



HAL
open science

Steady-state formulation of metal forming processes : Contact coupling and treatment of history-dependent material models with unstructured meshes

Shitij Arora

► **To cite this version:**

Shitij Arora. Steady-state formulation of metal forming processes : Contact coupling and treatment of history-dependent material models with unstructured meshes. Mechanics of materials [physics.class-ph]. Université Paris sciences et lettres, 2020. English. NNT : 2020UPSLM084 . tel-04099109

HAL Id: tel-04099109

<https://pastel.hal.science/tel-04099109>

Submitted on 16 May 2023

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

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



THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à MINES ParisTech

**Steady-state formulation of metal forming processes:
contact coupling and treatment of history-dependent
material models with unstructured meshes**

Formulation Stationnaire de mise en forme des métaux: Couplage de contact et traitement de dépendance d'histoire dans les maillages non-structurés

Soutenue par

Shitij ARORA

Le 24/11/2020

École doctorale n°364

**Ecole Doctorale Sciences
Fondamentales et Ap-
pliquées**

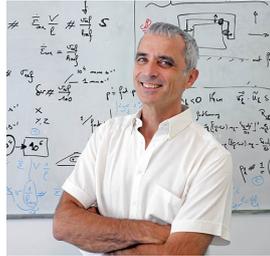
Spécialité

**Mécanique Numérique et
Matériaux**

Composition du jury :

Antoine Courtoux Ingénieur, ArcelorMittal	<i>Examineur</i>
Hubert Geijselaers Professeur, Université de Twente	<i>Rapporteur</i>
Katia Mocellin Maître de Recherches, Mines Paris	<i>Examinatrice</i>
Pierre Montmitonnet Directeur de Recherches, CNRS	<i>Directeur de Thèse</i>
Jean-Philippe Ponthot Professeur, Université de Liège	<i>Président</i>
Ugo Ripert Ingénieur, Transvalor	<i>Examineur</i>
Emmanuelle Rouhaud Professeure, UT Troyes	<i>Rapporteuse</i>

Acknowledgements



I would like to dedicate this manuscript to my ex-supervisor, Prof. Lionel Fourment (deceased), who passed away in 2019 fighting a terminal illness. I would like to thank him (posthumously) for his continued support, follow-up and encouragement, even during the painful periods of his therapy. He left no stone unturned to make sure that I always following the right direction towards my objectives. The principal ideas and formulations on the contact and elastoplasticity presented in this thesis are the outcome of his painstaking efforts and his rich academic knowledge. It is all due to the productive discussions with him that I could get meaningful results really quickly and that I could reach to a fruitful end to this PhD. I would also like to thank Prof. Pierre Monmitonnet for his guidance and agreeing to supervise me in a tough situation and accommodate me among his other projects. It was only due to the long discussions on the elastoplasticity with him and his experiences from the Lam3 formulation that I could get results on this part of the thesis in a very short period of time. It is due to his vast experience that he could always break down a difficult concept to make sure that I followed every detail. Not only did he ensure his diligent availability but he also showcased utmost patience with me, even during the most terrible phases of the PhD. I would also like to thank Prof. Katia Mocellin for her availability for discussions, presenting the subject on my/Lionel's behalf at various international conferences, and remaining helpful in every possible way. I would like to specifically thank Prof. Pierre & Prof. Katia for their support during the preparation of the PhD defense. I would also like to thank Ugo Ripert for sharing his knowledge in the subject from his personal experience in the development of steady-state ForgeNxt[®] and for his tips and tricks in getting things done in a complex code like Forge. I would also like to thank the members of the Forge Stationnaire consortium for sharing their experiences, which also helped me understand the industrial viewpoint. I also thank all the members of the jury for their insightful comments/questions on the manuscript and on the date of PhD defense. I would like to specifically thank my wife Shefali Sudi for always believing in me and making sure that I do not give up. I thank her for giving me a beautiful gift, my son Sharvin, in the final year of my PhD. I would also like to thank my parents and my sister who supported me in every thick and thin and made sure that I reach this milestone.

Contents

Contents	iii
List of Figures	iv
List of Tables	viii
1 Modeling of continuous forming processes	1
1.1 Brief introduction to continuous metal forming processes	2
1.2 Numerical simulation of continuous forming processes	4
1.3 Types of numerical models	5
1.4 Steady-state modeling and context of the project	5
1.5 Overview of the Thesis	7
2 Bibliography on numerical resolution of steady-state processes	11
2.1 Kinematic Description of material flow	12
2.2 Classification of resolution methods for continuous forming processes	16
2.3 Mathematical modeling of continuous forming	19
3 Existing steady-state formulation and limitations	43
3.1 Generalized ForgeNxt®2016 steady-state algorithm	44
3.2 Thermo-mechanical problem formulation	44
3.3 Free-surface resolution formulation	56
3.4 Contact treatment in a steady-state frame	60
3.5 Application to hot-rolling test-cases	62
3.6 Reasoning for instabilities observed with Hot rolling of a long product from oval to square section (VP-Test-Case-02)	69
3.7 Conclusions from the chapter	70
4 A stabilized contact constraint for fixed-point, multi-field, steady-state formulations	75
4.1 Problem with contact coupling	76
4.2 Consistent contact coupling	77
4.3 Description of the nodal form of free-surface equation	78
4.4 Test of new free-surface formulation with analytical test-cases	80
4.5 Updated contact coupling	89
4.6 Performance with hot-rolling test-cases	96
4.7 Conclusions from the chapter	116
5 Steady-state resolution with history dependent EVP behavior	121
5.1 Introduction to elastoplasticity	123
5.2 New ForgeNxt®2016 EVP resolution algorithm	129
5.3 Validation of ForgeNxt®2016 EVP Algorithm-1 with a simple cold rolling problem	139
5.4 Improvement of speedup with ForgeNxt®2016 EVP Algorithm-2	145
5.5 Validation of ForgeNxt®2016 EVP algorithm with complex test cases	147
5.6 Conclusions from the Chapter	153

6	Conclusions and Perspectives	157
6.1	Conclusions	158
6.2	Perspectives	160
A	Appendix	xi
A.1	Finding the bubble shape function N_b	xi
	List of acronyms	xiii
	List of symbols	xv

List of Figures

1.1.1	Representation of discontinuous and continuous processes	2
1.1.2	Flow chart of a steel production industry	3
1.1.3	Typical five-stand tandem cold rolling mill	4
1.1.4	Different types of rolling products	4
1.3.1	Rolling process modeling with coupling with roll deformation	5
1.4.1	The different phases of a metal forming process (rolling) [10]	6
1.4.2	Comparison of steady-state resolution with incremental and steady-state methods	6
1.4.3	Simulation of a shape rolling of a long product with ForgeNxt [®] steady-state solver, released version (ForgeNxt [®] 2016) [10]	7
2.1.1	Lagrangian kinematic description	13
2.1.2	Lagrangian Description	13
2.1.3	Eulerian Description	14
2.1.4	Kinematic Description of material flow with Arbitrarily Lagrangian Eulerian (ALE)	14
2.1.5	ALE Description	14
2.1.6	Mapping among undeformed, deformed and reference frames in the reference frame method by [5]	16
2.2.1	Computation of steady-state shape with ALE approach [41]	17
2.3.1	Modeling material behavior [2]	20
2.3.2	Resolution of history-dependent (elastoviscoplasticity) problems with [18] approach	23
2.3.3	Contact Kinematics for Eulerian formulation	24
2.3.4	Free surface phenomenon in different processes [10]	28
2.3.5	Resolution methods for free surface computation [44] with the assumption of constant velocity vector per element	29
2.3.6	Petrov Galerkin test functions	31
2.3.7	Problem with contact with global resolution [44]	32
2.3.8	Correction of free-surface [60]	32
2.3.9	Different free-surface computation methods tested by [53]	32
2.3.10	Correction of free-surface [18]	33
2.3.11	Contact treatment with Optimal control method [24]	34
2.3.12	Problem with contact with global resolution [44]	34
3.1.1	ForgeNxt [®] 2016 fixed-point iterative steady-state algorithm for hot-rolling problems	44
3.2.1	Schematic representation of a rolling problem	45
3.2.2	Initialization of geometry in [7]	45

3.2.3	Quasi-linear P1+/P1 “mini-element” for \vec{v}, p interpolation	50
3.2.4	Details of the thermo-mechanical resolution in the iterative algorithm	55
3.3.1	free-surface correction: Contact condition and C_k in SUPG test function	56
3.3.2	Vectorial form of free-surface computation with Free-surface correction with 2-Degrees of freedom (DoF) formulation (SUPG-dif) formulation	59
3.3.3	Regularisation of the volume mesh [7]	60
3.3.4	Details of the free-surface resolution step in the iterative algorithm	60
3.4.1	Complete ForgeNxt [®] 2016 fixed-point algorithm with contact coupling	62
3.5.1	Simulation model for Hot rolling of a thick flat rectangular section plate (VP-Test-Case-01) from Constellium	64
3.5.2	Comparison of deformed section (steady-state) shape of the VP-Test-Case-01 with the steady-state and incremental formulations	64
3.5.3	Comparison of equivalent strain for VP-Test-Case-01	65
3.5.4	Comparison of von Mises stress (in MPa) for VP-Test-Case-01	66
3.5.5	Comparison of Tool Force and Equivalent strain convergence for VP-Test-Case-01 with the ForgeNxt [®] 2016 formulation	67
3.5.6	Comparison of free-surface correction convergence and flux loss for VP-Test-Case-01 with the ForgeNxt [®] 2016 formulation	67
3.5.7	VP-Test-Case-02 from Aubert & Duval tested with the steady-state algorithm	68
3.5.8	Comparison of tool force and equivalent strain convergence and flux loss for VP-Test-Case-01 with the ForgeNxt [®] 2016 formulation	69
3.5.9	Comparison of free-surface correction convergence and flux loss for VP-Test-Case-02 with the ForgeNxt [®] 2016 formulation	69
3.6.1	Comparison of equivalent strain, equivalent rate of deformation and contact-status near the tool contact area for VP-Test-Case-02 at two consecutive iterations	70
4.3.1	Finding an optimized value for the SUPG stabilization coefficient	79
4.4.1	Principle of analytical test for free-surface correction	80
4.4.2	Gauss analytical functions tested on 2D sheet	82
4.4.3	h convergence for 2D Gauss function with different formulations for free-surface resolution	82
4.4.4	Sinusoidal analytical function and problems with SUPG-NC method	83
4.4.5	h convergence for 2D Sinusoidal function with different formulations for free-surface resolution	83
4.4.6	Quadratic analytical function tested on 3D cylindrical and square tubes	84
4.4.7	h convergence for quadratic deformation on 3D Cylindrical tube tested with different free-surface resolution formulations	85
4.4.8	h convergence for quadratic deformation on 3D square tube tested with different free-surface resolution formulations	86
4.4.9	Sinusoidal analytical function tested on 3D cylindrical and square tubes	86
4.4.10	h convergence for sinusoidal deformation on 3D cylindrical tube tested with different free-surface resolution formulations	87
4.4.11	h convergence for sinusoidal deformation on 3D square tube tested with different free-surface resolution formulations	88
4.5.1	Location of node l for implicit formulation on 3D (surface) mesh	92
4.5.2	Implementation of Implicit contact constraint	93
4.6.1	Comparison of deformed section (steady-state) shape of the VP-Test-Case-01 with the steady-state and incremental formulations	97
4.6.2	Comparison of equivalent strain for VP-Test-Case-01	98
4.6.3	Comparison of von Mises stress for VP-Test-Case-01	99
4.6.4	Comparison of Tool force convergence for VP-Test-Case-01 with different steady-state formulations	99

4.6.5	Comparison of geometry convergence for VP-Test-Case-01 with different steady-state formulations	100
4.6.6	Comparison of equivalent strain convergence for VP-Test-Case-01 with different steady-state formulations	100
4.6.7	Comparison of flux loss for VP-Test-Case-01 with different steady-state formulations	101
4.6.8	Comparison of deformed section (steady-state) shape of the VP-Test-Case-02 with the steady-state and incremental formulations	102
4.6.9	Comparison of VP-Test-Case-02 steady-state solution from different formulations with the incremental solution	102
4.6.10	Comparison of Tool force convergence for VP-Test-Case-02 with different steady-state formulations	103
4.6.11	Comparison of geometry convergence for VP-Test-Case-02 with different steady-state formulations	103
4.6.12	Comparison of equivalent strain convergence for VP-Test-Case-02 with different steady-state formulations	104
4.6.13	Comparison of flux loss for VP-Test-Case-02 with different steady-state formulations	104
4.6.14	Test-Case 03: Square to Rectangle Shape rolling	105
4.6.15	Coarse and fine mesh models tested for VP-Test-Case-03	106
4.6.16	Rate-temperature dependent stress-strain curve used for resolution of VP-Test-Case-03	107
4.6.17	VP-Test-Case-03: Temperature field (in °C) profile at the input plane Γ_{in} of the VP-Test-Case-03 model	108
4.6.18	Comparison of deformed section (steady-state) shape of the VP-Test-Case-03-adiabatic with the steady-state and incremental formulations	109
4.6.19	Comparison of equivalent strain (on left) and von Mises stress (in MPa, on right) for VP-Test-Case-03-adiabatic with different steady-state contact formulations in comparison to the ForgeNxt [®] inc solution	109
4.6.20	Comparison of deformed section (steady-state) shape of the VP-Test-Case-03-fully coupled with the steady-state and incremental formulations	111
4.6.21	Comparison of axial velocity (on left) and binary contact (on right) for VP-Test-Case-03-fully coupled with Steady-State Implicit in comparison to the incremental solution	112
4.6.22	Comparison of stress in XX (transverse, in MPa, on left) and YY (vertical, in MPa, on right) directions (in MPa) respectively for VP-Test-Case-03-fully coupled with Steady-State Implicit in comparison to the incremental solution	112
4.6.23	Comparison of ZZ (rolling direction, in MPa, on left) direction stress and von Mises stress (in MPa, on right) for VP-Test-Case-03 with Steady-State Implicit in comparison to the incremental solution	113
4.6.24	Comparison of Temperature (left) and Equivalent strain (right) for VP-Test-Case-03-fully coupled with Steady-State Implicit in comparison to the incremental solution	113
4.6.25	Comparison of Tool force convergence for VP-Test-Case-03-fully coupled with different steady-state formulations	114
4.6.26	Comparison of Geometry convergence for VP-Test-Case-03-fully coupled with different steady-state formulations	114
4.6.27	Comparison of Equivalent strain convergence for VP-Test-Case-03-fully coupled with different steady-state formulations	115
4.6.28	Comparison of Temperature convergence for VP-Test-Case-03-fully coupled with different steady-state formulations	115
4.6.29	Comparison of flux loss for VP-Test-Case-03-fully coupled with different steady-state formulations	116
5.1.1	Graphical representation of hardening rules [1]	124

5.1.2	Incremental computation of elastoplastic stress	126
5.1.3	Steady-state computation of elastoplastic stress with structured mesh	128
5.1.4	Iterative resolution of elastoplastic constitutive law for integration on a streamline	128
5.2.1	Problem with adoption of Lam3 algorithm with unstructured mesh	129
5.2.2	Comparison of different strategies for building streamlines in unstructured meshes	130
5.2.3	Methodology of adaption of Lam3 algorithm in ForgeNxt	131
5.2.4	Building streamlines in a domain with unstructured mesh	131
5.2.5	Proposed basic algorithm for ForgeNxt [®] 2016 EVP adapted from [3]	132
5.2.6	P0 to P1 mapping of the state-variables	133
5.2.7	Comparison of P0-P1 field mapping methods- P1 solutions	134
5.2.8	Comparison of P0-P1 field mapping methods- P1 error field	135
5.2.9	Building P1+ field from P0 field known at integration point (centroid of element) and P1 field known at the nodes and interpolation at <i>pre</i>	135
5.2.10	Detailed ForgeNxt [®] 2016 EVP Algorithm-1 with Sub-iter loop	138
5.3.1	Mesh models tested for the pseudo-2D cold rolling (EVP) test-case	140
5.3.2	Convergence of different deviatoric stress components for Cold rolling of a thick flat rectangular section 2D plate (EVP-Test-Case-01)	142
5.3.3	Convergence of deviatoric stress components	142
5.3.4	Comparison of solution for EVP-Test-Case-01 with steady-state and incremental simulations	144
5.3.5	Comparison of solution for EVP-Test-Case-01 with steady-state and incremental simulations	144
5.3.6	Comparison of equivalent stress and strain fields with different simulations	144
5.4.1	Algorithm-2 for EVP steady-state resolution with streamline loop in ForgeNxt [®] 2016	145
5.4.2	Convergence of different deviatoric stress components for EVP-Test-Case-01-d with ForgeNxt [®] 2016EVP Algorithm-1 & Algorithm-2	147
5.5.1	Mesh model for the Cold rolling of flat rectangular section 3D plate (EVP-Test-Case-02)	148
5.5.2	Comparison of solution for EVP-Test-Case-02 with steady-state and incremental simulations	149
5.5.3	Comparison of output plane shape (Initial half-width: 40 mm) for EVP-Test-Case-02 with incremental and steady-state formulations	149
5.5.4	Convergence of deviatoric stress components for EVP-Test-Case-02 with steady-state algorithms	150
5.5.5	Mesh models used for the Test-case 03	150
5.5.6	Comparison of solution for Cold rolling of a long product (EVP-Test-Case-03) with steady-state and incremental simulations	151
5.5.7	Comparison of output plane shape for EVP-Test-Case-03 with incremental and steady-state formulations	151
5.5.8	Convergence of deviatoric stress components for EVP-Test-Case-03 with steady-state algorithms	152
6.2.1	Unsymmetric rolling cases	160
6.2.2	Tube Rolling test case forces Fx (red) Fz (blue) on roll stand 1 (left) and 2 (right) computed with incremental and steady-state algorithms	160
6.2.3	Tube Rolling test case forces Fx (red) Fz (blue) on roll stand 3 (left) and 4 (right) computed with incremental and steady-state algorithms	161
6.2.4	Tube Rolling test case forces Fx (red) Fz (blue) on roll stand 5 computed with incremental and steady-state algorithms	161
6.2.5	Steady-state model necessary for modeling the interstand phenomena	162
6.2.6	Accounting for the boundary condition at output plane Γ_{out} [1]	162
6.2.7	Algorithm-3 for EVP steady-state resolution with streamline loop in ForgeNxt [®] 2016	163

A.1.1 Sub-elements inside the tetrahedral element	xi
---	----

List of Tables

2.3.1 Comparison of resolution with VP, EVP and EP models [19]	23
3.5.1 Rheological and tribological properties of the test-cases	63
3.5.2 Global performance of the VP-Test-Case-01 with the ForgeNxt [®] 2016 algorithm in comparison to the ForgeNxt [®] inc solution (on 12 cores)	65
3.5.3 Global performance of the VP-Test-Case-02 with the ForgeNxt [®] 2016 algorithm in comparison to the ForgeNxt [®] inc solution (on 12 cores)	68
4.4.1 Different analytical test functions tested with free-surface formulation	81
4.4.2 Comparison of L_2 and L_∞ % errors for the Gauss function deformation on a 2D flat sheet shown in Figure 4.4.2	82
4.4.3 Comparison of L_2 and L_∞ % errors for the sinusoidal function deformation on a 2D flat sheet shown in Figure 4.4.4	83
4.4.4 Comparison of L_2 and L_∞ % errors for the quadratic function deformation on a 3D cylindrical tube shown in Figure 4.4.6a	85
4.4.5 Comparison of L_2 and L_∞ % errors for the quadratic function deformation on a 3D square tube shown in Figure 4.4.6b	85
4.4.6 Comparison of L_2 and L_∞ % errors for the sinusoidal function deformation on a 3D cylindrical tube shown in Figure 4.4.9a	87
4.4.7 Comparison of L_2 and L_∞ % errors for the sinusoidal function deformation on a 3D square tube shown in Figure 4.4.10b	87
4.6.1 Global performance of the VP-Test-Case-01 with the updated steady-state algorithm in comparison to the ForgeNxt [®] inc solution (on 12 cores)	97
4.6.2 Global performance of the VP-Test-Case-02 with the updated steady-state algorithm in comparison to the incremental solution (on 12 cores)	101
4.6.3 Properties used for the VP-Test-Case-03	106
4.6.4 Global performance of the VP-Test-Case-03-adiabatic with the updated steady-state algorithm in comparison to the incremental solution (on 12 cores)	108
4.6.5 Global performance of the VP-Test-Case-03-fully coupled with the updated steady-state algorithm in comparison to the incremental solution (on 12 cores)	111
5.3.1 Hensel-Spittel properties used for the EVP resolution of EVP-Test-Case-01	140
5.3.2 Different models and properties used for the VP initialization	141
5.3.3 Different EVP-Test-Case-01 variants tested	141
5.3.4 Comparison of global results for EVP-Test-Case-01 from ForgeNxt [®] 2016 EVP Algorithm-1 in comparison to the ForgeNxt [®] incremental solver (ForgeNxt [®] inc) resolution (on 1 core)	143
5.4.1 Comparison of global results for EVP-Test-Case-01-d with the ForgeNxt [®] 2016 EVP Algorithm-1 and Algorithm-2 in comparison to the ForgeNxt [®] inc resolution (on 1 core)	147
5.5.1 Comparison of global results for EVP-Test-Case-02 from the ForgeNxt [®] 2016 EVP Algorithm-1 and Algorithm-2 in comparison to the ForgeNxt [®] inc resolution (on 1 core)	148

5.5.2 Comparison of global results for EVP-Test-Case-03 from the ForgeNxt® 2016 EVP in comparison to the ForgeNxt® inc resolution (on 1 core) 151

Chapter 1

Modeling of continuous forming processes

Contents

1.1	Brief introduction to continuous metal forming processes	2
1.2	Numerical simulation of continuous forming processes	4
1.3	Types of numerical models	5
1.4	Steady-state modeling and context of the project	5
1.5	Overview of the Thesis	7

1.1 Brief introduction to continuous metal forming processes

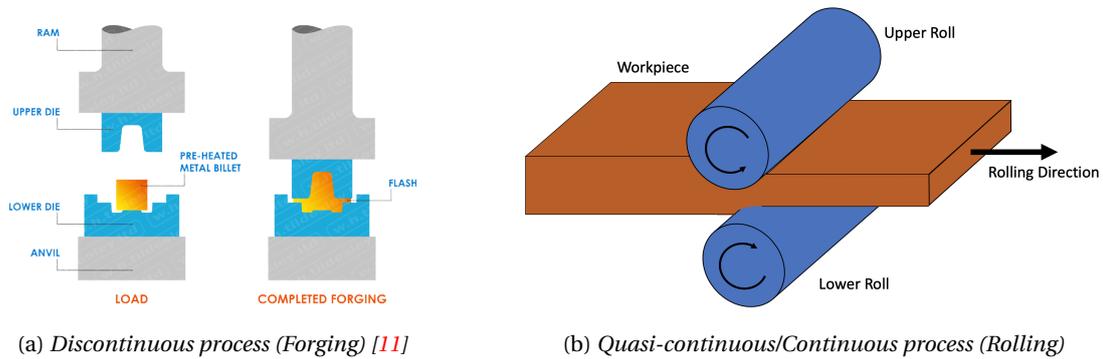


Figure 1.1.1: Representation of discontinuous and continuous processes

Material processing is a quintessential need of the human for creating products that cater to our diverse needs in every stratum of life. A material processing method can be categorized into different types depending on whether it is used: to reshape (for ex. forming), to cut (for ex. sawing, planing, broaching, drilling, grinding, turning and milling), to join (welding, soldering) or to add material (casting, additive manufacturing). The principal interest of the present thesis is in the forming processes that enable the transformation of a product by changing the shape of an input workpiece. The forming processes can be further categorized into *discontinuous* and *continuous* processes: the former has a discrete operation cycle, for example forging (see Figure 1.1.1a), and the latter has a quasi-continuous operation cycle like rolling, extrusion etc. (see Figure 1.1.1b). An important difference between the two categories is that the continuous processes have a considerably large steady-state regime in comparison to the transient phases of material engagement & release.

