM performance prediction models. Technical report, Colorado School of Mines & The Norwegian Institute of Technology (ISDT, LAS VEGAS NV), 1996.
- [111] J. G. Rots, P. Nauta, G. M. A. Kusters, and J. Blaauwendraad. Smearred crack approach and fracture localization in concrete. *Heron*, 30, 1985.
- [112] F. F. Roxborough and H. R. Phillips. Rock Excavation by Disc Cutter. *Int. J. Rock Mech. Min. Sci.*, 12(12):361–366, 1975.
- [113] F.F Roxborough. Cutting rock with picks. *Min. Engr*, 132:445–454, 1973.
- [114] M.C. Shaw. Dimensionless analysis for wear systems. *Wear*, 43:263–266, 1977.
- [115] H. Si. On Refinement of Constrained Delaunay Tetrahedralizations. *In Proceedings of the 14th International Meshing Roundtable*, 2006.
- [116] H. Si and K. Gaerther. Meshing Piecewise Linear Complexes by Constrained Delaunay Tetrahedralizations. *In Proceedings of the 14th International Meshing Roundtable*, pages 147–163, 2005.
- [117] A. Signorini. Sopra alcune questioni di elastostatica. In *Atti della Societa Italiana per il Progresso delle Scienze*, 1933.
- [118] J. Simo and J. Ju. Strain and stress based continuum damage models-I formulation. *Int. J. Solids Structures*, 23:821–840, 1987.
- [119] J. Simo and J. Ju. Strain and stress based continuum damage models-II computational aspects. *Int. J. Solids Structures*, 23:841–869, 1987.
- [120] J.C. Simo and T.J.H. Hughes. *Computational Inelasticity*. Springer, 1998.
- [121] J.C. Simo and T.A. Laursen. An augmented Lagrangian treatment of contact problems involving friction. *Computers and Structures*, 42:97–116, 1992.
- [122] J.C. Simo and N. Tarnow. The discrete energy momentum method. Conserving algorithms for nonlinear elasto-dynamics. *Journal of Applied Mathematics and Physics*, 43:757–792, 1992.
- [123] R. L. Taylor. Explicit Dynamics with Rigid-Flexible Interfaces. November 1997. First Draft.

- [124] G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième Memoire: Recherches su les paralléloédres primitifs. *Journal fur dir Reine und Angewandte Mathematik*, 134:198–287, 1908.
- [125] E.W. Weisstein. Circumcircle. <http://mathworld.wolfram.com/Circumcircle.html>, 2008.
- [126] E.W. Weisstein. Circumsphere. <http://mathworld.wolfram.com/Circumsphere.html>, 2008.
- [127] J.R. Williams and R. O’Connor. Discrete Element Simulation and the Contact Problem. *Archives Comp. Meth. Engng*, 6(4):279–304, 1999.
- [128] K.L. Woo and T.R. Thomas. Contact of rough surfaces: A review of experimental works. *Wear*, 58:331–340, 1980.
- [129] W. L. Wood. Practical time-steeping schemes. *Clarendon Press, Oxford*, 1990.
- [130] W.L. Wood, M. Bossak, and O.C. Zienkiewicz. An alpha modification of newmark’s method. *International Journal for Numerical Methods in Engineering*, 15:15621566, 1981.
- [131] P. Wriggers. *Nichtlineare Finite-Element-Methoden*. Springer, 2000.
- [132] P. Wriggers. *Computational Contact Mechanics*. Springer, second edition, 2006.
- [133] P. Wriggers and C. Mieke. Contact constraints within coupled thermomechanical analysis-a finite element model. *Computer Methods in Applied Mechanics and Engineering*, 113:301–319, 1994.
- [134] G. Zavarise, P. Wriggers, and U. Nackenhorst. *A Guide for Engineers to Computational Contact Mechanics*. Consorzio TCN scarl, 2006.
- [135] Z. H. Zhong. *Finite Element Procedures for Contact Impact Problems*. Oxford University Press, 1993.
- [136] O.C. Zienkiewicz, B. Boroomand, and J.Z. Zhu. Recovery procedures in error estimation and adaptivity. Part I: Adaptivity in linear problems. *Comput. Methods Appl. Mech. Engrg.*, 176:111–125, 1999.
- [137] O.C. Zienkiewicz and R.L. Taylor. *El Método de los elementos finitos. Formulacion básica y problemas lineales*, volume 1. CIMNE, 4 edition, 2000.
- [138] O.C. Zienkiewicz and R.L. Taylor. *El Método de los elementos finitos. Las bases*. CIMNE, 2000.
- [139] O.C. Zienkiewicz and R.L. Taylor. *El Método de los elementos finitos. Mecánica de sólidos y de fluidos, dinámica y no linealidad*, volume 2. CIMNE, 4 edition, 2000.

- [140] O.C. Zienkiewicz and J.Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part 1. The recovery Technique. *Int. J. Num. Meths. Engrg.*, 33(1):1331–1364, 1992.