The continuous forming of metal changes the shape of an input workpiece by virtue of its plastic deformation. Depending on the direction of loading, the process could be uniaxial (like flat product rolling) or multi-axial (like long-product rolling, extrusion, etc.). In other classification, the temperature of the product distinguishes if it is a hot or a cold-forming process. In hot-forming, elastic retraction becomes insignificant, hence only plastic behavior needs to be accounted for, whereas, in cold-forming, the elastoplastic constitutive behavior assumptions allow us to model the residual stresses. The rate-dependent effects in elastic or plastic components can be modeled with viscoelastic or viscoplastic/ elastoviscoplastic material models respectively. Among the vast range of continuous metal forming processes, the present thesis is mainly focused on the *rolling process* that corresponds to traction and compression of the material between a pair(s) of counter-rotating rolls or rolling mills. The material can be rolled as soon as the metal solidifies (see Figure 1.1.2), and becomes sufficiently strong to withhold the compression forces. Even though the examples presented for the current context are inspired from the steel industry, its application is general.

Classification of rolling processes

As seen in the Figure 1.1.2, the continuous casting step makes it possible to obtain very long products right after the iron is smelted from its ore, and yields semi-finished products in shapes of blooms, billets, and slabs [7]. Further, the metal is subjected to plastic deformation in successive rolling passes commonly in roughing and finishing mills to achieve the desired shape and dimensions. The rolls are generally sets of cylinders arranged in pairs and housed in a stand. They either reduce the thickness and/or change the shape of workpiece material by exerting force, measured in hundreds or even thousands of tonnes. During the forming process, the temperature, the strain, and the strain-rate distributions in the rolled material play a significant role in the metallurgical

kinetics.

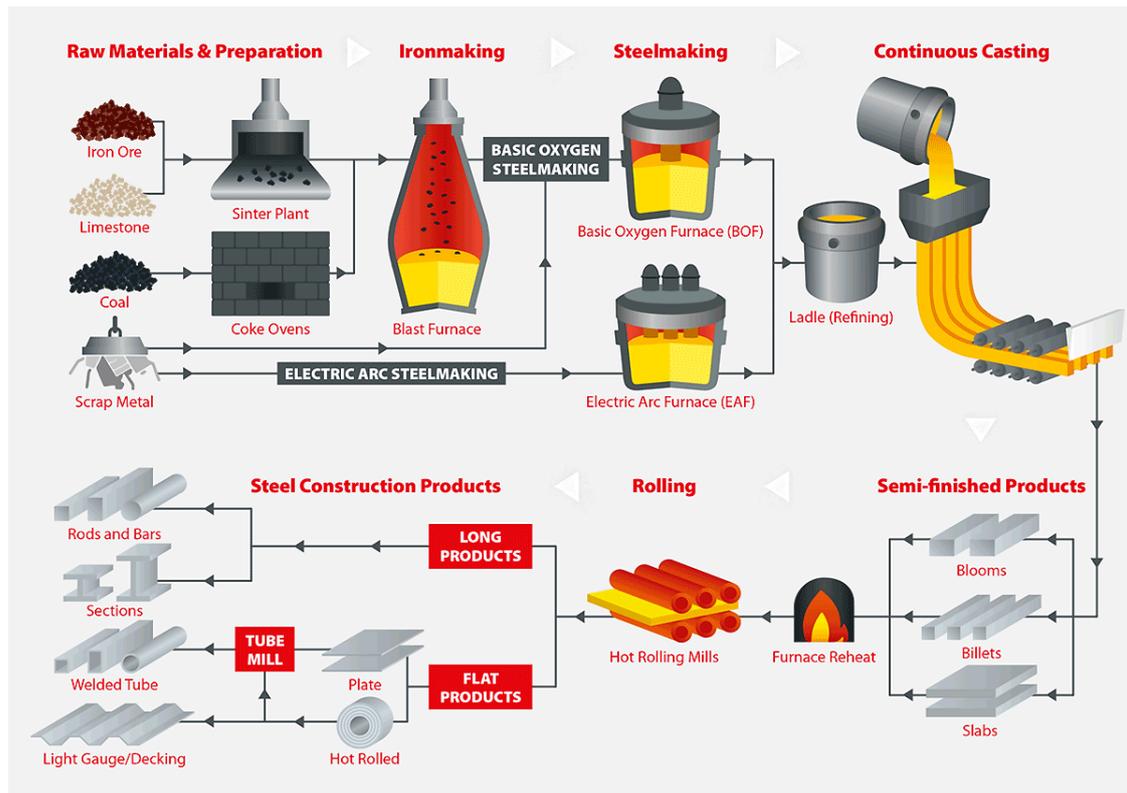


Figure 1.1.2: Flow chart of a steel production industry [7]

The rolling process/products can be categorized as follows:

Hot Rolling vs Cold Rolling The resistance of steel to the change of shape reduces at high temperatures, so it is generally (hot) rolled from semi-finished products by controlling the temperature above the *re-crystallization starting temperature*, generally $0.5T_m$, in a reheat furnace. Here T_m is the melting temperature in Kelvin (K). This increases material workability and reduces the flow stress of the deforming metal. On cooling the workpiece to the room temperature (by normalizing), the hot rolled steel shrinks without any residual stresses, but this shrinkage is non-uniform and gives insufficient control on the size and shape of the finished product and a scaled finish due to the oxidation. These properties make hot-rolled steel most suitable for structural components and other applications such as Railroad tracks, I-beams, agricultural equipment, sheet metal, automotive frames, etc., where precise shapes and tolerances are of less importance.

Below certain sections or thickness, the metal cools down due to the contact with the rolls, and it becomes impossible to continue with the hot rolling. Further reduction of size, if needed, must be performed in cold by different mills which are necessarily designed for cold conditions. The word *cold* means that the rolling of the material is carried out below $0.5T_m$ or at room temperature. The cold-rolled products have higher strength because of strain-hardening; it is not the case with hot rolling as re-crystallization results in softening and jointly to the relaxation of residual stresses. We can achieve very precise shapes with cold rolling since the process is performed at room temperature and thus the steel does not shrink as it cools. Hence it is possible to go down to thinner gauges or smaller sections, especially with tighter tolerances, which are impossible with hot rolling. In addition, the cold rolling results in products with improved surface finish as there is no oxidation. The tandem cold rolling process consists of passing a metal strip through a sequence of pairs of independently driven work rolls, with each work roll supported by a backup roll of a larger diameter. Figure 1.1.3 shows a typical five-stand cold rolling process. As the strip moves through the individual pairs of work rolls, its thickness is successively reduced by very high com-

pression stress in a small region (i.e. the roll bite) between the rolls. In terms of production cost, the hot rolled steel is generally more economical than cold rolled steel as no reheating is required. In addition, there are fewer operations involved in hot rolling, which reduces the cost.

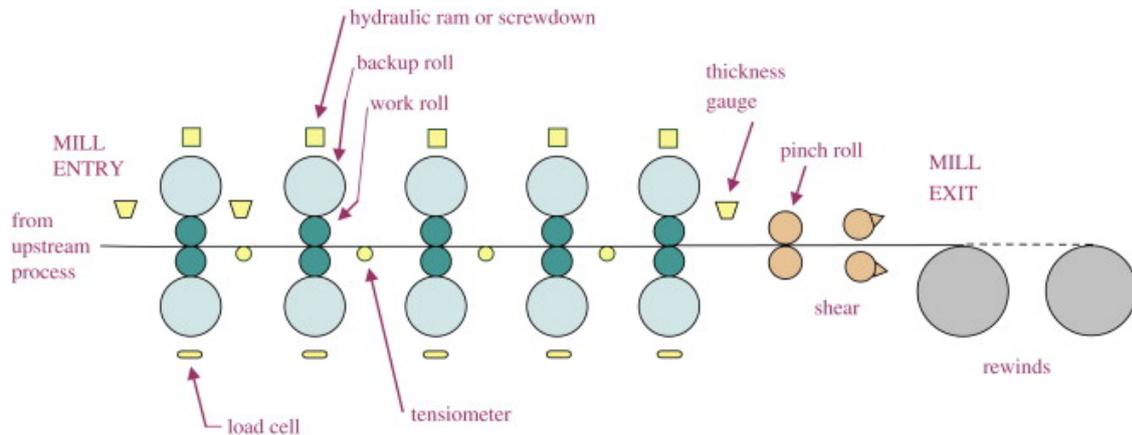


Figure 1.1.3: Typical five-stand tandem cold rolling mill [9]

Types of Rolling products There are two main classes of the rolled products: *flat products* that are sheets (see Figure 1.1.4a) or strips (see Figure 1.1.4b) of uniform thickness, and *long products* (see Figure 1.1.4c) that have a particular cross-section, ranging from rectangular bars to double flange H sections. The flat products are rolled between two horizontal rolls placed one above the other in an open housing, whereas, the long products use a series of specially shaped and angled rolls or stands and are used to transform the section to the required shape.



(a) Flat plate rolling

(b) Strip rolling

(c) Long product

Figure 1.1.4: Different types of rolling products

1.2 Numerical simulation of continuous forming processes

Numerical simulation is an excellent tool for the detailed understanding of the forming process physics. It allows one to analyze the influence of various design & process parameters and access information that is generally difficult or impossible to obtain during the course of deformation, for example, temperature/strain in the core. Numerical simulation is also an important tool used for the optimization of the process aiming at reduction of expended resources, and improvement of the product quality. In addition, the numerical simulation is relatively inexpensive in comparison to the experimental testing. The history of numerical modeling of continuous metal forming is a long journey in itself; it entailed looking into the problems from different interdisciplinary viewpoints for deep understanding. The numerical tools has nevertheless been evolving with time and has enabled modeling larger geometric and material complexities. The numerical methods

can model the process in the dependence of time (steady or unsteady state), space (2D or 3D), and complexity of physics (coupled-physics, multi-scale) for simulating the process as close as possible to the reality. The usage of the high fidelity tools has enabled many industries to produce products with complex shapes and excellent precision and also optimizing the process by reducing material wastage. As a consequence, the numerical simulation tools occupy an important place in the manufacturing industry as they are essential for the realization of high-quality products and for increasing profit margins in the present competitive international market.

1.3 Types of numerical models

The numerical models for simulating a forming process can be classified as *online* or *offline*. The former aims in determining, almost in real-time, solutions to readjust certain parameters such as the position of the tools or their rotation speed. These methods use either simple closed-form solution models or are based on interpolation solutions [12] and can help to set up the rolling mills in the real-time. The latter is based on much more complex and enriched models [1–3] to simulate the behavior of the material in all its details and complexity. The augmented computational power and the refinement of the formulations make possible the detailed analyses of the mechanical and thermal fields of the material. They can as well predict any defect or find corrective measures [6] to forming defects (shape, width, strip profile, flatness, with tighter tolerances), residual stresses, metallurgical defects, and surface quality. A full time-dependent 3D Finite element simulation involving roll stand as well as workpiece deformation (see Figure 1.3.1), production line modeling [5], complete thermal coupling, with microstructure and surface evolution models with precise physical parameters makes possible to predict all the aforementioned forming problems. In fact, to obtain the desired geometry, the section is progressively reduced by rolling through several successive passes. These simulations cannot be run in parallel since each successive pass depends on the solution from the preceding one. Hence the cost of computation with this detailed modeling is enormous and occasionally becomes impractical without a judicious simplification. The scope of the current thesis is focused on the *offline methods* but with alternative approaches to reduce the computation time.

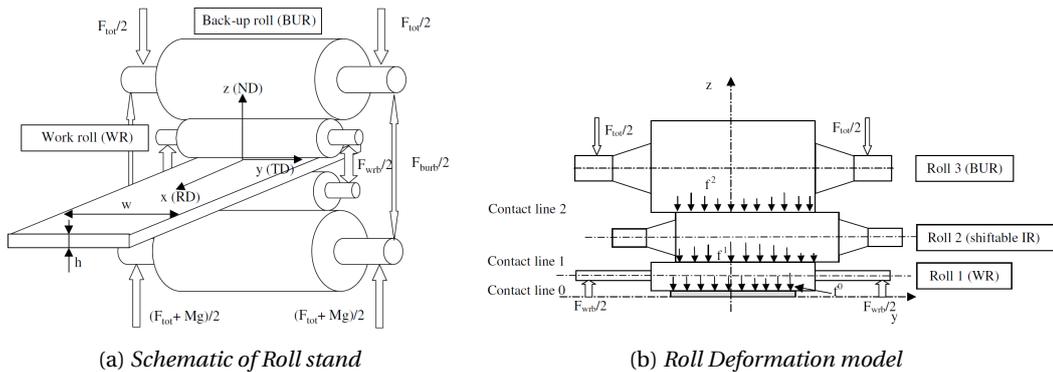


Figure 1.3.1: Rolling process modeling with coupling with roll deformation [5]

1.4 Steady-state modeling and context of the project

The continuous metal forming processes can be described with distinctive: (a) material engagement, (b) steady-state, and (c) release, phases as seen in Figure 1.4.1. The engagement and the release phases of the process are of transient nature, whereas in the steady-state phase, the material flow remains in a quasi-stationary state. The steady-state phase is often the longest of the three phases and often (depending on the industrial context) the most important one. The rolling process results in a product that has uniform properties in the steady-state part and the extreme ends of the product are sheared off after the material processing.

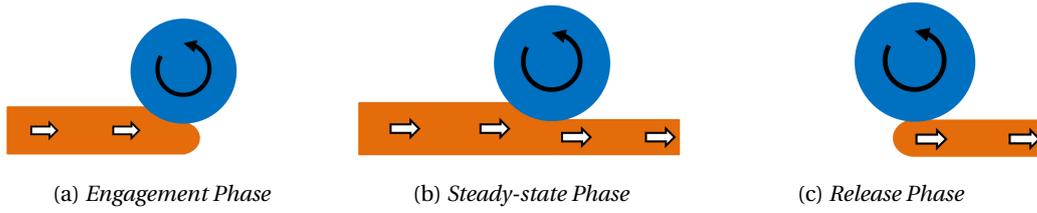


Figure 1.4.1: The different phases of a metal forming process (rolling) [10]

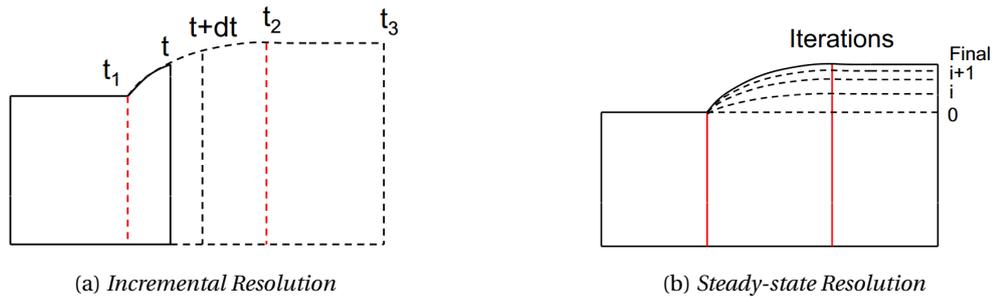


Figure 1.4.2: Comparison of steady-state resolution with incremental and steady-state methods

The computation of steady-state phase of a continuous forming process from numerical modeling necessitates that a sufficient length of the geometry is modeled. The deformation of the material occurs principally under the tools and this region is relatively small compared to the total length of the work-piece modeled. Therefore, the accuracy of the calculation is mainly dependent on the refinement under the tool contact as well as the downstream which is necessary to maintain the solution (of geometry and gradient variables). In general, these processes are modeled with *incremental methods* (see Figure 1.4.2a), and the kinematics is described with a time-dependent partial differential equation. They are adapted to large deformations and can be used to solve a large number of (even discontinuous) forming problems (forging, hammering, drawing). However, the calculation times involved are quite substantial and can become excessive; from a few hours to several days, or several weeks according to the desired accuracy and the method considered. In recent years, CEMEF has been striving to develop new numerical methods dedicated to the simulation of continuous or semi-continuous processes with a small contact and deformation zone, such as rolling, spinning, or drawing, with the objective of reducing the computation time. A pseudo-incremental ALE approach by [8] was one of these lines of research. It offers accelerations between 2 to 7 times depending on the configuration studied, but this approach has not proven to be the most efficient for simulating steady-state processes in comparison to the direct resolution with the steady-state formulation. The latter is undoubtedly the fastest but is difficult to formulate as the forming problem generally results in a strongly coupled multi-field form with each unknown solved using a separate equation. Generally, with these formulations, the steady-state could be found directly using iterative search (see Figure 1.4.2b). Ugo Ripert's thesis [10] was thus dedicated to seeking a new stationary formulation which was adapted in the ForgeNxt[®] 2016, and was compatible with the existing velocity and pressure formulation, unstructured tetrahedral meshes and parallel computation following a domain partitioning approach. Although the proposed formulation was quite satisfactory in terms of accuracy for some of the test cases studied, yet, it was limited to hot forming problems with simple contact surfaces between the tool and the workpiece. With complex geometries, the resolution with an existing algorithm often results in an unstable solution. The present thesis is a continuation work in this direction with the principal objective of the improvement of robustness and enrichment of the formulation to model elasto-plasticity based history-dependent materials for simulating cold forming processes.

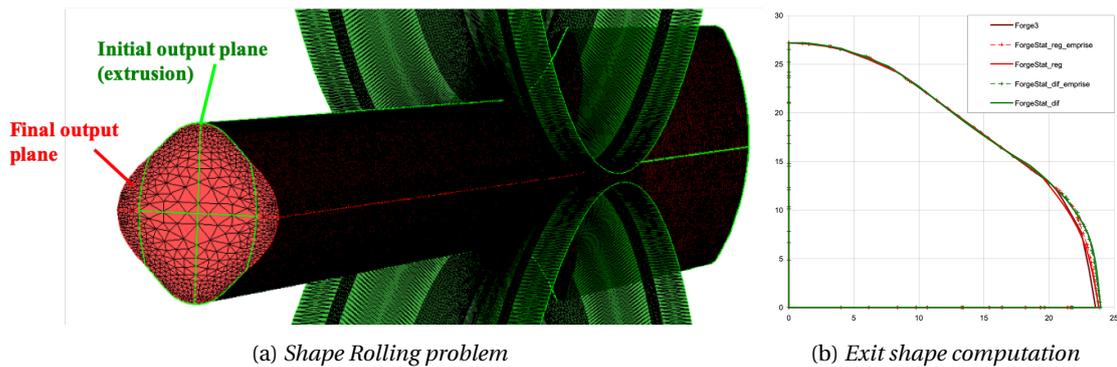


Figure 1.4.3: Simulation of a shape rolling of a long product with [ForgeNxt®2016](#) [10]

Thus, a new project *Forge Stationnaire Bis* was envisaged by a consortium of seven industrial partners which are principally concerned with rolling of metal products, namely *Arcelor Mittal*, *Ugitech*, *Vallourec*, *Lisi-Aerospace*, *Framatome* and *Aubert-Duval*. *Transvalor S.A.* was involved as the official producer and seller of the [Transvalor S.A. ForgeNxt®](#) software. The principal objectives of this project are listed below:

1. Firstly, an improvement of the algorithm robustness is envisaged for improving its performance with more complex cases. In consequence, the objective is to first identify the reason for the instability in the solution with the existing formulation. Further, a remedy to this problem must be proposed and the claim must be tested with more complex hot rolling problems. The validation of the simulation results with the new formulation is to be done with the results obtained from the time-dependent incremental resolution of the same problem
2. Secondly, the existing steady-state formulation is to be extended to cold forming problems with considerable elastic effects at material unloading. Hence the formulation must be able to treat elastoplasticity and elastoviscoplasticity based history-dependent constitutive laws. There are several challenges involved in this problem, mainly being able to trace the history variables inside an unstructured mesh domain. In addition, special attention must also be made to minimize the state-variables diffusion in order to find a reasonably accurate solution, yet faster than the incremental method.

1.5 Overview of the Thesis

Chapter 2 is focused on the bibliographic study of the numerical steady-state modeling of the continuous forming processes. The chapter presents an introduction to the kinematic description of material flow in forming applications, the basic equations, their resolution with incremental and steady-state formulations highlight the basic equations involved, and different numerical strategies possible to resolve these equations. More specifically it highlights a brief introduction to the contact formulation and treatment of history dependency while resolving these equations. The former is also listed as the reason for the solution instability with the existing [ForgeNxt®2016](#) solver developed by Ugo Ripert in his PhD thesis [10]. Even though there is not much literature available in the two subjects with the unstructured meshes, we would invoke the ideas based on structured meshes and understand their possibilities and limitations for their adaption to [ForgeNxt®2016](#).

Chapter 3 presents the details of the existing [ForgeNxt®2016](#) algorithm and the related formulation. This chapter is a preliminary read to understand the existing steady-state formulation developments in Forge in order to understand the improvements proposed in chapters 4 and 5. This chapter also presents the resolution of model hot rolling test cases and the instability problem observed in the steady-state solution with the existing solver. This problem is the motivation for the first objective of this thesis which is covered in Chapter 4.

Chapter 4 proposes a modification of the existing [ForgeNxt[®]2016](#) algorithm for the improvement of its robustness. The main change proposed is consistent contact coupling. A new nodal form of the contact constraint is derived for velocity computation using a *nodal condensation* technique. This constraint is firstly developed in explicit form and the updated steady-state solver is termed as [Steady-State Explicit](#). In the consequence, a further improved version is envisaged with an implicit velocity constraint in the [Steady-State Implicit](#) steady-state solver. The modified solvers are tested on multiple industrial flat and shape rolling test cases proposed in the project and the impact of the contact coupling consistency on the solution accuracy and stability is studied. It means that the updated solver is compared for accuracy with the solution obtained from the time-dependent or incremental resolution in [ForgeNxt[®]inc](#) and is compared for stability with the solution from the [ForgeNxt[®]2016](#) solver.

Chapter 5 extends the iterative steady-state algorithm to elastoplasticity and elastoviscoplasticity materials, which are history-dependent. An initial algorithm based on streamline integration is adopted from the work of [4] and the required framework for building streamlines in the unstructured mesh is developed. After an initial assessment of the algorithm, a further improved version is proposed to speed up the resolution. Both the algorithms are implemented in [ForgeNxt[®]2016](#) tested on fictitious flat and shape cold rolling test cases with unstructured mesh. The obtained solution is validated with the results from the incremental resolution of the same problem in [ForgeNxt[®]inc](#).

Chapter 6 presents the summary of the developments and conclusions on the important improvements brought to the existing steady-state formulation. Further, the open questions unanswered in this thesis are presented in the perspectives.

Bibliography

- [1] J. Donea, A. Huerta, J.P Ponthot, and A. Rodríguez-Ferran. Arbitrary Lagrangian-Eulerian methods. *Encyclopedia of Computational Mechanics Second Edition*, pages 1–23, 2017.
- [2] M.S. Gadala, M.R. Movahhedy, and J. Wang. On the mesh motion for ALE modeling of metal forming processes. *Finite Elements in Analysis and Design*, 38(5):435–459, 2002.
- [3] S. Guerdoux. *Numerical simulation of the friction stir welding process*. PhD thesis, ENSMP, 2007.
- [4] A. Hacquin. *Modélisation Thermomécanique Tridimensionnelle du Laminage - Couplage Bande/ Cylindres*. PhD thesis, ENSMP, 1996.
- [5] P. Montmitonnet. Hot and cold strip rolling processes. *Computer Methods in Applied Mechanics and Engineering: Special issue*, 195(48):6604 – 6625, 2006. ISSN 0045-7825.
- [6] P. Montmitonnet, L. Fourment, U. Ripert, Q. Ngo, and A. Ehrlacher. State of the art in rolling process modelling. *BHM Berg- und Hüttenmännische Monatshefte*, 08 2016. doi: 10.1007/s00501-016-0520-4.
- [7] NSC. An introduction to steel construction products, 2018. URL <https://www.newsteelconstruction.com/wp/an-introduction-to-steel-construction-products/>.
- [8] S. Philippe. *Développement d'une formulation arbitrairement Lagrangienne Eulérienne pour la simulation tridimensionnelle du laminage de produits plats*. PhD thesis, ENSMP, 2009.
- [9] J. Pittner and M.A. Simaan. Optimal control of continuous tandem cold metal rolling. In *2008 American Control Conference*, pages 2834–2839. IEEE, 2008.
- [10] U. Ripert. *Méthode itérative de recherche de l'état stationnaire des procédés de mise en forme : application au laminage*. PhD thesis, ENSMP, 2014. URL <http://www.theses.fr/2014ENMP0006>.
- [11] W.H. Tildesley. Drop forging, 2018. URL <https://www.whtildesley.com/processes/drop-forging>.
- [12] S.H. Zhang, G.L. Zhang, J.S. Liu, C.S. Li, and R.B. Mei. A fast rigid-plastic finite element method for online application in strip rolling. *Finite Elements in Analysis and Design*, 46(12): 1146 – 1154, 2010. ISSN 0168-874X. doi: <https://doi.org/10.1016/j.finel.2010.08.005>. URL <http://www.sciencedirect.com/science/article/pii/S0168874X10001307>.

Chapter 2

Bibliography on numerical resolution of steady-state processes

Contents

2.1	Kinematic Description of material flow	12
2.1.1	Lagrangian description	12
2.1.2	Eulerian description	13
2.1.3	ALE Description	14
2.1.4	Reference frame kinematic description	15
2.2	Classification of resolution methods for continuous forming processes	16
2.2.1	Incremental Resolution	16
2.2.2	Non-incremental Resolution	17
2.3	Mathematical modeling of continuous forming	19
2.3.1	Conservation laws	19
2.3.2	Material constitutive behavior	19
2.3.3	Elasto-viscoplasticity in the context of steady-state formulations	20
2.3.4	Boundary conditions: Contact	23
2.3.5	Free-surface computation	28
2.3.6	Contact treatment with free surface computation	31

The numerical modeling of metal forming processes entails the resolution of a thermo-mechanical problem arising from the equilibrium equations. The said problem is solved to compute the unknown displacement/velocity and temperature fields. With the finite element method, the infinite basis problem is reduced to a finite basis and solved on a time-space grid. In the incremental formulation, at each time-increment, the equilibrium equations are solved to find the unknown displacement/velocity field, followed by the shape update from the displacement field in a displacement-based formulation, or the time integration of velocity in a velocity-based formulation. In principle, incremental methods can be used to resolve both continuous and discontinuous forming problems (see Figure 1.1.1). In the former, both the transient and stationary phases of the process can be modeled. When the steady-state is reached, both the deformed shape and state variables at the exit of an imaginary control volume (as represented in Figure 1.4.1b) around the tool, do not change with time. The incremental formulation is particularly well adapted to the description of transient phenomena. It is thus possible to analyze the defects at the ends of the workpiece [37]. In addition, monitoring of the free-surfaces and the transport of state variables is natural. Indeed, the mesh is attached to the material and is deformed with it. The state variables are stored at nodes or elements and they are therefore naturally updated at each time increment. However, the incremental resolution approach to find the steady-state is computationally costly. It requires resolution of the time-dependent problem on a large domain which may involve large mesh distortion and hence re-meshing becomes necessary. The objective of this chapter is to study in detail the possible methodologies that have been used in literature to model the steady-state of continuous forming processes using alternative formulations. The principal focus is on the background of contact description and modeling history-dependent materials [Elasto-viscoplastic \(EVP\)](#) and [Elastoplastic \(EP\)](#) in the context of steady-state formulations.

2.1 Kinematic Description of material flow

The kinematic description is the mathematical description of the motion of the material particles. This description can be different depending on the frame of reference from which the particles are observed. Either each material point can be individually followed in the material frame R_X or the flow of material particles can be observed from a stationary point in the spatial frame R_x . For modeling steady-state of the continuous forming processes, other reference frame based approaches, like [ALE](#) method, can also be used. Each of these kinematic descriptions has its advantages and disadvantages. A short summary of these ideas are presented below:

2.1.1 Lagrangian description

Lagrangian description is the most common method for the material motion in Solid mechanics. A material/ Lagrangian frame R_X is used such that each material particle is tracked. The mesh is attached to the material points and moves with them. The map ϕ (in Figure 2.1.1) represents the motion of particles with material coordinates \mathbf{X} in the reference configuration Ω_X to the spatial ones \mathbf{x} in the current configuration Ω_x . The material motion is represented with the equation (2.1.1). The mesh deformation corresponds to the material deformation, as shown in Figure 2.1.2. The velocity \vec{v} of a particle is defined with equation (2.1.2), with $|_X$ representing that the material coordinate \mathbf{X} is kept fixed.

$$\mathbf{x} = \phi(\mathbf{X}, t) \quad (2.1.1)$$

$$\vec{v} = \frac{\partial \mathbf{x}}{\partial t} \Big|_X \quad (2.1.2)$$

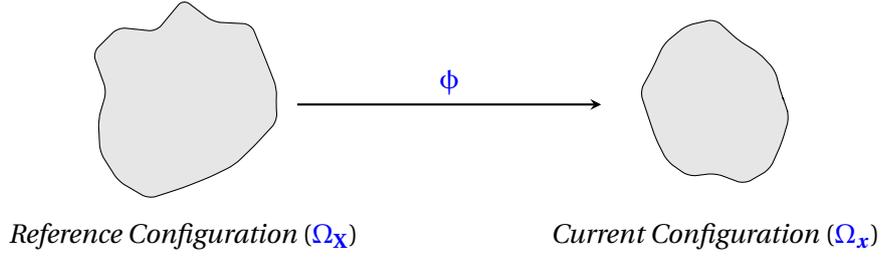


Figure 2.1.1: Lagrangian kinematic description

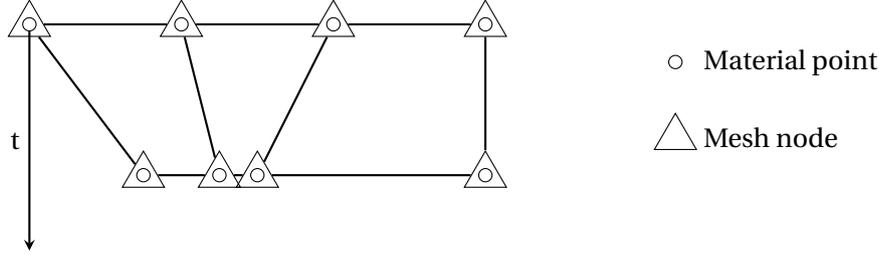


Figure 2.1.2: Lagrangian Description

Since the material points coincide with the mesh during the material motion, there are no relative motion. Hence, there are no convective effects in Lagrangian calculations, meaning, the material derivative of a variable reduces to a simple time derivative. The fact that the computational grid is always attached to the same material point, gives a significant advantage from the computational viewpoint. The evolution of the history-dependent variable f can be described with the material derivative of the variable. It is understandable that the history-dependency of the material is treated naturally [7] and the free-surface is automatically calculated.

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t}|_x \quad (2.1.3)$$

The Lagrangian formulations can be classified into: (i) **Total Lagrangian formulation (TL)**, in which the initial configuration (at t_0) is used as the reference, and (ii) **Updated Lagrangian formulation (UL)**, in which the configuration at time t is used as the reference. The TL formulation is not quite robust while dealing with large deformations (as in metal forming), so the UL formulation is generally used in such scenarios.

2.1.2 Eulerian description

In the Eulerian description, the material flow is observed from a fixed point in space, and hence the frame of reference is spatial R_x . In the said description, the nodes and elements remain fixed and the material flows through the mesh (as seen in Figure 2.1.3). The computation domain Ω_x is a control volume and the spatial coordinates are referred with x . This description is commonly used to simulate fluid mechanics problems where the flow gradient is large and the problem is easier to pose on spatial coordinates. The mesh grid is constructed in the control volume whose shape does not evolve (fixed). The boundary conditions are applied at the fixed mesh nodes. There is a relative motion between the mesh and the material particles, thus the material derivative of the history-dependent variable f results in a temporal as well as a convective part as represented in the equation (2.1.4).

$$\frac{\partial f}{\partial t}|_x = \frac{\partial f}{\partial t}|_x + \vec{v} \cdot \nabla_x f \quad (2.1.4)$$

The left hand term with $|_x$ is the material derivative and the temporal part with $|_x$ describes the spatial time derivative. When the steady-state is reached, the temporal term ceases to exist and the history-dependent variable changes only due to the convection effects.

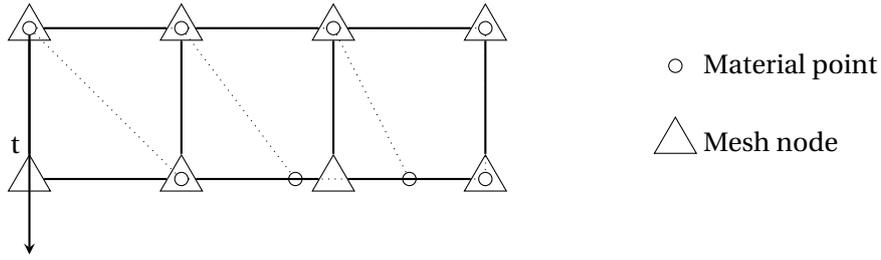


Figure 2.1.3: Eulerian Description

2.1.3 ALE Description

Another manner to describe the motion particles is with the **ALE** method that combines the interesting aspects of the classical kinematic description methods while minimizing their drawbacks as far as possible. In **ALE** method, neither the material configuration $\Omega_{\mathbf{x}}$ nor the spatial configuration $\Omega_{\mathbf{x}}$ is used as the reference. A third domain called the ALE referential configuration $\Omega_{\boldsymbol{\chi}}$ with coordinates $\boldsymbol{\chi}$ is introduced to identify as grid points. The referential domain is mapped to the material and spatial domains as shown in Figure 2.1.4. The map ϕ describes (see equation (2.1.5a)) the particle motion. This map is dissociated into the map ζ^{-1} representing the particle motion (equation (2.1.5b)) with respect to the referential grid and ψ represents the grid motion (equation (2.1.5c)) in the spatial domain. The velocities of the material and the mesh with reference to grid are given with equation (2.1.5d). The relative velocity \vec{c} between the mesh and the material is thus found with equation (2.1.5e) and is the convective velocity used for the transport of state-variables.

$$\mathbf{x} = \phi(\mathbf{X}, t) \quad (2.1.5a)$$

$$\boldsymbol{\chi} = \zeta^{-1}(\mathbf{X}, t) \quad (2.1.5b)$$

$$\mathbf{x} = \psi(\boldsymbol{\chi}, t) = \psi(\zeta^{-1}(\mathbf{X}, t), t) \quad (2.1.5c)$$

$$\hat{w} = \frac{\partial \boldsymbol{\chi}}{\partial t} \Big|_{\mathbf{x}}; \quad \hat{v} = \frac{\partial \mathbf{x}}{\partial t} \Big|_{\boldsymbol{\chi}} \quad (2.1.5d)$$

$$\vec{c} = \vec{v} - \hat{v} = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\chi}} \cdot \hat{w} \quad (2.1.5e)$$

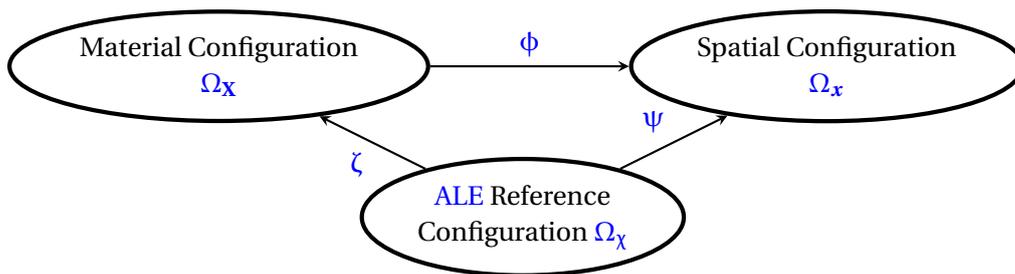


Figure 2.1.4: Kinematic Description of material flow with ALE

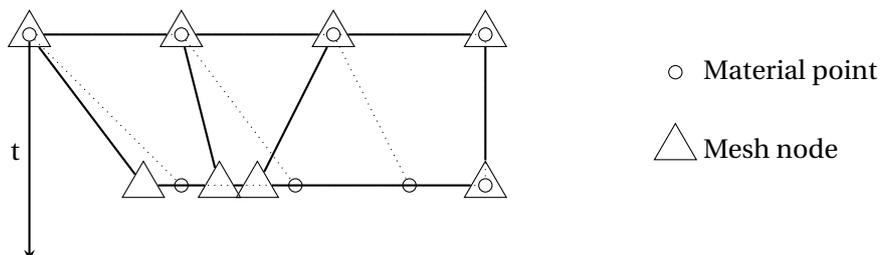


Figure 2.1.5: ALE Description

The ALE formulation is quite interesting from the fact that both material and mesh movement can be modeled. The evolution of the history-dependent variable f can be found with its material derivative. This derivative can be expressed in terms of the local derivative in terms of referential coordinates χ held fixed and a convective term taking into account the relative velocity \vec{c} between the material and the reference system.

$$\frac{\partial f}{\partial t}|_{\mathbf{x}} = \frac{\partial f}{\partial t}|_{\chi} + \vec{c} \cdot \nabla_{\mathbf{x}} f \quad (2.1.6)$$

The ALE formulation is well suited for problems where the region of interest moves in a fixed domain, such as crack propagation [31] or fluid-structure interaction [12]. In metal forming processes, it has been applied to processes like ring rolling [51], cutting operations [15], friction welding and mixing [17]. It has also been applied to the simulation of continuous processes such as three-dimensional profiling [8, 57], thermo-mechanical coupling with rolling mill [41] and polymer extrusion [16]. The idea can also be used to find directly the steady-state of the metal forming process [8, 36]. However, the ALE method still needs resolution of time-dependent equations which limits the speedup that can be achieved in comparison to the incremental resolution [44].

2.1.4 Reference frame kinematic description

A Reference Frame method was proposed by [5, 6] for the simulation of steady-state forming processes. This method aims to use a mapping technique based on a reference configuration Ω_r to eliminate the time-dependency in the equations. The basic principle behind the Reference frame method is that in the steady-state, the deformed configuration \mathbf{x} is an image of the undeformed configuration \mathbf{X}_0 at time t . In other words, the configuration \mathbf{X}_0 evolves to \mathbf{x} . Thus, a control volume based on the undeformed configuration. Next a reference configuration Ω_r identifies the material particles with coordinates \mathbf{r} and is mapped onto the undeformed configuration \mathbf{X}_0 at a given time instant t with the map φ as represented in the equation (2.1.7c).

$$\Omega_{\mathbf{x}} = \phi(\Omega_{\mathbf{X}_0}, t) \quad (2.1.7a)$$

$$\Omega_{\mathbf{X}_0} = \phi^{-1}(\Omega_{\mathbf{x}}, t) \quad (2.1.7b)$$

$$\Omega_{\mathbf{X}_0} = \varphi(\Omega_r, t) \quad (2.1.7c)$$

The length of the reference configuration is same as the control volume describing the undeformed configuration, and the undeformed configuration at any time instant t can be found with the following relation:

$$\varphi(\mathbf{r}, t) = \varphi(\mathbf{r}) + \vec{v}t \quad (2.1.8)$$

At the steady-state, the material points referenced by \mathbf{r} at different times would be the same. Hence all the fields in the reference configuration do not vary with time. The evolution of the history-dependent variables are thus computed in the reference configuration, which involves resolution of the following convection equation.

$$\frac{\partial f}{\partial t}(\mathbf{r}, t) = \frac{\partial f}{\partial \mathbf{r}}(\mathbf{r}, t) = 0 \quad (2.1.9)$$

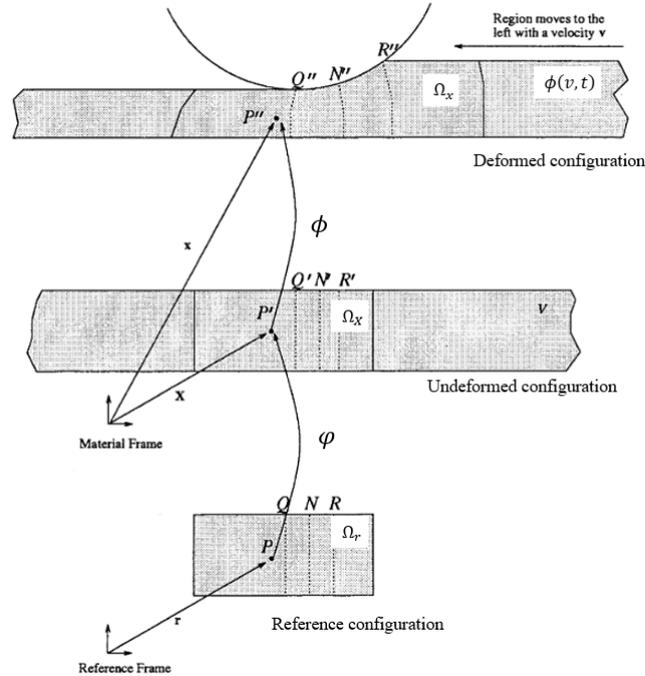


Figure 2.1.6: Mapping among undeformed, deformed and reference frames in the reference frame method by [5]

2.2 Classification of resolution methods for continuous forming processes

This section discusses the different strategies used for the numerical resolution of the continuous metal forming problem. The forming problem entails solving a multi-field problem to compute a set of unknowns like displacement \bar{u} /velocity \bar{v} , temperature T . A set of secondary unknowns like stress σ , strain ϵ , damage, etc. can describe the state of the material being formed. The said problem can be solved either with generalized incremental methods, if the user is interested in the both (transient and steady-state) the phases of the continuous forming problem. Otherwise, if the user is interested only in the steady-state phase, the non-incremental strategies can find a high fidelity solution with huge computation advantage. This section is focused on highlighting these strategies.

2.2.1 Incremental Resolution

The *Incremental Resolution* entails the resolution of the time-dependent thermo-mechanical multi-field problem in the Lagrangian frame with time integration. The said equations can be solved either with Explicit or Implicit time-integration. Even though the former is quite robust in terms of solving high frequency, transient problems, a very small time-step size is necessary for the stability of the numerical solution. On the other hand, an implicit solver, like [ForgeNxt[®]inc](#), makes possible to use large computation time-step. However, an iterative resolution of the equilibrium state is necessary. In addition, the time step size cannot be very large to keep the solution stable. The incremental formulation can be used to simulate the complete forming process including the transient engagement and release phases. The free surface computation and history-dependency are treated naturally [7] with these formulations. The UL method is used as large deformations involved. The deformation of the material is computed from the displacement field in a displacement-based formulation or the time integration of velocity in a velocity-based formulation. The history-dependent behavior can be accounted simply with the time integration of the state-variables. If the main interest of the problem is the steady-state of the process, the

simulation progresses until the time instant t when the shape and the state-variables within an imaginary control volume around the tool (loading region) become stationary. The incremental formulation caters to a wide spectrum of metal forming problems, and is quite robust. However, it is disadvantageous as a large number of time increments are required. If a multistand/multi-pass rolling process is to be simulated, the entire length of the workpiece must be modelled. Thus the computation may last for days/ weeks and requires an augmented storage due to the sizable model.

2.2.2 Non-incremental Resolution

Alternatively, if the objective of the simulation is just to find the steady-state behavior of the process or the product, the non-incremental methods give a significant computation advantage. These methods aim to compute directly the steady-state of the continuous forming problem. In comparison to the incremental methods, the formulation of these methods is rather difficult. One of the important aspects of the non-incremental formulation is that the computation domain is described only by an imaginary control volume (in Figure 1.4.1b) in which the steady-state is defined. However, finding the free surface boundary with non-incremental methods is by invoking an additional equation. Based on the chosen strategy/equation used for resolution of the free surface boundary, two different approaches are possible: (i) non-steady-state and (ii) steady-state.

2.2.2.1 Non steady-state approach

The non-steady-state approach solve the steady-state with the ALE approach. While solving the continuous forming problem, the thermo-mechanical equations are solved in the Lagrangian frame and the mesh is updated in the Eulerian frame with the time integration. It thus combines the advantages of Lagrangian (free-surface monitoring) and Eulerian (regular and fixed mesh) formulations while minimizing their defects [13]. Its decoupled formulation is the most used, mainly because it is very easy to implement in a Lagrangian code [57]. Two steps are added to update the geometry in the subsequent increment and transfer the fields to it. Generally, the global mesh is fixed during this step, and only the boundary surface is updated and hence the corresponding nodes are re-positioned. After the new surface has been computed, the volume mesh is regularized. This method brings strong stability and robustness [8, 36] in addition to reducing the computation times in comparison to the updated Lagrangian formulation. The shape computation with this method is quite similar to the incremental method (see Figure 2.2.1), so the speed-up achieved is not huge [44] in comparison to the steady-state approach.

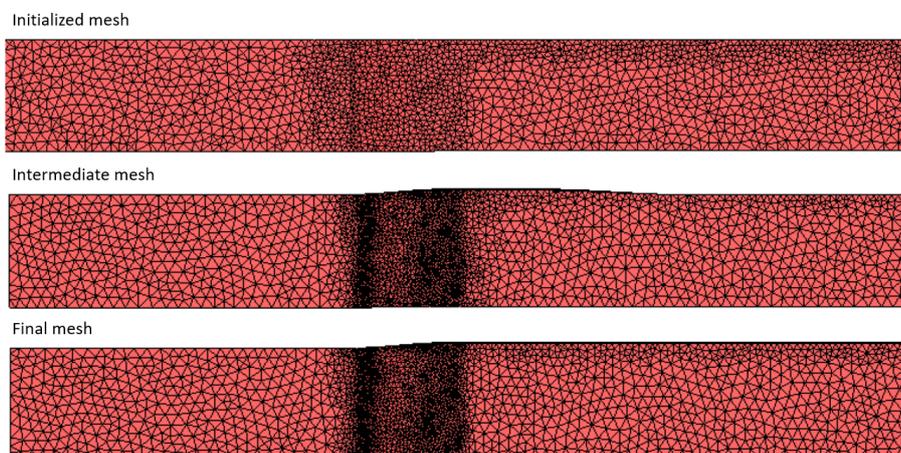


Figure 2.2.1: Computation of steady-state shape with ALE approach [41]

2.2.2.2 Steady-state approach

The steady-state approach aims at completely eliminating the time-dependency from the equations. This is done by introducing either a new reference mapping that moves along the control volume as used by [6] or by introducing an additional global equation that corrects the mesh coordinates to follow the flow streamlines. In the former, the movement of the reference frame along with the control volume eliminates the time-dependency from the equations. The latter employs a Eulerian formulation and the additional equation results in two principle unknowns of the strongly coupled mechanical problem: the velocity and the shape. Both these methods eliminate the time-dependency from the equations and are much faster than the traditional incremental or non-steady-state approaches. The system of equations can be solved either with direct or iterative methods.

Direct Resolution

The Direct resolution of the system of equations entails the resolution of all unknowns of the thermo-mechanical-shape problem simultaneously. This method has been applied both to the Balagangadhar Reference frame approach [5, 46] and the Eulerian elasto-visco-plastic formulation [42]. The coupled version was also developed to simultaneously determine the velocity field, pressure, and surface position and applied to the polymer extrusion process in the works of [14, 34]. The advantage of direct methods is that the complete multi-field problem is resolved at the same time. Hence, the equilibrium is automatically satisfied on the computed shape. The main bottleneck of the direct methods is that the stiffness matrix size is generally huge as the DoF are five times higher. This leads to several problems: a large number of iterations are required to reach a certain convergence threshold; the convection must be taken into account at each iteration, which is time-consuming; and a much larger system of equations needs to be solved, which requires huge storage.

Iterative Resolution

A decoupled, iterative approach is more practical for solving the strongly coupled problems. It has been studied for the Reference frame approach by [61] and the Eulerian approach by [44]. The iterative resolution method is more efficient since the geometry update and the convective increment are calculated only once per global iteration. The Eulerian formulation approach with iterative resolution has been widely used in the 1980s and 1990s [19, 27, 35, 53, 60]. The solution is obtained in a few iterations (order of magnitude of the ten). It should be noted that the refinement of the domain does not significantly increase the number of iterations of the algorithm contrary to an incremental approach (of ForgeNxt[®]inc) where the time-step (and thus the number of increments) must be adapted to the mesh size to keep the contact surface and the state-variables stabilized. This approach, therefore, brings an enormous reduction in the computation time. This iterative method of calculating the stationary state has been used for the simulation of 2D [39] and 3D [25, 38, 59] polymer extrusion. In these works, the contact zone chosen was intuitive, hence, the correction of the domain was focused only on the free-surface downstream to predict the swelling at the exit of the control volume. On the metal forming side, this approach was applied to two-dimensional sheet rolling [37] and three-dimensional [19, 32, 35], to shape rolling [24, 26, 27], with mandrel tube rolling [48, 53, 60] and coupled with the consideration of the deformation of the rolling stand [19]. [26] used this method to reduce the computation time of their Lagrangian formulation [28] by a factor of 10. A decoupled approach using a staggered iterative algorithm has also been adopted in ForgeNxt[®]2016. Each of the variables \vec{v} , σ , $\vec{\epsilon}$, \mathbf{x} are solved separately until equilibrium is satisfied, usually with fixed-point iteration method [44, 45].

2.3 Mathematical modeling of continuous forming

Once the kinematic description and the resolution method is chosen, the the process is described with the equilibrium conditions, the material behavior and the boundary conditions. This section is focused on the generalized description of these equations.

2.3.1 Conservation laws

The conservation laws are the linear momentum equilibrium and the mass conservation of the material. The mechanical equilibrium equation is derived from the Newton's second law of motion (material derivative of the total force) and described as following:

$$\vec{\nabla} \cdot \boldsymbol{\sigma} + \rho \vec{b} = 0 \quad \text{in} \quad \Omega_x \quad (2.3.1a)$$

$$\vec{\nabla}_0 \cdot \mathbf{P} + \rho_0 \vec{b}_0 = 0 \quad \text{in} \quad \Omega_x \quad (2.3.1b)$$

In continuous forming, the body force term \vec{b} and the acceleration term $\frac{d\vec{v}}{dt}$ in the equilibrium are considered to be negligible. The equation in the deformed configuration is represented with the equation (2.3.1a). In the current configuration Ω_x it is represented with the equation (2.3.1b)). It is recommended to refer to the definitions of each of the variables in the Glossary. In the equation (2.3.1b), \mathbf{P} is the First Piola-Kirchhoff stress tensor measure and is related to Cauchy's stress with the relation (2.3.2a). Here, \mathbf{F} is the deformation gradient tensor and J represents the Jacobian defined by the determinant of the deformation gradient tensor. \mathbf{P} is non-symmetric as it expresses the force in the current configuration in terms of area in the reference configuration. It is better replaced with the Second Piola-Kirchhoff stress tensor (defined in equation (2.3.2b)) which is symmetric.

$$\mathbf{P} = J \boldsymbol{\sigma} \cdot \mathbf{F}^{-T}, \quad \text{where } J = \det \mathbf{F} \quad (2.3.2a)$$

$$\mathbf{S} = J \mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \mathbf{F}^{-T} \quad (2.3.2b)$$

The mass equilibrium or the continuity equation (2.3.3a) is represented with the material derivative of the density ρ (in the current configuration). The partial time-derivative of the density becomes zero at steady-state, and the gradient of the density is negligible for viscoplastic materials, but not with elastoplastic materials. In the reference configuration, the density ρ_0 (2.3.3b) is used.

$$\frac{d\rho}{dt} = \frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{v}) = \frac{\partial \rho}{\partial t} + \rho \vec{\nabla} \cdot (\vec{v}) = 0 \quad (2.3.3a)$$

$$\rho J = \rho_0 \quad (2.3.3b)$$

At the onset of the steady-state, the classical rate independent equilibrium equation (2.3.1) is applicable and extensively used in literature [11, 19, 33, 46]. However, at the steady-state, the material derivative of the force is also nullified, which leads to a rate equilibrium equation and is also used by [42, 50, 55].

2.3.2 Material constitutive behavior

The material constitutive model in the mechanical description (in macro scale) commonly relates the stress tensor $\boldsymbol{\sigma}$ with the strain tensor $\boldsymbol{\epsilon}$. This relation is generally obtained by fitting the experimental testing data of material samples in a mathematical model. The mathematical model may have dependency on the deformation history, material hardening, creep, etc. With rate-dependent plasticity, the relation is rather established with the strain-rate $\dot{\boldsymbol{\epsilon}}$ tensor. A generalized constitutive relations can be described as following:

$$\boldsymbol{\sigma} = \bar{\psi}(\dot{\boldsymbol{\epsilon}}(\vec{v}), T) \quad (2.3.4a)$$

$$\dot{\boldsymbol{\sigma}} = \bar{\psi}(\boldsymbol{\sigma}, \dot{\boldsymbol{\epsilon}}(\vec{v}), T, t) \quad (2.3.4b)$$

The first equation (2.3.4a) relates the stress σ with the strain-rate $\dot{\epsilon}$ with the functional $\bar{\psi}$. Such a relation is used with viscoplastic behavior laws. The material behavior dependency on the rate of loading or the strain-rate is such that at higher rates, larger material strength is observed (as shown in Figure 2.3.1a). With higher temperature, the material becomes less rigid. The equation (2.3.4b), relates the stress-rate $\dot{\sigma}$ with the stress σ and strain-rate $\dot{\epsilon}$. This is what is termed as the time history-dependency, which defines the state of material at a time instant $t + \Delta t$ w.r.t. the state at a previously known time instant t . This model is representative of elasto-viscoplastic materials (as shown in Figure 2.3.1b).

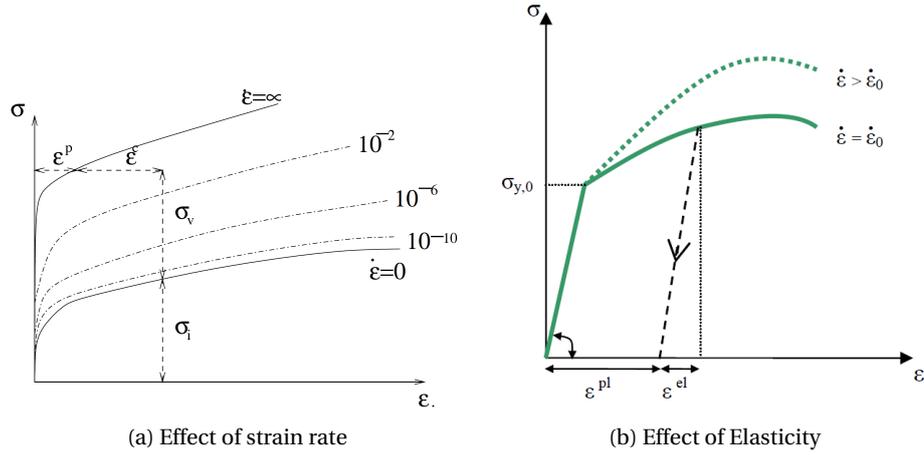


Figure 2.3.1: Modeling material behavior [2]

The elasticity depicts the recovery behavior of the material to go back to unloaded state after the load is removed. At the micro scale, elasticity is caused by the reversible elongation of the inter-atomic bonds under the applied stress of the material particles whereas the plasticity is caused by the slip between grains. This slip between grains becomes dominant at high temperature and low strain-rate (above $0.5 T_m$, expressed in Kelvin). It is ultra-dominant in superplasticity. More generally, be it for hot or cold deformation, the main mechanism of plasticity is the dislocation slip on slip systems (= crystallographic plane + easy sliding direction). In short, the elasticity does not account for any damage in material whereas plasticity leads to permanent damage.

2.3.3 Elasto-viscoplasticity in the context of steady-state formulations

Since one of the primary objectives of this thesis is to propose a resolution strategy for the steady-state resolution of cold forming processes with unstructured meshes, it is quintessential to investigate the history of elasto-viscoplasticity in the context of steady-state formulations.

Frame indifference of stress-rate (Objectivity)

Objectivity or frame indifference is an important concept in continuum mechanics as the material constitutive relation should be independent of reference frame and rigid body motion. If the stress and strain are material quantities, the objectivity is automatically satisfied, however, for spatial quantities, the objectivity of the stress rate must be ensured. The time derivative of the Cauchy's stress $\dot{\sigma}$ is non-objective. In rate-dependent constitutive equations, the objective rates are constructed with pull-back and push-forward procedures. The most commonly used objective stress rates are shown in equation (2.3.5). The Truesdell stress rate is found from the time derivative of the second Piola-Kirchhoff's stress tensor in equation (2.3.5a). If the pull-back and push-forward operations are performed with the spin tensor \mathbf{w} , the resulting objective stress rate is called the Jaumann stress-rate as shown in equation (2.3.5b).

$$\overset{\circ}{\sigma} = \dot{\sigma} - \mathbf{L} \cdot \sigma - \sigma \cdot \mathbf{L}^T \mathbf{w} + tr(\mathbf{L}) \sigma \quad (2.3.5a)$$

$$\overset{\vee}{\boldsymbol{\sigma}} = \dot{\boldsymbol{\sigma}} + \boldsymbol{\sigma} \boldsymbol{w} - \boldsymbol{w} \boldsymbol{\sigma} \quad (2.3.5b)$$

Classification of elastoplastic modeling

In elastoplasticity modeling, the primary aim is the decomposition of the two phenomena and model them separately with respective behavior models. There are two main approaches in the literature that use either additive or multiplicative decomposition of the elastic and plastic parts of strain-rate described in the equations (2.3.6a) and (2.3.6b) respectively.

$$\dot{\boldsymbol{\epsilon}} = \dot{\boldsymbol{\epsilon}}^e + \dot{\boldsymbol{\epsilon}}^p \quad (2.3.6a)$$

$$\mathbf{F} = \mathbf{F}^e \cdot \mathbf{F}^p \quad (2.3.6b)$$

The usage is dependent on the degree of elastic or plastic behavior in deformation behavior. For example, in the small deformation plasticity with Lagrangian description, the additive decomposition is used. The elastic part is modeled with Hooke's equation [23], and the plastic part with the plasticity flow rule [21]. But the Hooke's model is not valid for large (finite) deformation problems as in metal forming. Among the multiple approaches in the literature, the most famous approach used for solving such problems is the extension of small deformation elastoplasticity to large deformations. This approach by Prandtl-Reuss [21] proposed to decompose deformation-rate tensor $\dot{\boldsymbol{\epsilon}}$ to elastic $\dot{\boldsymbol{\epsilon}}^e$ and plastic $\dot{\boldsymbol{\epsilon}}^p$ parts similar to the strain in small deformation problems. The elastic behavior is modeled with Truesdell's hypoelasticity theory with equation (2.3.7a) that relates the Jaumann stress-rate $\overset{\vee}{\boldsymbol{\sigma}}$ with the elastic strain-rate $\dot{\boldsymbol{\epsilon}}^e$. The plastic part is described with the plastic flow potential (2.3.7b) and the evolution of plastic strain (2.3.7c) respectively.

$$\overset{\vee}{\boldsymbol{\sigma}} = \mathbb{C}^e : \dot{\boldsymbol{\epsilon}}^e = \mathbb{C}^e : (\dot{\boldsymbol{\epsilon}} - \dot{\boldsymbol{\epsilon}}^p) \quad (2.3.7a)$$

$$\bar{f}(\boldsymbol{\sigma}, t) = 0 \quad (2.3.7b)$$

$$\dot{\boldsymbol{\epsilon}}^p = \lambda^{pl} \frac{\partial \bar{g}}{\partial \boldsymbol{\sigma}} \quad (2.3.7c)$$

The hypoelasticity definition has shortcomings as the constitutive law can lead to the creation or dissipation of energy in closed-loop deformation cycle [56]. Hence, it is recommended to use this description only for problems where large elastic strains are not envisaged. The multiplicative decomposition [30] is useful with large deformation flow theories of plasticity.

$$\mathbf{L} = \dot{\mathbf{F}} \cdot \mathbf{F}^{-1} \quad (2.3.8a)$$

$$\mathbf{L} = \mathbf{L}^e + \mathbf{L}^p = \dot{\mathbf{F}}^e \cdot (\mathbf{F}^e)^{-1} + \mathbf{F}^e \cdot \dot{\mathbf{F}}^p \cdot (\mathbf{F}^p)^{-1} \cdot (\mathbf{F}^e)^{-1} \quad (2.3.8b)$$

The deformation gradient tensor \mathbf{F} is decomposed into elastic and plastic deformation gradients as represented in equation (2.3.8b). This method entails two successive mappings for the plastic and elastic deformations and introduces stress relaxed intermediate configurations around all material points that are geometrically incompatible. This deformation gradient is used to describe the respective elastic and plastic components of the velocity gradient \mathbf{L} . The elastic part is modeled with a hyperelastic description that can be used in general for hypoelasticity. The symmetric part of plastic velocity gradient tensor is called the plastic strain-rate. Even though the multiplicative decomposition of strain-rate tensor is more wholesome, the Prandtl-Reuss equations are a good compromise between precision and simplicity for modeling the constitutive behavior in the metal forming problems, in particular in small time-steps solutions of the Updated Lagrangian formulation.

Computation of state variables from evolution equations

One of the important aspects of the steady-state formulation is the resolution of history-dependent state-variables \mathbf{f} whose evolution is described with its material derivative. The evolution of time-history variable \mathbf{f} in the Eulerian frame can be described with the equation (2.1.4). In the said equation, the temporal part of the derivative is nullified when the steady-state is reached, so the evolution equation of the history variable is given as below.

$$\dot{\mathbf{f}} = \vec{v} \cdot \vec{\nabla}_x \mathbf{f} \quad (2.3.9)$$

For viscoplastic materials, the material consistency \mathbf{K} may depend on the equivalent strain $\bar{\epsilon}$, and hence brings in the history dependency in these models. For elastoplastic/ elasto-viscoplastic materials, the history dependency comes from the fact that the stress-rate $\dot{\boldsymbol{\sigma}}$ is dependent on the stress (as seen in equation (2.3.4b)). However, in comparison, the elastoplastic models results in a strongly coupled problem in velocity and stress and that the material stiffness is not continuous. For treating such equations, special measures are required to be taken. In general, the evolution equation can either be solved with (i) the global resolution [42, 45, 46] or (ii) the streamline integration [19]. The global resolution method is a generalized method applicable on any kind of mesh and is resolved with the Finite Element method. The problem domain is discretized and the weak form (2.3.10) is resolved to compute the unknown state-variable.

$$\int_{\Omega_x} N_k^{\text{SUPG}} (\dot{\mathbf{f}} - \vec{v} \cdot \vec{\nabla}_x \mathbf{f}) d\omega = 0 \quad (2.3.10)$$

The said equation is convection dominant, and hence it is recommended to use [Streamline Upwind Petrov Galerkin \(SUPG\)](#) finite elements [6] for the stabilization of the numerical solution. In elasto-viscoplasticity, the onset of plastic flow (2.3.7c) leads to a sudden change from a highly stiff elastic state to a much less stiff plastic state. While implementing with Finite element method, the tangent matrix must be defined to compute the stress. With elasto-viscoplasticity, the tangent becomes discontinuous. For keeping the tangent continuous, an internal variable \mathbf{t} continuous in both elastic and plastic state can be used to track the extent of material damage [43]. Another problem associated with the usage of global resolution is that the state-variable \mathbf{f} is P0 continuous, and its gradient $\vec{\nabla}_x \mathbf{f}$ in equation becomes non differentiable. Hence, for its resolution discontinuous Galerkin methods [4] can be applied. Otherwise, smoothed P1 state-variable can be computed by averaging the solution on the nodes. In general, the global resolution methods tend to have significant cross-wind diffusion and must also be considered while making the choice of the test function and are also susceptible to oscillations.

Alternatively, the history-variables can be integrated along the flow streamlines. The streamline integration methods eliminate the need of computation of tangent, and can compute the state-variable $\mathbf{f} + \Delta \mathbf{f}$ by solving simple algebraic equation (2.3.11) for the time increment Δt similar to incremental methods. However, the streamline methods, are generally applied only with structured meshes in which the mesh coordinates can be moved to align with the global streamlines of the material flow.

$$\Delta \mathbf{f} = \int_t^{t+\Delta t} \dot{\mathbf{f}} dt \quad (2.3.11)$$

The advantage of this formulation is that in incremental formulation, this time-step Δt is chosen such that it ensures stability of the contact surface. The time-step Δt is however inconsequential in steady-state formulations and must rather be modeled. One such manner to model the time is with a pseudo-time-step $\Delta \tilde{t}$ as proposed by [18] in his [Heterogeneous pseudo-time-step \(HPTS\)](#) approach:

$$\Delta \tilde{t} = \frac{x_{int} - x_{int-1}}{|\vec{v}_{int}|} \quad (2.3.12)$$

In the equation (2.3.12), the pseudo-time-step is computed from the Backward Eulerian method with the ratio of distance between consecutive integration points and velocities at the integration point int . For using this, the consecutive integration points, int and $int-1$, must however be aligned

on the same streamline. This is true only if the consecutive elements can be aligned, and hence limit the application of this idea with structured meshes. It can also be noted that the consecutive integration points on the same streamline may be at non-homogeneous distances, so the pseudo-time-step computed is heterogeneous. This approach was implemented in Lam3[®] software. The streamlines are first traced, mostly during the shape update/ mesh relocation step and then the state-variables are integrated on these streamlines using the equation (2.3.11). For the complete stability of the field, the number of iterations required is as many as the number of integration points on the streamline. For the structured mesh, the number of integration points is equivalent to the number of element sections in the principal material flow direction. [1] compared the global resolution method with the streamline integration method for integrating strains and concluded that the global resolution (with Galerkin function) method is faster than the streamline integration. In ForgeNxt[®] 2016, the resolution of equivalent strain $\bar{\epsilon}$ for viscoplastic materials is done with the global resolution method because of the unstructured mesh.

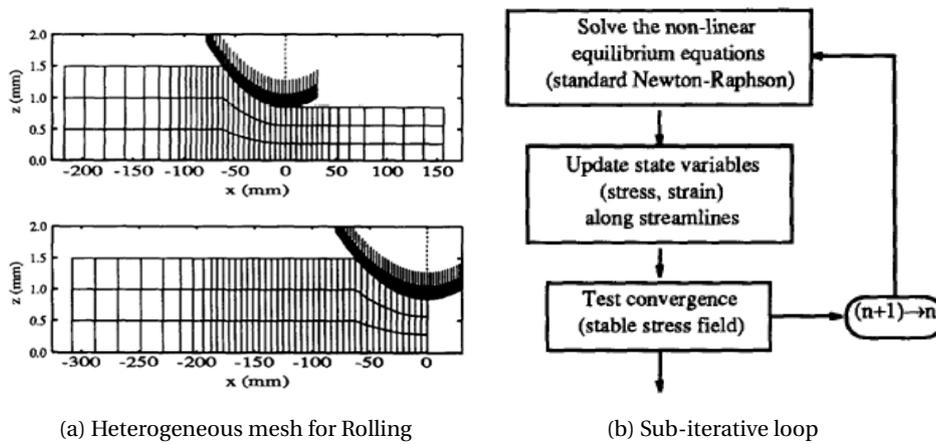


Figure 2.3.2: Resolution of history-dependent (elastoviscoplasticity) problems with [18] approach

For the resolution of elasto-viscoplasticity, an initialization of the material behavior is necessary, either by assuming a completely elastic or completely plastic behavior. With initial elastic assumption, the radial-return mapping algorithm [47] can be used to compute the elastoplastic stresses. [18] used Prandtl-Reuss equations for the description of elasto-viscoplasticity, due to the flexibility of usage with both time dependent incremental and time-independent steady-state formulations. [18] introduced a sub-iterative loop (see Figure 2.3.2b) for the stabilization of the stress field in the complete computation domain in the global fixed-point iteration. A comparison of results for the rolling problem with different material properties (viscoplastic, elastoviscoplastic, and elastoplastic) from [19] are presented in Table 2.3.1. The increase of resolution time from viscoplasticity to elastoplasticity models is clearly visible.

Material	Global Iterations	N-R iterations	CPU Time
VP	20	106	53 min
EVP (No yield limit)	10	527	180 min
EVP (with yield limit)	11	832	260 min
EP (with no viscous effect)	20	2135	570 min

Table 2.3.1: Comparison of resolution with VP, EVP and EP models [19]

2.3.4 Boundary conditions: Contact

Another important aspect of the mathematical modeling of the continuous forming problem is the computation of contact surface. The impenetrability condition is applied on the free surface boundary of the workpiece, such that it prohibits the material particle and spatial coordinates to

penetrate inside the tool. Hence, this boundary is an unknown of the problem and evolves continuously over time in the Lagrangian resolution. However, at steady-state conditions, the contact surface is stable, and unique. Neither the forces nor the displacements are prescribed at this surface. Rather a mix of two (Robin conditions) are applied with the [Karush–Kuhn–Tucker \(KKT\)](#) conditions [58]. The contact conditions introduce nonlinearity in the boundary value problem. In this section, a short summary of the most important notions and concepts of contact mechanics is presented.

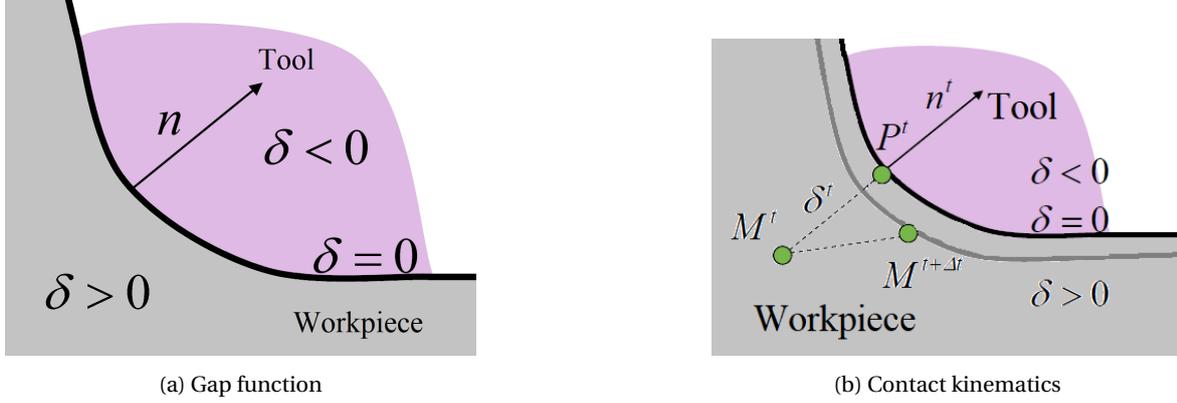


Figure 2.3.3: Contact Kinematics for Eulerian formulation

2.3.4.1 Contact Kinematics

A problem setup consisting of one single deformable body and a rigid obstacle is commonly referred to as *Signorini* contact, generally described with a Lagrangian formulation with the displacement field \vec{u} . Even with velocity-based formulations, like in [Transvalor S.A. ForgeNxt® \(Fg3®\)](#), the displacement field can be computed from the time integration of the velocity. The contact surface Γ_c is a subset of the free-surface boundary Γ of the workpiece. The distance between the free surface nodes of the workpiece and the surface segments of the tool are defined with the signed gap function δ . It is the normal distance measured from a material point M on the surface of the workpiece to the nearest point P on the surface of the tool.

$$\delta = \overrightarrow{MP} \cdot \vec{n}^{tool} \quad (2.3.13)$$

This description is based on the most intuitive choice in Contact Mechanics called the closest point projection (CPP). In the equation (2.3.13), \vec{n}^{tool} is called the unit tool normal, and defined on the workpiece surface towards the tool surface. The gap function δ is governed by the impenetrability condition (see equation (2.3.14a)). The material point $M^{t+\Delta t}$ describes the position of M^t moving with velocity \vec{v} , and the point $P^{t+\Delta t}$ on the tool represents the position of P^t moving with \vec{v}^{tool} as represented in equations (2.3.14b) and (2.3.14c).

$$\delta^{t+\Delta t} \geq 0 \quad (2.3.14a)$$

$$M^{t+\Delta t} = M^t + \vec{v} \Delta t \quad (2.3.14b)$$

$$P^{t+\Delta t} = P^t + \vec{v}^{tool} \Delta t \quad (2.3.14c)$$

The gap function $\delta^{t+\Delta t}$ can be expanded as in (2.3.15a). The second order expansion terms are neglected, and the time derivative of the gap function in the first order term is given with equations (2.3.15b).

$$\delta^{t+\Delta t} = \delta^t + \frac{d\delta^t}{dt} \Delta t + \underbrace{\frac{d^2\delta^t}{dt^2} \frac{\Delta t^2}{2}}_{O(\Delta t^2)} \quad (2.3.15a)$$

$$\frac{d\delta}{dt} = \frac{d}{dt} (\vec{M}\vec{P} \cdot \vec{n}^{tool}) = \left(\frac{d\vec{O}\vec{P}}{dt} - \frac{d\vec{O}\vec{M}}{dt} \right) \cdot \vec{n}^{tool} + \vec{M}\vec{P} \cdot \frac{d}{dt} (\vec{n}^{tool}) = (\vec{v}^{tool} - \vec{v}) \cdot \vec{n}^{tool} + \vec{M}\vec{P} \cdot \frac{d\vec{n}^{tool}}{dt} \quad (2.3.15b)$$

The terms described with the time derivative of position vectors of points M^t and P^t are replaced by the respective material \vec{v} and tool velocities \vec{v}^{tool} found from equations (2.3.14b) and (2.3.14c). In the discretized form, the time-derivative of the tool normal $d\vec{n}^{tool}/dt = 0$ for a very small time-step Δt , as the \vec{n}^{tool} is assumed to remain unchanged, and is recomputed on the updated geometry at the beginning of each time increment. Hence, for simplification, this term is eliminated in the subsequent development. Using the impenetrability condition (2.3.14a), its expansion and the derived terms in (2.3.15), the contact constraint $h(\vec{u})$ is determined as a function of displacement in the following equation. The displacement is computed as the time integration of the velocity $\vec{u} = \vec{v}\Delta t$.

$$\Rightarrow \delta^{t+\Delta t} = \delta^t + (\vec{v}^{tool} - \vec{v}) \cdot \vec{n}^{tool} \Delta t \text{ or } h(\vec{u}) = (\vec{u} - \vec{u}^{tool}) \cdot \vec{n}^{tool} - \delta \leq 0 \quad (2.3.16)$$

The *Hertz–Signorini–Moreau* (HSM) conditions or the Signorini conditions in the normal direction as shown below (2.3.17). They not only define a non-smooth and nonlinear contact law, but also suggest that the displacement is multi-valued at $\delta = 0$.

$$\begin{cases} h(\vec{u}) = (\vec{u} - \vec{u}^{tool}) \cdot \vec{n}^{tool} - \delta^t \leq 0 \\ \sigma_n \leq 0 \\ h(\vec{u}) \sigma_n = 0 \end{cases} \quad (2.3.17)$$

The second condition implies that no adhesive stresses are allowed in the contact zone. The contact normal stress (σ_n) is related to the local stress tensor σ with the following relation:

$$\sigma_n = (\sigma \cdot \vec{n}) \cdot \vec{n} \quad (2.3.18)$$

Finally, the third Signorini condition, better known as the complementarity condition, forces the gap to be closed when non-zero contact pressure occurs (material in contact) and the contact pressure to be zero when the gap is open (no contact). Note, that the *Signorini* conditions are equivalent to the classical *KKT* conditions that also arise in many other classes of problems such as constrained optimization, and thus standard solution techniques (e.g. based on Lagrange multiplier methods and active set strategies) from optimization theory can readily be adapted for contact mechanics.

Methods of enforcement

Different methods are possible to enforce the contact constraint (2.3.17) in the mechanical resolution. It requires to first invoke the virtual work principle on a domain Ω_x subjected under a load at the contact in the direction of the normal. We describe the functional Φ^e in equation (2.3.19) such that left side part comes from the internal potential energy (Φ) due to the deformation of the workpiece, and the right side part from the work done at the contact surface. For simplicity, the contact is considered to be bilateral and frictionless.

$$\forall, u^* \quad \left\langle \frac{\partial \Phi^e}{\partial u}, \vec{u}^* \right\rangle = \int_{\Omega_x} \vec{\nabla} \cdot \sigma \cdot \vec{u}^* d\omega = - \underbrace{\int_{\Omega_x} \sigma : \epsilon^* d\omega}_{\Phi} + \int_{\Gamma_c} (\sigma_n \vec{u}^* \cdot \vec{n}) ds = 0 \quad (2.3.19)$$

The minimizing of Φ with respect to the displacements \vec{u} leads to a solution that satisfies the equilibrium equation satisfying the contact boundary condition. The contact stress σ_n that ensures the impenetrability of contact nodes is determined. There are different manners the contact constraint can be applied.

Lagrange multiplier method

In the Lagrange multiplier method, the contact constraint is enforced by introducing an additional vector of unknown Lagrange multipliers λ into the mechanical system. The global size of this vector is equal to the number of constraints applied. Adding the Lagrange multiplier contributions to the underlying energy potential Φ results in the following extended overall potential, keeping in mind the behavior of $h(\bar{u}) \leq 0$ and also that the Lagrangian multiplier $\lambda \leq 0$. This means that at the contact surface the Lagrangian potential (right hand term) is positive.

$$L(\bar{u}, \lambda) = \Phi - \int_{\Gamma_c} h(\bar{u}) \lambda ds \quad (2.3.20)$$

Here, the contact force λ is equivalent to the contact stress σ_n described in the equation (2.3.19). The solution of the problem with the Lagrange multiplier is defined with the saddle point solution of the Lagrangian functional.

$$L(\bar{u}, \lambda) = \text{MAX}_{\lambda} \text{MIN}_{\bar{u}'} L(\bar{u}', \lambda) \quad (2.3.21)$$

Total variation of equation (2.3.21) leads to a mixed variational formulation with the stationary point to be determined by:

$$\forall, \bar{u}^* \quad \left\langle \frac{\partial L(\bar{u}, \lambda)}{\partial u}, \bar{u}^* \right\rangle = \int_{\Omega_x} \boldsymbol{\sigma} : \boldsymbol{\epsilon}^* d\omega - \int_{\Gamma_c} \lambda \bar{u}_n^* ds = 0 \quad (2.3.22a)$$

$$\forall, \lambda^* \quad \left\langle \frac{\partial L(\bar{u}, \lambda)}{\partial \lambda}, \lambda^* \right\rangle = - \int_{\Gamma_c} h(\bar{u}) \lambda^* ds = 0 \quad (2.3.22b)$$

The resultant mixed formulation of (\bar{u}, λ) is discretized with Finite Elements to retrieve the following system of equations.

$$\mathbf{A}\bar{\mathbf{U}} - \mathbf{B}^T \boldsymbol{\Lambda} = \bar{\mathbf{F}} \quad (2.3.23a)$$

$$-\mathbf{B}\bar{\mathbf{U}} = \bar{\mathbf{C}} \quad (2.3.23b)$$

While the Lagrange multiplier method allows for an exact satisfaction of the given constraints, on the other hand, it entails an undesirable increase in global system size. Even more severe is the fact that the saddle point structure of the system matrix usually poses difficulties in state-of-the-art iterative linear solvers as the positive definiteness of the stiffness matrix is not ensured.

Penalty method

The penalty method is another very widely used approach for constraint enforcement. The basic idea behind this method is to remove the constraints explicitly from the system and introduce a penalization of any constraint violation instead. This is typically reflected in an additional penalty potential contribution, which together with the underlying elastic potential Φ forms the following extended overall potential:

$$\Phi_p = \Phi + \frac{1}{2} \rho_c (h(\bar{u}))^2 \quad (2.3.24)$$

where ρ_c is the user-defined and problem-specific penalty parameter. The functional Φ_p does not have additional unknowns, as the case with the Lagrange multiplier method, but rather represents a problem of constrained optimization. The variation of equation (2.3.24) leads to a purely \bar{u} -based formulation here. Simply speaking, the penalization term works such as it associates large energies to \bar{u} solution violating the constraint $h(\bar{u}) = 0$. If the underlying potential Φ is convex, the new potential Φ_p retains its convexity. Thus, the penalty system to be solved is usually positive definite and bears large similarities with the original non-penalized system. This is a very desirable property with regard to the unmodified reuse of certain numerical techniques. The new potential is differentiable and the derivative is continuous.

$$\frac{d\Phi_p}{du} = \frac{d\Phi(\bar{u})}{du} + \rho_c h(\bar{u}) h'(\bar{u}) \quad (2.3.25)$$

However, penalty methods suffer from one serious drawback, viz. their dependency on the penalty parameter ρ_c . It can easily be found that the exact solution obtained with a Lagrange multiplier method is only recovered for the limit case $\rho_c \rightarrow \infty$. The Lagrangian multiplier at this state can be determined as $\lambda = -\rho_c h(\bar{\mathbf{u}})$. This situation inevitably generates a dilemma: it is not possible to choose very large penalty parameters because the resulting systems become more and more ill-conditioned. On the other hand, the finite values of the penalty parameter always allow for a certain violation of the given constraints, which might become unacceptably large. In addition, with ρ_c being problem-specific, no universally valid rule can be established for its determination. Nevertheless, penalty methods are widely used in practice, mostly due to their easy and efficient implementation.

Augmented-Lagrangian method

Simply speaking, the idea of the Augmented Lagrange approach is to find an optimal compromise between Lagrange multiplier and penalty methods. This allows an exact enforcement of constraints in combination with a penalty-like regularization for easier numerical treatment. Similar to Lagrange multiplier methods, the Augmented-Lagrangian method introduces a vector of Lagrangian multipliers λ . The Augmented-Lagrangian functional is written as:

$$L_\rho(\bar{\mathbf{U}}, \Lambda) = \Phi - \int_{\Gamma_c} h(\bar{\mathbf{u}}) \lambda ds + \frac{1}{2} \rho_c [h(\bar{\mathbf{u}})]^2 \quad (2.3.26)$$

This equation can be examined in more detail, revealing that the penalty term vanishes in the case of an exact satisfaction of the constraints $h(\bar{\mathbf{u}}) = 0$. Thus, the Augmented Lagrange formulation reduces to its Lagrange multiplier counterpart in the limit case, meaning that it has exactly the same solution as the Lagrange multiplier method. Nevertheless, this method comes with the important advantage that it allows for an easier numerical treatment than a pure Lagrange multiplier method. Constraint violation is penalized like for penalty methods, but this is achieved without changing the solution, even for finite values of the penalty parameter ρ_c . The well-known *Uzawa algorithm* is commonly used as an alternative for solving equation (2.3.26). The Lagrange multipliers are assumed to be given quantities and thus they can be removed as additional unknowns and the system is solved for $\bar{\mathbf{u}}$ only. This naturally defines an iterative procedure, i.e. an augmentation loop with iteration index i , for approaching the exact Lagrange multiplier solution. Therein, the penalty approach is used as kernel and the Lagrange multipliers $\lambda^{(i)}$ are fixed within each iteration step. Usually, the procedure starts with an initial guess $\lambda^{(i=0)} = 0$, making the first augmentation step identical to the ordinary penalty method. An update of the Lagrange multipliers is then obtained from the following equation:

$$\lambda^{(i+1)} = \lambda^{(i)} - \rho_c h^{(i+1)}(\bar{\mathbf{u}}) \quad (2.3.27)$$

The outer iteration on the Lagrange multiplier λ is repeated until a user-defined convergence criterion, usually monitoring the constraint $h^{(i)}(\bar{\mathbf{u}}) = 0$ in a suitable norm, is met.

2.3.4.2 Frictional Contact

The real contact behavior in an ideal contact surface is determined by the frictional response to tangential loading. For modeling the friction in contact, the relation of tangential stress ($\bar{\boldsymbol{\tau}}_f$) with the tangential slip ($\Delta \bar{\mathbf{u}}_t$) must be described. This relation computes the tangential stress at which slip occurs between the two surfaces. The *Tresca model* defines a linear relation with a scalar quantity g_f such that:

$$\bar{\boldsymbol{\tau}}_f = -g_f \frac{\Delta \bar{\mathbf{u}}_t}{\|\Delta \bar{\mathbf{u}}_t\|} \quad \Delta \bar{\mathbf{u}}_t = \Delta \bar{\mathbf{u}} - (\Delta \bar{\mathbf{u}} \cdot \bar{\mathbf{n}}) \bar{\mathbf{n}} \quad (2.3.28)$$

The *Coulomb model* permits to define the limit tangential stress $\bar{\boldsymbol{\tau}}_f$ dependent on the normal contact pressure σ_n .

$$\bar{\boldsymbol{\tau}}_f = -\mu_f \sigma_n \frac{\Delta \bar{\mathbf{u}}_t}{\|\Delta \bar{\mathbf{u}}_t\|} \quad (2.3.29)$$

Another way to model the friction is with a *viscoplastic model* which is common with the problems of hot forming.

$$\vec{\tau}_f = -\alpha_f K_f (T, \bar{\epsilon}) \|\Delta \vec{u}_t\|^{q-1} \Delta \vec{u}_t \quad (2.3.30)$$

2.3.5 Free-surface computation

The free-surface computation problems are frequently encountered in fluid mechanics, especially to describe the interface between two different phases (e.g. water and air). A framework for continuously tracking the free-surface is important to determine the interphase boundary. There are two main categories for modeling interfaces in Eulerian grids: (i) Interface/ Front tracking (ii) Interface capturing. The front tracking methods treat the free-surface as a sharp interface that is continuously tracked [52]. In this method, boundary fitted grids are used which are continuously updated. These methods advect a known interface from an initial configuration and are used in the applications where the interface is not expected to break or merge. The interface capturing methods are more diverse and do not treat the free-surface as a sharp interface. Some examples of this method are Marker-and-cell methods [20] which associate a mass-less particle to the free-surface which is followed with time. The volume of fluid method [22] is also very common, which as the name suggests tracks the volume of each phase in the grid/ element. Other possibilities generally used for interface capturing are the Level-Set methods [40] which use a signed distance function to track the interface. The steady-state processes, the shape of the product is determined with the free surface boundary. The free surface exhibits a physical phenomena [10], described as the lateral spread in flat rolling, groove filling in shape rolling, chip size in metal cutting and die swell in case of polymer extrusion (see Figure 2.3.4). A precise determination of the free surface is imperative to understand if the product would conform to the designed shape.

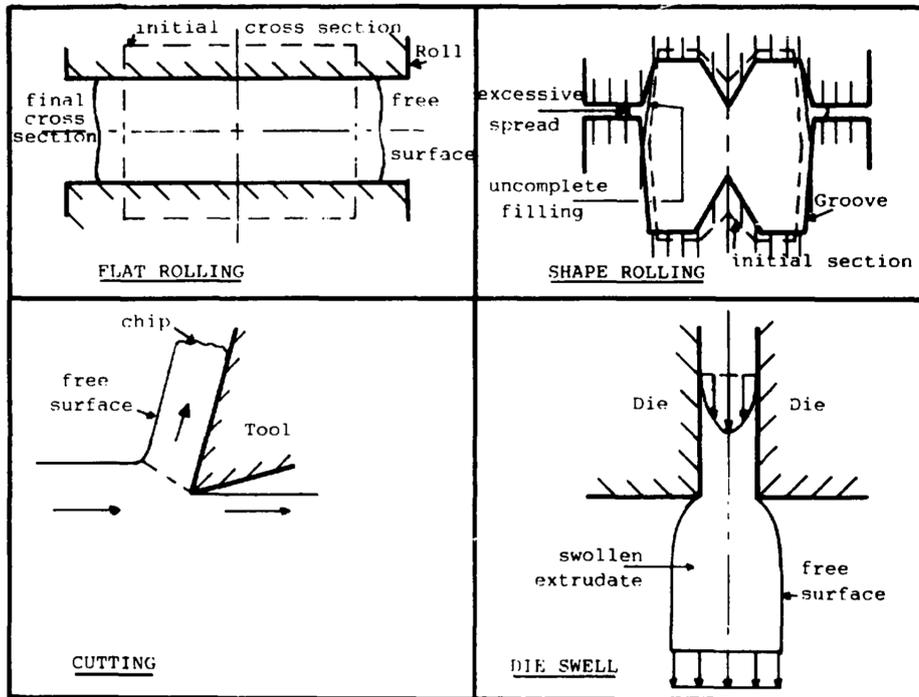


Figure 2.3.4: Free surface phenomenon in different processes [10]

The need for free-surface computation with the Eulerian description is to describe a sharp definition of the free surface boundary. Following methods have been used in literature for the adjustment of free-surface of the control volume:

2.3.5.1 Streamline method

$$\frac{dx}{v_x} = \frac{dy}{v_y} = \frac{dz}{v_z} \quad \Rightarrow \text{for } v_x = 1, \quad v_y = \frac{dy}{dx}; \quad v_z = \frac{dz}{dx} \quad (2.3.31a)$$

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} (v_y dx) \quad z_{i+1} = z_i + \int_{x_i}^{x_{i+1}} (v_z dx) \quad (2.3.31b)$$

As the definition of streamline suggests, the displacements of particles are in the direction of the flow, and in the steady-state, the pathline of the particles define the streamline. The coordinates of the free-surface thus must be aligned with the streamline to apply the free surface conditions. One manner to apply is with the streamline condition (2.3.31a). With the implementation of the streamline method, one can compute the new coordinates of a node in the directions normal to the rolling direction (\mathbf{x}) by integrating equation (2.3.31b) as seen in Figure 2.3.5a. In the figure, the velocity vector is assumed to be constant on the element

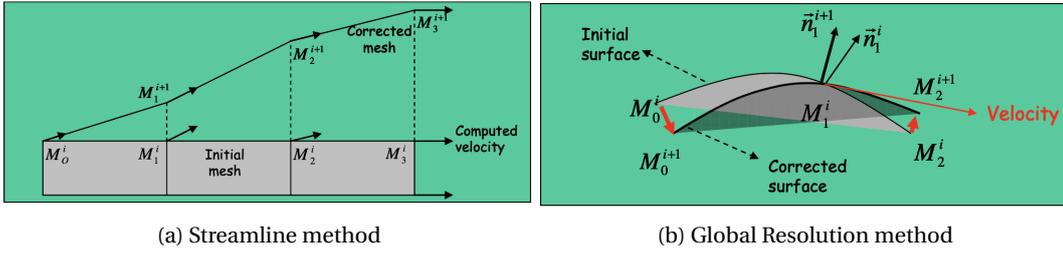


Figure 2.3.5: Resolution methods for free surface computation [44] with the assumption of constant velocity vector per element

The resolution with this method is extremely fast because there are no finite element equations to be solved. However, to describe the streamlines it becomes imperative to have a structured mesh as the consecutive nodes and the integration points are aligned along the streamlines. These streamlines connect the section at the input plane of the control volume to its output plane. The number of streamlines is therefore equal to the number of surface nodes on the input plane. The ease of usage of this method makes it interesting to the applications of polymer extrusion [25, 59], sheet rolling [19, 37], tube rolling [48, 60] or shape rolling [26, 27]. The geometry of the free-surface of the domain of the workpiece, in particular the zone of pre-deformation and the elastic retraction, is an unknown a priori. It is determined iteratively, as a function of the velocity field computed from the virtual power principle and the contact surface with the tool. In the case of unstructured meshes, this alignment of consecutive nodes/integration points is not possible. Hence, streamline methods are seldom used with unstructured meshes. The main drawbacks of the streamline method are its requirement for the structured mesh and that it is not adaptable with parallelization and mesh partitioning, which is common with Finite element solvers.

2.3.5.2 Global Resolution method

A more general method, compatible with Finite element solvers, is the global resolution methods. It is adoptable both with structured or unstructured meshes. [3] proposed the global resolution method that imposes a free-surface condition (2.3.32) on the boundary of the material. This condition imposes a zero material flux across the free-surface boundary (see Figure 2.3.5b). The right hand term $\vec{n}(\mathbf{x})$ is the unit normal defined on the surface material mesh surface.

$$\vec{v} \cdot \vec{n}(\mathbf{x}) = 0 \quad \text{on} \quad \Gamma \quad (2.3.32)$$

In the said equation \mathbf{x} represents the spatial coordinates of the free surface boundary of the workpiece being formed. The equation (2.3.32) represents the strong form of the free surface condition, which can be solved with Finite element method by invoking the weighted residual form or the weak form of the equation. In this form, the equation computes the corrected coordinates which

are described with the linear trial functions. However, for the test functions there are a multitude of possibilities as the free surface condition is a convection dominant equation.

Galerkin formulation

[32] used standard Galerkin test function N_k for the weighted residual form of the free-surface equation.

$$S_k = \int_{\Gamma} N_k (\vec{v} \cdot \vec{n}(\mathbf{x})) d\Gamma \quad (2.3.33)$$

This test function can be used as long as the slope of the free surface boundary is not very large. Hence the method works for simple quadratic shapes but with complex shapes with large curvatures, the Galerkin method leads to numerical oscillations [45].

Least Square formulation

For convection dominant problems (odd ordered partial differential equations), the **Petrov-Galerkin (PG)** functions are recommended as test functions. These functions provide stability to the numerical solution and deters oscillations observed with the Galerkin shape functions. In practice, these functions give larger weightage to the upwind side contributions to the finite element residual. In the large class of such elements, the most commonly used is the Least-Square function, which is also called the optimal control method, was used by [24] for the resolution of free-surface problem. The Least-square functional Φ_{LS} suppresses the numerical oscillations seen with the Galerkin method. In addition, the resultant function is a symmetric and constant about node k . The optimal control method aims to minimize the following free-surface functional Φ_{LS} :

$$\text{Min } (\Phi_{LS}), \quad \Phi_{LS} = \frac{1}{2} \int_{\Gamma} (\vec{v} \cdot \vec{n}(\mathbf{x}))^2 d\Gamma \quad (2.3.34)$$

The advantage of the optimal control method is that it is easily adaptable to the multi-DoF free surface problems like free surface boundary with edges. This needs to minimize the free surface residual in two linearly independent directions, which is natural with the Least-Square function. This method has been successfully applied to compute the free surface in the processes like polymer extrusion [10], shape rolling [24], and tube rolling [54], but is found to take a much longer time for resolution in comparison to the streamline method. Although the free-surface computation time is much less in comparison to the total resolution time for each fixed point iteration, because the mechanical resolution takes the larger part of the computation time.

SUPG formulation

Alternatively, the **SUPG** shape function N_k^{SUPG} [9] is another **PG** function frequently used for Finite element resolution of convection dominant problems:

$$N_k^{\text{SUPG}} = N_k + \alpha C_k, \quad C_k = \frac{h_e}{\vec{v}_e} \vec{\nabla} N_k \cdot \vec{v}, \quad \alpha = \frac{1}{2} \quad (2.3.35)$$

The right hand side of **SUPG** function in equation (2.3.35) provides a shift in the weights derived from Galerkin hat function N_k . in 1D, there is only one upwind and one downwind side element, hence, the shift is constant. On the other hand in 2D, this shift is proportional to the angular location of the element about node k . In the formulation of the **SUPG** function, C_k is the cosine of angle between the gradient of shape function $\vec{\nabla} N_k$ at node k and the averaged velocity \vec{v} on the element. The cosine C_k is between $[-1, 1]$ and differentiates the elements that lie on the upwind $[-1, 0)$ or downwind $(0, 1]$ of the node k . The h_e, \vec{v}_e are the characteristic length and velocity of the facet element e . An important factor that contributes to the magnitude of shift is the stabilization constant α . The non-dimensional parameter α generally takes positive values between $\in (0, 1]$. The standard value for α used in [29, 45] is $\frac{1}{2}$.

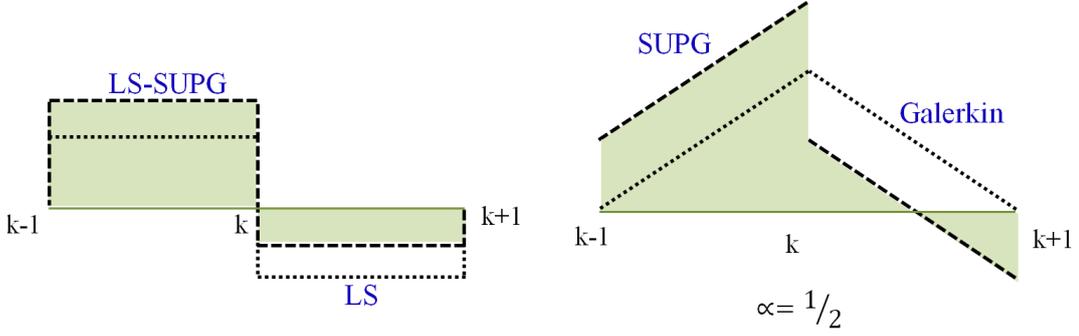


Figure 2.3.6: Petrov Galerkin test functions

Even though the **SUPG** formulation results in much better accuracy and speed for free surface problems, it is difficult to adapt for multi-**DoF** problems. A scalar can be used with mesh regularization to find free surface with the same scalar equation, but lacks robustness and takes really long time for convergence. On the other hand, a linear independence of free surface directions can be assumed to compute residual in each of these directions. This idea proves to be very robust and reduces the free surface computation time significantly with much better accuracy.

New **PG** formulations

Both Least-Square and **SUPG** functions are types of broader Petrov Galerkin functions. The former is advantageous as it can be easily adapted to multidimensional free-surface correction problems, whereas the latter is preferable for its much better accuracy in predicting the free-surface boundary and ability to modify the magnitude of shift is the stabilization constant α . [45] proposed a modified **Least-square Streamline Upwind Petrov-Galerkin (LS-SUPG)** (Figure 2.3.6) for convection dominant problems which adopts the properties from both Least Square and SUPG functions.

$$N_k^{\text{LS-SUPG}} = (1 + \alpha C_k) \left(\vec{v} \cdot \frac{\partial \vec{n}}{\partial t_k} \right) \quad (2.3.36)$$

2.3.5.3 Transport method

[53] proposed to decompose the velocity into its tangential and normal components to solve the steady-state shape. The resultant equation (2.3.37) is solved to compute the unknown shape \mathbf{x} using Galerkin finite element method. However, this equation is convection dominant, so [49] proposed to use stabilized Finite elements with SUPG test function [9].

$$\vec{v}_t \vec{\nabla} \mathbf{x} = \vec{v}_n \quad (2.3.37)$$

2.3.6 Contact treatment with free surface computation

As it can be understood that the surface flow is tangential both on the contact boundary and the free surface boundary. The free surface boundary either conforms to a streamline [53] or a free surface condition [44] as seen in the previous section. The contact boundary serves an impenetrability condition, which pushes and projects any node penetrating inside the contact with the tool onto the contact surface. Following are the main challenges and observations with the treatment of contact with steady-state formulations:

1. No matter the method chosen for finding the free surface boundary, it results in oscillations at the interface between the free surface boundary and the contact [44] as seen in Figure 2.3.7a. The reason being that the nature of the two treatments is completely different, as a Dirichlet kind of constraint is applied on a free surface flow as shown in Figure 2.3.7b. [29] suggested suppressing these oscillations with the removal of contribution of elements containing contact nodes from the free-surface computation residual.

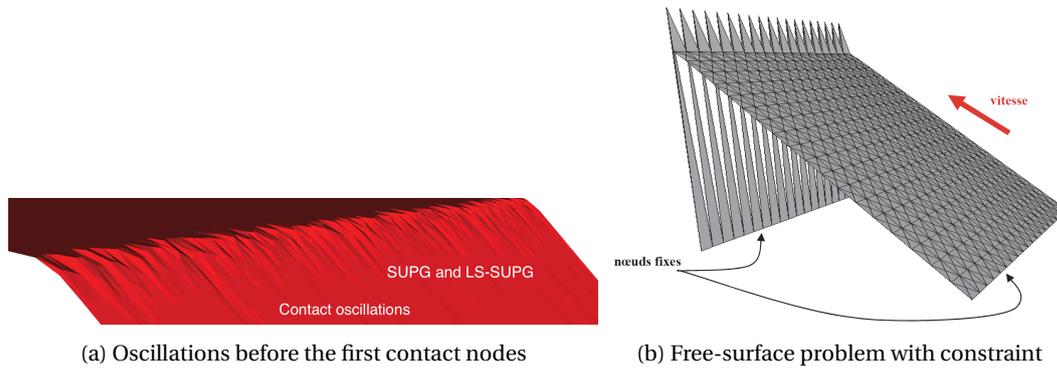


Figure 2.3.7: Problem with contact with global resolution [44]

2. The free surface and the contact surface may be computed either in the same step or in different steps. [60] proposed not to project the contact nodes during the free-surface computation and to add another stage for the nodes projection (see 2.3.8). In the following step, the nodes which are in compression and the ones that penetrate the tools are projected on the surface of the tools. However, the two boundaries are or may not be computed at the same time as both lead to several problems and thus make the contact treatment as one of the most recognized challenges of the steady-state formulations. The oscillations at the interface are suppressed using smoothing methods.

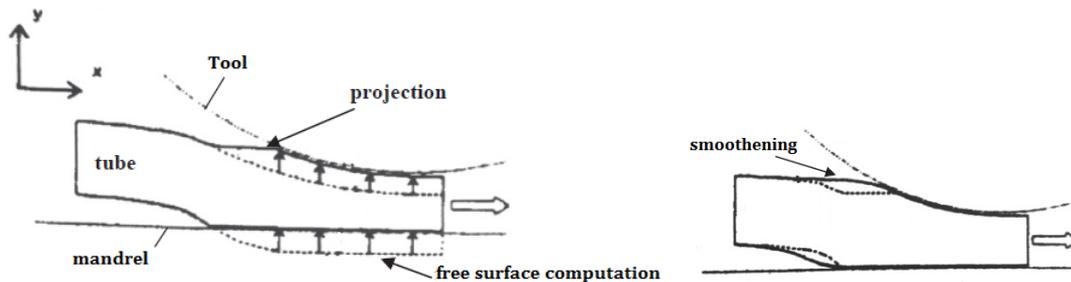


Figure 2.3.8: Correction of free-surface [60]

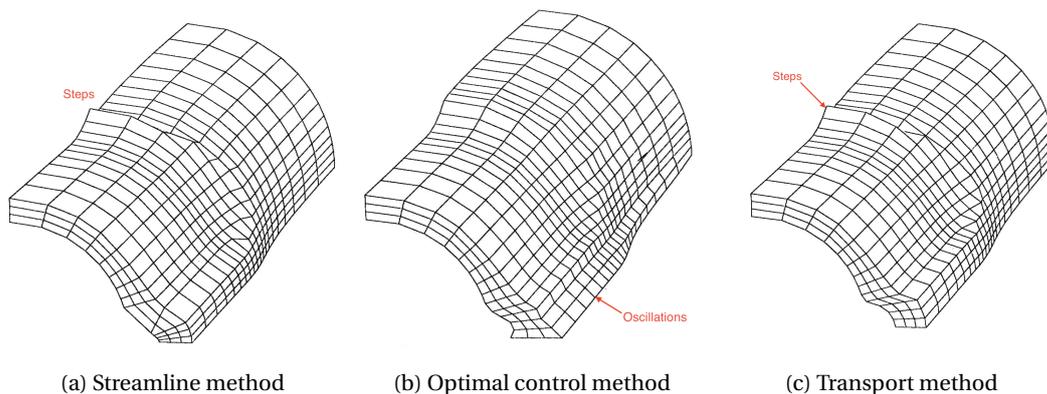


Figure 2.3.9: Different free-surface computation methods tested by [53]

3. [53] studied the resolution of the two surfaces with the *streamline method*, but found that the contact conditions are not respected as the iterations progress and a loss of contact is observed. If forced to remain in contact, "steps" appear (see Figure 2.3.9a) in the upstream of the contact zone and can hinder the further calculation. Smoothing is applied to suppress these discontinuities. [48] introduced a new expression for the smoothing of free-surfaces

after the projection step. It is applied as a function of the position x of a node on the streamline:

$$dy = \left[\frac{(x - x_0)}{(x_e - x_0)} \right]^a \Delta y_e \quad dz = \left[\frac{(x - x_0)}{(x_e - x_0)} \right]^a \Delta z_e \quad (2.3.38)$$

In the equation above, x_0 is the coordinate at the entry plane, x_e is the coordinate of the first node in contact with the tools, x is the current node coordinate, and $\Delta y_e, \Delta z_e$ are the lateral displacements of the first node in contact and a is a parameter that influences the smoothing correction. With $a = 1$, we get strong correction on the complete free-surface. From [48], the value of $1 \leq a \leq 4$. In addition, [53] also studied the free-surface problem with the *optimal control method* (see Figure 2.3.9b) and the *transport method* (see Figure 2.3.9b) and found that they all converge towards the same solution. The transport method also results in steps formation before the first contact point similar to the streamline method. Indeed, the two methods propagate the surface nodes in a similar manner following the flow. The optimal control method does not reveal this problem, however, oscillations were observed at the lateral free-surface.

4. For describing the contact surface, it is important to determine the first and the last contact coordinates with the tool. The algorithms developed by [18, 27] were focussed on the determination of these points (with the tool) on the free surface boundary streamlines. The last point is generally obtained from the inversion of normal stress sign (see Figure 2.3.10) and the first point is found with certain geometric criteria.

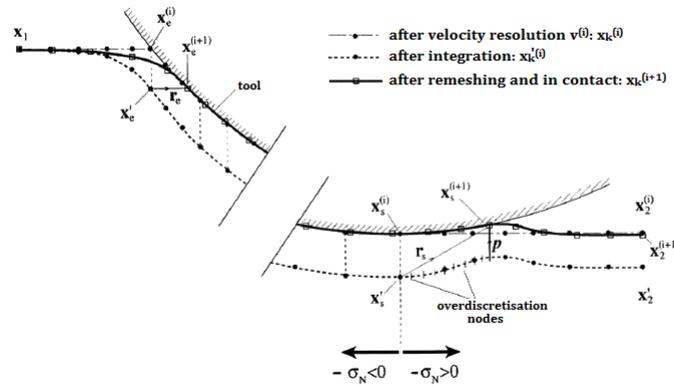


Figure 2.3.10: Correction of free-surface [18]

5. Using a unilateral contact constraint leads to a significant overestimation of the lateral spread. It is therefore advisable to fix the nodes in contact during the calculation of free-surface [53]. [24] also used the *global resolution* with the *optimal control method* using a bilateral contact (fixed nodes). This method was tested with complex geometries [10] and it was observed the lateral spread that allows the material to flow into the grooves of the tools is suppressed (see Figure 2.3.11). If allowed, it causes the degeneration of mesh. Hence, it was proposed to compute the free surface initially with the mesh normal and then project it on the tool surface in another step using the tool normal.

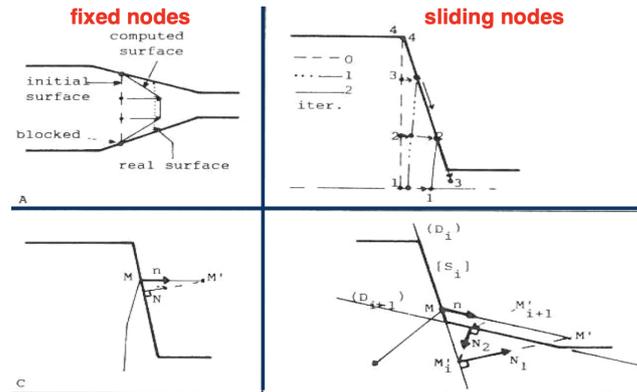


Figure 2.3.11: Contact treatment with Optimal control method [24]

6. In the *ForgeNxt*[®]2016 solver, the free-surface problem is computed with the global resolution method. The contact constraint is treated in the same step as the free surface calculation. If applied as a unilateral constraint, the contact condition in free-surface resolution results in either overestimated (see Figure 2.3.12a) or complete loss (see Figure 2.3.12b) of the contact and the computation stops without finding the solution.

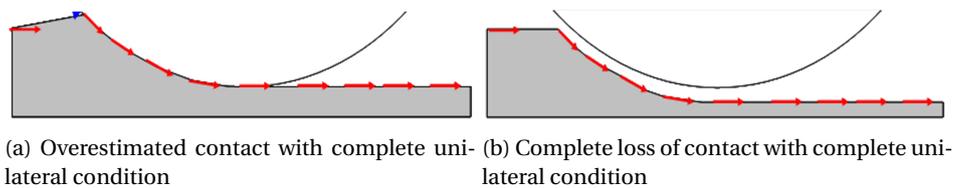


Figure 2.3.12: Problem with contact with global resolution [44]

So a unilateral condition is applied on all the nodes which are not in contact and a bilateral condition is applied on the nodes in contact. This means that a previous knowledge of the contact surface is necessary to apply this differential treatment of free surface boundary nodes. This contact surface is predicted in the mechanical step with a constraint on the velocity field. The contact conditions are applied with the penalty method and an associated Lagrangian is computed at the end of each step and is used to update the contact surface. It is on this contact surface, the bilateral condition is applied in the free surface resolution. This treatment improves the contact retention until the end of the simulation. The contact surface is again updated after the free surface computation from the geometric criterion which looks for the penetration nodes. However, certain adhesion parameters become necessary to ensure the adherence of the nodes to the tool in the normal direction, and without suppressing the lateral spread. More details on the contact coupling used in *ForgeNxt*[®]2016 are presented in detail in the following chapter 3.

Résumé

La modélisation de procédés continus dans une hypothèse stationnaire implique la résolution d'un problème thermo-mécanique. La première étape de la résolution est la description cinématique de l'écoulement du matériau qui peut être effectuée soit dans un cadre Lagrangien, soit en Eulerien, soit en ALE ou soit avec la technique de "Reference Frame". Le problème thermo-mécanique en Lagrangien est généralement dépendant du temps et résolu avec une formulation incrémentale. La formulation non-incrémentale peut résoudre directement l'état stationnaire est beaucoup moins coûteuse en terme de temps de calcul en comparaison avec la formulation incrémentale. Parmi les différentes formulations non-incrémentales, la formulation stationnaire est la meilleure en terme d'accélération et c'est ce qui est utilisé dans l'algorithme de ForgeNxt® 2016. Cette approche implique un fort couplage entre les champs de vitesse et la forme du domaine et donc ici l'utilisation d'un algorithme itératif de point-fixe. Cet algorithme a deux étapes principales: (i) résolution thermomécanique pour calculer les champs de vitesse et de température, et (ii) résolution de surface libre pour calculer la forme stationnaire du domaine. La recherche bibliographique est donc concentrée sur la formulation stationnaire, pour la modélisation mathématique de la mise en forme avec les équations d'équilibre, le comportement des matériaux, la résolution du contact, et le calcul de surface libre. L'étude bibliographique est focalisée sur la modélisation de l'élastoplasticité et ainsi sur le traitement de la surface libre et du contact dans la formulation stationnaire. Pour l'élastoplasticité, parmi les méthodes additives (Prandtl-Reuss) et multiplicatives pour sa modélisation, les premières décrivent mieux les problèmes de mise en forme de plasticité hypoélastique dominante et sont donc recommandées pour les lois de comportement élastoplastiques. Les variables d'état dépendant de l'historique peuvent être résolues avec: (i) l'intégration sur les lignes de courant et (ii) la résolution par éléments finis (globale) des équations. La première est simple à implémenter (pour les maillages structurés) et ne nécessite aucune résolution par éléments finis. En revanche, l'adaptation de la seconde est générale et ne nécessite pas de maillage structuré. Toutefois, elle impose une description du gradient résiduel qui peut devenir indéfini lorsque la variable interne change brusquement au début de la plasticité. Cependant, le défi le plus important est la description des lignes de courant dans les maillages non structurés, défi qui n'a pas été tenté dans la bibliographie. Le traitement du contact dans la formulation stationnaire est un problème notoire. Pour le contact, les équations de Signorini décrivent la condition d'impénétrabilité. Cette condition est appliquée dans la direction normale, et elle empêche la matière de pénétrer dans l'outil. On peut l'imposer avec, soit les multiplicateurs de Lagrange ou soit la méthode de pénalisation. Dans les formulations stationnaires, le plus grand problème est associé avec le couplage du contact dans les deux étapes (mécanique et surface libre). Dans la résolution de la surface libre, la condition d'impénétrabilité est imposée avec pénalisation en même temps que la condition de surface libre pour éviter la pénétration de noeuds de la surface libre dans l'outil. Par conséquent, une condition unilatérale du contact est imposée sur une ligne de courant, ce qui pose plusieurs problèmes: on observe des oscillations ou des marches avant le contact de l'outil. Cependant, le traitement du contact pour la formulation stationnaire est un problème difficile et reste une piste peu explorée. Néanmoins, c'est l'un des paramètres critiques pour stabiliser la solution et augmenter la robustesse de l'algorithme. Dans la bibliographie, ce couplage n'est pas étudié, et les auteurs traitent généralement ces problèmes de manière indépendante. Dans sa thèse, Ripert a essayé de traiter le couplage du contact mais sans étudier la consistance des deux systèmes d'équations.

Bibliography

- [1] A. Agrawal and P.R. Dawson. A comparison of Galerkin and streamline techniques for integrating strains from an Eulerian flow field. *International journal for numerical methods in engineering*, 21(5):853–881, 1985.
- [2] C. Aliaga. *Simulation numérique par éléments finis en 3D du comportement thermomécanique au cours du traitement thermique d'aciers : application à la trempe de pièces forgées ou coulées*. PhD thesis, ENSMP, 2000. URL <http://www.theses.fr/2000ENMP0882>.
- [3] B. Armand. *Contribution à la modélisation par éléments finis des surfaces libres en régime permanent*. PhD thesis, ENSMP, 1987.
- [4] D.N. Arnold, F. Brezzi and B. Cockburn, and L.D. Marini. Unified analysis of discontinuous galerkin methods for elliptic problems. *SIAM journal on numerical analysis*, 39(5):1749–1779, 2002.
- [5] D. Balagangadhar and D. A. Tortorelli. Design of large-deformation steady elastoplastic manufacturing processes. part i: a displacement-based reference frame formulation. *International Journal for Numerical Methods in Engineering*, 49(7):899–932, 2000.
- [6] D. Balagangadhar, G.A. Dorai, and D.A. Tortorelli. A displacement-based reference frame formulation for steady-state thermo-elasto-plastic material processes. *International journal of solids and structures*, 36(16):2397–2416, 1999.
- [7] T. Belytschko, W.K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2013.
- [8] R. Boman, L. Papeleux, Q.V. Bui, and J.P. Ponthot. Application of the Arbitrary Lagrangian-Eulerian formulation to the numerical simulation of cold roll forming process. *Journal of Materials Processing Technology*, 177(1-3):621–625, 2006.
- [9] A.N. Brooks and T.J.R. Hughes. Streamline Upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer methods in applied mechanics and engineering*, 32(1):199–259, 1982.
- [10] J.L. Chenot, P. Montmitonnet, A. Bern, and C. Bertrand-Corsini. A method for determining free surfaces in steady state finite element computations. *Computer Methods in Applied Mechanics and Engineering*, 92(2):245 – 260, 1991. ISSN 0045-7825. doi: [https://doi.org/10.1016/0045-7825\(91\)90242-X](https://doi.org/10.1016/0045-7825(91)90242-X). URL <http://www.sciencedirect.com/science/article/pii/004578259190242X>.
- [11] P.R. Dawson and E.G. Thompson. Finite element analysis of steady-state elasto-visco-plastic flow by the initial stress-rate method. *International Journal for Numerical Methods in Engineering*, 12(1):47–57, 1978.
- [12] J. Donea, S. Giuliani, and J.P. Halleux. An Arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions. *Computer methods in applied mechanics and engineering*, 33(1-3):689–723, 1982.

- [13] J. Donea, A. Huerta, J.P. Ponthot, and A. Rodriguez-Ferran. Arbitrary Lagrangian-Eulerian methods. *Encyclopedia of Computational Mechanics*, 2004.
- [14] K.R.J. Ellwood, T.C. Papanastasiou, and J.O. Wilkes. Three-dimensional streamlined finite elements: Design of extrusion dies. *International Journal for Numerical Methods in Fluids*, 14 (1):13–24, 1992. doi: 10.1002/flid.1650140103. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/flid.1650140103>.
- [15] M.S. Gadala, M.R. Movahhedy, and J. Wang. On the mesh motion for ALE modeling of metal forming processes. *Finite Elements in Analysis and Design*, 38(5):435–459, 2002.
- [16] V. Ganvir, A. Lele, R. Thaokar, and B.P. Gautham. Prediction of extrudate swell in polymer melt extrusion using an Arbitrary Lagrangian-Eulerian (ALE) based finite element method. *Journal of Non-Newtonian Fluid Mechanics*, 156(1-2):21–28, 2009.
- [17] S. Guerdoux. *Numerical simulation of the friction stir welding process*. PhD thesis, ENSMP, 2007.
- [18] A. Hacquin. *Modélisation Thermomécanique Tridimensionnelle du Laminage - Couplage Bande/ Cylindres*. PhD thesis, ENSMP, 1996.
- [19] A. Hacquin, P. Montmitonnet, and J.P. Guillerault. A steady state thermo-elastoviscoplastic finite element model of rolling with coupled thermo-elastic roll deformation. *Journal of Materials Processing Technology*, 60(1):109 – 116, 1996. ISSN 0924-0136.
- [20] F.H. Harlow and E.H. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluids*, 8(12):2182–2189, 1965.
- [21] R. Hill. *The mathematical theory of plasticity*, volume 11. Oxford university press, 1998.
- [22] C.W. Hirt and B.D. Nichols. A computational method for free surface hydrodynamics. *Pressure Vessel Technology*, 1981.
- [23] R. Hooke. 1678. lectures de "potentia restitutiva", or of spring, explaining the power of springing bodies. *Printed for John Martyn printer to the Royal Society, Bell in St. Paul's church-yard*, 1931.
- [24] P. Montmitonnet J.L. Chenot, P. Buessler, and F. Fau. Finite element computation of spread in hot flat and shape rolling with a steady state approach. *Engineering Computations*, 8:245–255, 12 1991.
- [25] A. Karagiannis, A. N. Hrymak, and J. Vlachopoulos. Three-dimensional studies on bicomponent extrusion. *Rheologica Acta*, 29(1):71–87, 01 1990.
- [26] A. N. Kim, H.W. Lee, J.H. Min, and Y.T. Im. Steady state finite element simulation of bar rolling processes based on rigid-viscoplastic approach. *International Journal for Numerical Methods in Engineering*, 63(11):1583–1603, 2005. doi: 10.1002/nme.1328. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1328>.
- [27] H.J. Kim, T.H. Kim, and S.M. Hwang. New free surface scheme for analysis of plastic deformation in shape rolling. *Journal of Materials Processing Technology*, 104:81–93, 08 2000. doi: 10.1016/S0924-0136(00)00519-7.
- [28] S.Y. Kim and Y.T. Im. Three-dimensional finite element analysis of non-isothermal shape rolling. *Journal of Materials Processing Technology*, 127(1):57 – 63, 2002. ISSN 0924-0136. doi: [https://doi.org/10.1016/S0924-0136\(02\)00256-X](https://doi.org/10.1016/S0924-0136(02)00256-X). URL <http://www.sciencedirect.com/science/article/pii/S092401360200256X>.

- [29] P. Knobloch. On the choice of the SUPG parameter at outflow boundary layers. *Adv. Comput. Math.*, 31:369–389, 10 2009. doi: 10.1007/s10444-008-9075-6.
- [30] E.H. Lee. Elastic-plastic deformation at finite strains. *Journal of Applied Mechanics*, 1969.
- [31] H.S. Lee and R.B. Haber. Eulerian-Lagrangian methods for crack growth in creeping materials. In *Proceedings of the 1993 ASME Winter Annual Meeting*, pages 141–153. Publ by ASME, 1993.
- [32] Y.S. Lee, P.R. Dawson, and T.B. Dewhurst. Bulge predictions in steady state bar rolling processes. *International journal for numerical methods in engineering*, 30(8):1403–1413, 1990.
- [33] A.M. Maniatty, P.R. Dawson, and G.G. Weber. An Eulerian elasto-viscoplastic formulation for steady-state forming processes. *International journal of mechanical sciences*, 33(5):361–377, 1991.
- [34] E. Mitsoulis. Three-dimensional non-newtonian computations of extrudate swell with the finite element method. *Computer Methods in Applied Mechanics and Engineering*, 180(3): 333 – 344, 1999. ISSN 0045-7825. doi: [https://doi.org/10.1016/S0045-7825\(99\)00172-3](https://doi.org/10.1016/S0045-7825(99)00172-3). URL <http://www.sciencedirect.com/science/article/pii/S0045782599001723>.
- [35] K. Mori and K. Osakada. Simulation of three-dimensional deformation in rolling by the finite-element method. *International Journal of Mechanical Sciences*, 26(9):515 – 525, 1984. ISSN 0020-7403. doi: [https://doi.org/10.1016/0020-7403\(84\)90005-5](https://doi.org/10.1016/0020-7403(84)90005-5). URL <http://www.sciencedirect.com/science/article/pii/0020740384900055>.
- [36] K. Mori and K. Osakada. Finite element simulation of three-dimensional deformation in shape rolling. *International Journal for Numerical Methods in Engineering*, 30(8):1431–1440, 1990. doi: 10.1002/nme.1620300807. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1620300807>.
- [37] K. Mori, K. Osakada, and T. Oda. Simulation of plane-strain rolling by the rigid-plastic finite element method. *International Journal of Mechanical Sciences*, 24(9):519 – 527, 1982. ISSN 0020-7403. doi: [https://doi.org/10.1016/0020-7403\(82\)90044-3](https://doi.org/10.1016/0020-7403(82)90044-3). URL <http://www.sciencedirect.com/science/article/pii/0020740382900443>.
- [38] Y. Mu, G. Zhao, S. Qin, and A. Chen. Numerical simulation of three-dimensional polymer extrusion flow with differential viscoelastic model. *Polymers for Advanced Technologies*, 18 (12):1004–1014, 2007. doi: 10.1002/pat.959. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/pat.959>.
- [39] R.E. Nickell, R.I. Tanner, and B. Caswell. The solution of viscous incompressible jet and free-surface flows using finite-element methods. *Journal of Fluid Mechanics*, 65(1):189–206, 1974. doi: 10.1017/S0022112074001339.
- [40] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.
- [41] S. Philippe. *Développement d’une formulation arbitrairement Lagrangienne Eulérienne pour la simulation tridimensionnelle du laminage de produits plats*. PhD thesis, ENSMP, 2009.
- [42] X. Qin and P. Michaleris. Eulerian elasto-visco-plastic formulations for residual stress prediction. *International Journal for Numerical Methods in Engineering*, 77(5):634–663, 2009. doi: 10.1002/nme.2429.
- [43] S.B. Brown and K.H. Kim and L. Anand. An internal variable constitutive model for hot working of metals. *International journal of plasticity*, 5(2):95–130, 1989.

- [44] U. Ripert. *Méthode itérative de recherche de l'état stationnaire des procédés de mise en forme : application au laminage*. PhD thesis, ENSMP, 2014. URL <http://www.theses.fr/2014ENMP0006>.
- [45] U. Ripert, L. Fourment, and J.L. Chenot. An upwind least square formulation for free surfaces calculation of viscoplastic steady-state metal forming problems. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1):1, 2015.
- [46] J.Y. Shanghvi and P. Michaleris. Thermo-elasto-plastic finite element analysis of quasi-state processes in Eulerian reference frames. *International journal for numerical methods in engineering*, 53(7):1533–1556, 2002.
- [47] J.C. Simo and T.J.R. Hughes. *Computational inelasticity*, volume 7. Springer Science & Business Media, 2006.
- [48] G. Sola. *Contribution a la modélisation thermomécanique tridimensionnelle par éléments finis du laminage a chaud des tubes : calcul multicage*. PhD thesis, ENSMP, 1994. URL <http://www.theses.fr/1994ENMP0478>.
- [49] A. Soulaïmani, M. Fortin, G. Dhatt, and Y. Ouellet. Finite element simulation of two- and three-dimensional free surface flows. *Computer Methods in Applied Mechanics and Engineering*, 86(3):265 – 296, 1991. ISSN 0045-7825. doi: [https://doi.org/10.1016/0045-7825\(91\)90224-T](https://doi.org/10.1016/0045-7825(91)90224-T). URL <http://www.sciencedirect.com/science/article/pii/S004578259190224T>.
- [50] E.G. Thompson and S.W. Yu. A flow formulation for rate equilibrium equations. *International journal for numerical methods in engineering*, 30(8):1619–1632, 1990.
- [51] K. Traoré. *Simulation thermomécanique du laminage circulaire : Développement d'une formulation quasi-Eulérienne tridimensionnelle sur une architecture parallèle*. PhD thesis, ENSMP, 2001. URL <http://www.theses.fr/2001ENMP1294>.
- [52] S.O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of computational physics*, 1992.
- [53] M. Vacance. *Modélisation tridimensionnelle par éléments finis du laminage a chaud des tubes sans soudure*. PhD thesis, ENSMP, 1993. URL <http://www.theses.fr/1993ENMP0384>.
- [54] M. Vacance, E. Massoni, and J.L. Chenot. Multi stand pipe mill finite element model. *Journal of Materials Processing Technology*, 24:421 – 430, 1990. ISSN 0924-0136. doi: [https://doi.org/10.1016/0924-0136\(90\)90203-7](https://doi.org/10.1016/0924-0136(90)90203-7).
- [55] K.D. Van and M.H. Maitournam. Steady-state flow in classical elastoplasticity: applications to repeated rolling and sliding contact. *Journal of the Mechanics and Physics of Solids*, 41(11): 1691–1710, 1993.
- [56] K.Y. Volokh. An approach to elastoplasticity at large deformations. *European Journal of Mechanics - A/Solids*, 39:153–162, 05 2013. ISSN 0997-7538. doi: [10.1016/j.euromechsol.2012.11.002](https://doi.org/10.1016/j.euromechsol.2012.11.002). URL <http://dx.doi.org/10.1016/j.euromechsol.2012.11.002>.
- [57] H.H. Wisselink and J. Huétink. 3D FEM simulation of stationary metal forming processes with applications to slitting and rolling. *Journal of Materials Processing Technology*, 148(3): 328 – 341, 2004. ISSN 0924-0136. doi: <https://doi.org/10.1016/j.jmatprotec.2004.02.036>. URL <http://www.sciencedirect.com/science/article/pii/S0924013604002328>.
- [58] P. Wriggers and G. Zavarise. *Computational Contact Mechanics*, chapter 6. American Cancer Society, 2004. ISBN 9780470091357. doi: [10.1002/0470091355.ecm033](https://doi.org/10.1002/0470091355.ecm033). URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/0470091355.ecm033>.

- [59] X. Xu, G. Zhao, S. Qin, and W. Wang. Numerical simulation of viscoelastic extrudate swell through elliptical ring die. *Chinese Journal of Chemical Engineering*, 19(1):10 – 17, 2011. ISSN 1004-9541. doi: [https://doi.org/10.1016/S1004-9541\(09\)60170-1](https://doi.org/10.1016/S1004-9541(09)60170-1). URL <http://www.sciencedirect.com/science/article/pii/S1004954109601701>.
- [60] K. Yamada, S. Ogawa, and S. Hamazu. Three-dimensional analysis of mandrel rolling using rigid-plastic finite element method. In *Proc. Numiform 89, Thompson et al (eds.) Balkema, Rotterdam ISBN 9061918979 (1989)*. Publ by ASME, 1989.
- [61] Y. Yu. *A Displacement Based FE Formulation for Steady State Problems*. PhD thesis, Universiteit Twente, 2005.

Chapter 3

Existing steady-state formulation and limitations

Contents

3.1	Generalized ForgeNxt®2016 steady-state algorithm	44
3.2	Thermo-mechanical problem formulation	44
3.2.1	Description of the domain	45
3.2.2	Initialization of the thermo-mechanical problem	45
3.2.3	Mechanical Problem	46
3.2.4	Thermal Problem	53
3.2.5	Equivalent strain	54
3.2.6	Resolution of the thermo-mechanical problem in the iterative algorithm	55
3.3	Free-surface resolution formulation	56
3.3.1	Basic equations	56
3.3.2	Weak formulation of free-surface equation	57
3.3.3	Finite Element resolution of free-surface equation	57
3.3.4	Extension of the free-surface correction to multi-DoF (with edges)	58
3.3.5	Edge Detection	59
3.3.6	Volume Mesh Regularization and Mesh Update	59
3.3.7	Free-surface Resolution in the Iterative Algorithm	60
3.4	Contact treatment in a steady-state frame	60
3.4.1	Contact coupling	60
3.4.2	Summary of the ForgeNxt®2016 Fixed-point resolution algorithm	61
3.5	Application to hot-rolling test-cases	62
3.5.1	Flat rolling problem: VP-Test-Case-01	63
3.5.2	Shape rolling problem: VP-Test-Case-02	67
3.5.3	Comparison of solution	68
3.6	Reasoning for instabilities observed with VP-Test-Case-02	69
3.7	Conclusions from the chapter	70

The different possibilities for solving continuous forming problems with non-incremental methods, specifically with the steady-state formulation, have been discussed in the literature investigation in the previous Chapter 2. The present work is an extension of the steady-state algorithm developed by [7], which has been implemented in the ForgeNxt[®]2016. The algorithm is based on the mixed velocity-pressure formulation of Fg3[®], and also conforms to the inherent constraints of the Fg3[®] solver, such as the data structure, as well as the contact formulation. The proposed formulation promised an excellent computation speedup in comparison to the incremental resolution methods. However, the formulation is found to be not robust with complex industrial cases. Before providing a detailed solution to this problem, it is important to understand the different aspects of the existing ForgeNxt[®]2016 formulation for the resolution of continuous forming problems. Subsequently, the existing algorithm is tested with simple and complex industrial hot rolling test-cases to examine the problems experienced.

3.1 Generalized ForgeNxt[®]2016 steady-state algorithm

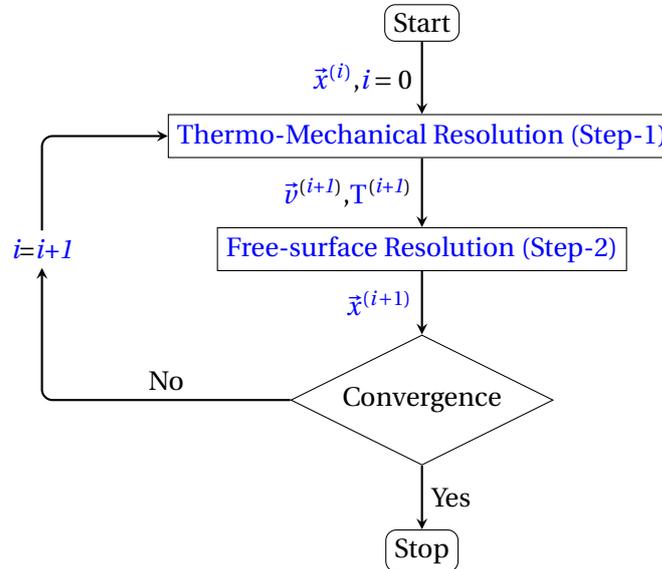


Figure 3.1.1: ForgeNxt[®]2016 fixed-point iterative steady-state algorithm for hot-rolling problems

The generalized iterative resolution algorithm adopted in ForgeNxt[®]2016 is shown in the Figure 3.1.1. There are two main steps involved. In the Step-1, the thermo-mechanical problem is resolved which involves the resolution of momentum and energy equilibrium equations. The kinematic description is done in the Eulerian frame. Hence, a control volume (fixed domain geometry) describing the computation domain (Figure 3.2.1) is modeled (Ω_x) on which the steady-state formulation is described. The Step-2 represents the computation of steady-state shape that conforms to the computed velocity field from the mechanical resolution. The computation domain Ω_x is corrected, and the fixed-point iterative loop continues until the convergence of the thermo-mechanical fields, the state-variables, and the steady-state shape in the computation domain.

3.2 Thermo-mechanical problem formulation

The velocity is the primary unknown of the Stokes problem and computed with the Finite Element Method. The thermo-mechanical problem is weakly coupled and is thus solved iteratively. In the incremental formulation, the state-variable equivalent strain $\bar{\epsilon}$ and the temperature field T are time-integrated, whereas in the steady-state formulation, time-independent formulations are described in the Eulerian domain. These equations are convection dominant problems and hence

solved with the FE based convection solver in the Cimlib library. More details on the thermo-mechanical equations of the steady-state algorithm are presented in the following sections.

3.2.1 Description of the domain

Before describing the equations, it is important to understand the complete computational domain including the subdomains on which the boundary conditions are applied. The Eulerian problem is described in the control volume domain Ω_x as shown in Figure 3.2.1. The control volume has the respective input Γ_{in} and output planes Γ_{out} through which the material flows into ($\vec{v} \cdot \vec{n} < 0$) and out ($\vec{v} \cdot \vec{n} > 0$) of the domain, similar to a fluid problem. For a rolling problem with symmetric rolls, the choice of symmetry plane(s) Γ_{sym} enables the reduction of the computational domain. The rest of the surface boundary Γ is considered to be free-surface. The surface $\Gamma_c \subset \Gamma$ is where the workpiece surface boundary is in contact with the tool. The vector \vec{n}^{tool} defines the unit normal defined at the surface of the workpiece in the direction of the tool. In addition, the free surface edge Γ_{edge} must as well be separately defined for problems with geometric singularities, especially for the free surface correction problems.

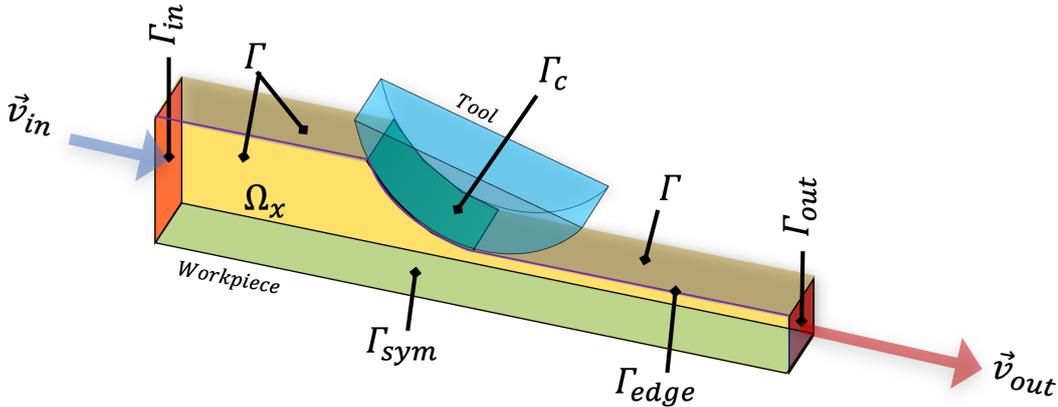


Figure 3.2.1: Schematic representation of a rolling problem

3.2.2 Initialization of the thermo-mechanical problem

An initial assumption of the geometry must be made. [7] presented three different methods (see Figure 3.2.1) for the initial geometry computation (i) Extrusion of the input plane in the direction of rolling, (ii) Material trimming (iii) Forging. In the *extrusion* method, the input plane Γ_{in} is extruded in the rolling direction. The nodes of the domain that intersect into the tool are projected back to the tool surface. So the initialization geometry is not far from the steady-state and thus only a few iterations are required for the convergence of the algorithm. The *material trimming* is a standard operation in Fg3[®] in which the workpiece material intersecting the tool is removed. In the *forging* method, the workpiece material is forged with the same tool (in the vertical direction) to get a smooth initial geometry.

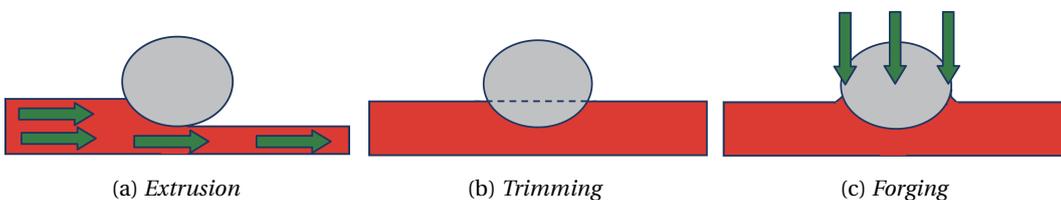


Figure 3.2.2: Initialization of geometry in [7]

3.2.3 Mechanical Problem

The material forming problem is described with the theory of continuum mechanics, and the material is considered to be homogeneous. The Boundary Value problem associated with the mechanical resolution of the forming problem is discussed in this section. The conservation laws, material model, and the respective boundary conditions of the rolling process provide a well-posed problem.

3.2.3.1 Conservation laws

The conservation of linear momentum (or the force equilibrium) is described on the computation domain Ω_x with equation (3.2.1a) on the current configuration with the Cauchy's stress tensor $\boldsymbol{\sigma}$ (see Figure 3.2.1). As in the incremental form, the gravitational and the inertial effects are negligible in comparison to the deformation stresses and hence neglected. We recall the conservation of the mass seen in equation (2.3.3a) and for viscoplastic materials the homogeneous density results in the incompressibility condition in equation (3.2.1b). It is to be impressed on the reader that the existing ForgeNxt[®]2016 formulation, only Viscoplastic (VP) constitutive models (for hot metal forming) are treated, hence, the elastic deformations are considered to be negligible. However, for the elastoplasticity constitutive models, this assumption is not true and the same would be updated in Chapter 5.

$$\vec{\nabla} \cdot \boldsymbol{\sigma} + \underbrace{\rho \vec{b}}_{=0} = 0 \quad \text{in} \quad \Omega_x \in \mathbb{R}^3 \quad (3.2.1a)$$

$$\vec{\nabla} \cdot \vec{v} = 0 \quad \text{in} \quad \Omega_x \in \mathbb{R}^3 \quad (3.2.1b)$$

3.2.3.2 Constitutive models

For the hot forming problems, the rate of deformation $\dot{\boldsymbol{\epsilon}}$ dependent viscoplastic constitutive models are used for modeling the material behavior. The viscoplastic models can be defined with the equation (3.2.2) as a relation between the deviatoric stress \boldsymbol{s} and deviatoric rate of deformation $\dot{\boldsymbol{\epsilon}}$. The same are found from the respective stress $\boldsymbol{\sigma}$ and rate of deformation $\dot{\boldsymbol{\epsilon}}$ tensors.

$$\boldsymbol{s} = \boldsymbol{\sigma} + p\mathbf{1} = \boldsymbol{\sigma} - \frac{1}{3} \text{tr}(\boldsymbol{\sigma}) \mathbf{1} \quad (3.2.2a)$$

$$\dot{\boldsymbol{\epsilon}} = \dot{\boldsymbol{\epsilon}} - \frac{1}{3} \text{tr}(\dot{\boldsymbol{\epsilon}}) \mathbf{1} \quad (3.2.2b)$$

The rate of deformation tensor is defined from the velocity field. To be clear, here T represents the transpose which is not to be confused with its general definition of Temperature:

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2} (\vec{\nabla} \vec{v} + \vec{\nabla}^T \vec{v}) \quad (3.2.3)$$

The one-dimensional representation of stress and strain are respectively defined with the equivalent (von Mises) stress $\bar{\sigma}$ and equivalent rate of deformation $\dot{\bar{\epsilon}}$ defined as following:

$$\bar{\sigma} = \sqrt{\frac{3}{2} \boldsymbol{s} : \boldsymbol{s}} \quad (3.2.4a)$$

$$\dot{\bar{\epsilon}} = \sqrt{\frac{2}{3} \dot{\boldsymbol{\epsilon}} : \dot{\boldsymbol{\epsilon}}} \quad (3.2.4b)$$

The equivalent strain $\bar{\epsilon}$ is defined with the time integration of the equivalent rate of deformation $\dot{\bar{\epsilon}}$ following a material point movement:

$$\dot{\bar{\epsilon}} = \frac{\partial \bar{\epsilon}}{\partial t} + \vec{v} \cdot \vec{\nabla} \bar{\epsilon} \quad (3.2.5)$$

While modeling material flows at hot temperatures, we invoke the viscoplastic potential Φ^{vp} to find the relation between the deviatoric stress \mathbf{s} with the rate of deformation $\dot{\mathbf{e}}$ (3.2.6a) with the Norton-Hoff constitutive law (3.2.6b).

$$\mathbf{s} = \frac{\partial \Phi^{vp}}{\partial \dot{\mathbf{e}}} \quad (3.2.6a)$$

$$\Phi^{vp} = \frac{K}{m+1} \left(\sqrt{3} \dot{\bar{\mathbf{e}}} \right)^{m+1} \quad (3.2.6b)$$

$$\mathbf{s} = \mu(\vec{\nu}) \dot{\mathbf{e}} = 2K \left(\sqrt{3} \dot{\bar{\mathbf{e}}} \right)^{m-1} \dot{\mathbf{e}} \quad (3.2.6c)$$

In the equation (3.2.6b), K is the material consistency, and m is the material sensibility to the rate of deformation. μ is the generalized nonlinear viscosity of the material, and the material behavior is dependent on the equivalent rate of deformation $\dot{\bar{\mathbf{e}}}$. Generally for metals, the material sensitivity index is $m < 0.3$, whereas $m = 0$ leads to a rigid plastic constitutive law and $m = 1$ leads to linear Newtonian behavior. Using the equation (3.2.4a), we can find the equivalent stress as following:

$$\bar{\sigma} = K \sqrt{3} \left(\sqrt{3} \dot{\bar{\mathbf{e}}} \right)^m \quad (3.2.7)$$

The dependence of material consistency K on strain and temperature T gives the expression for power law for a thermal-strain-hardening material in the equation (3.2.8), e.g.:

$$K(\bar{\epsilon}, T) = K_0 (\bar{\epsilon} + \epsilon_0)^n \exp\left(\frac{\beta}{T}\right) \quad (3.2.8)$$

Hence, the viscoplastic rate of deformation tensor $\dot{\mathbf{e}}^{vp}$ can be computed with the following relation, identical to the von Mises associated flow rule.

$$\dot{\mathbf{e}}^{vp} = \frac{3}{2} \dot{\bar{\mathbf{e}}} \frac{\mathbf{s}}{\bar{\sigma}} \quad (3.2.9)$$

3.2.3.3 Boundary conditions

For the unicity of the solution to the velocity (and the state variables) from the mechanical problem resolution, mixed boundary conditions on velocity and stress fields are applied at different boundary sub-domains shown in Figure 3.2.1. The input boundary Γ_{in} is always fixed during the steady-state resolution. The other sub-boundary domains remain fixed only in mechanical resolution and are updated after the Step-2.

Input and Output plane boundary conditions

At the Γ_{in} and Γ_{out} planes of the computational domain Ω_x , a stress-free condition is applied with equation (3.2.10a) as the inter-stand tension $\vec{T}_{IS}(\hat{e}_y, \hat{e}_z)$ is considered to be negligible for simplifying the equations. The \hat{e}_y, \hat{e}_z directions are orthogonal to the rolling direction. In addition, at the input and output plane nodes, a velocity boundary condition (3.2.10b) is applied in the tangential direction. We could also apply a constant normal velocity at these planes, but the equation (3.2.10b) is relatively simpler and logical to impose. The respective input and output planes are chosen far enough so that the plastic rate of deformation and the velocity gradient are nullified [8].

on $\Gamma_{in} \cup \Gamma_{out}$

$$\boldsymbol{\sigma} \cdot \vec{n} = \vec{T}_{IS}(\hat{e}_y, \hat{e}_z) \quad (3.2.10a)$$

$$\vec{\nu}_t = [\vec{\nu} - (\vec{\nu} \cdot \vec{n}) \vec{n}] = \vec{0} \quad (3.2.10b)$$

Symmetry boundary condition

At the symmetry planes Γ_{sym} , the material flow in the normal direction \vec{n}^{sym} and the stress in the tangential direction are nullified.

$$\text{on } \Gamma_{sym} \quad \vec{v} \cdot \vec{n}^{sym} = 0 \quad (3.2.11a)$$

$$\boldsymbol{\sigma} \cdot \vec{n}^{sym} - (\vec{n}^{sym} \cdot \boldsymbol{\sigma} \cdot \vec{n}^{sym}) \vec{n}^{sym} = 0 \quad (3.2.11b)$$

Free-surface conditions

On the boundary Γ , the free-surface conditions correspond to (i) stress-free state in the normal direction \vec{n} (3.2.12a), (ii) no material flow across the boundary (3.2.12b). However, only the free-surface condition on stress is applied in the mechanical step and the condition on velocity is applied in the free-surface correction step, which permits us to update the boundary of the domain.

$$\text{on } \Gamma_{sym} \quad \boldsymbol{\sigma} \cdot \vec{n} = \vec{0} \quad (3.2.12a)$$

$$\vec{v} \cdot \vec{n} = 0 \quad (3.2.12b)$$

Contact condition

At the contact surface Γ_c , separate conditions are applicable in the tangential and normal directions.

1. In the normal direction **KKT** (Signorini) conditions found for incremental formulation in the equation (2.3.17) for a contact with a rigid tool are presumed. However, when the steady-state conditions are reached after a finite time, these equations are described as:

$$\text{on } \Gamma_c \quad \left\{ \begin{array}{l} h(\vec{v}) = (\vec{v} - \vec{v}^{tool}) \cdot \vec{n}^{tool} \leq 0 \\ \sigma_n = (\boldsymbol{\sigma} \cdot \vec{n}) \cdot \vec{n} \leq 0 \\ h(\vec{v}) \sigma_n = 0 \end{array} \right. \quad (3.2.13)$$

In the steady-state $\vec{v}^{tool} \cdot \vec{n}^{tool} = 0$ and thus the constraint $h(\vec{v}_k)$ reduces to:

$$h(\vec{v}) = \vec{v} \cdot \vec{n}^{tool} \leq 0 \quad \text{on } \Gamma_c \quad (3.2.14)$$

2. In the tangential direction, the friction with the forming tools is modeled with the Norton law as introduced in the equation (2.3.30). The same is reproduced in (3.2.15) with the velocity formulation and relates the tangential stress $\vec{\tau}_f$, with the tangential sliding velocity $\Delta \vec{v}_t$ (3.2.15b). In the model, q is the slipping sensitivity coefficient.

$$\vec{\tau}_f = \boldsymbol{\sigma} \cdot \vec{n} - ((\boldsymbol{\sigma} \cdot \vec{n}) \cdot \vec{n}) \vec{n} = -\alpha_f K_f \|\Delta \vec{v}_t\|^{q-1} \Delta \vec{v}_t \quad \text{on } \Gamma_c \quad (3.2.15a)$$

$$\Delta \vec{v} = \vec{v} - \vec{v}^{tool} = \Delta \vec{v}_t + (\Delta \vec{v} \cdot \vec{n}) \vec{n} \quad (3.2.15b)$$

It is to be noted that Γ and the contact surface Γ_c are the unknowns of the fixed-point iterative steady-state algorithm. Starting from an initial assumption, the former is updated in the second step of the algorithm (see Section 3.3), whereas the latter is updated in both the steps (see Section 3.4.1).

3.2.3.4 Weak form of the Mechanical problem

Mixed (\vec{v}, p) form

The equations in the Section 3.2.3.1 to Section 3.2.3.3 represent the strong form of the Mechanical problem. There are 9 unknowns in the problem including the 3 components of velocity field \vec{v} and 6 components of stress tensor $\boldsymbol{\sigma}$. The problem can rather be written in mixed (\vec{v}, p) form, which is done by splitting the stress tensor $\boldsymbol{\sigma}$ into its deviatoric \mathbf{s} and hydrostatic parts as represented in equation (3.2.2a).

$$p = -\frac{1}{3} \text{tr} \boldsymbol{\sigma} \quad \text{in} \quad \Omega_x \quad (3.2.16)$$

In the mixed velocity-pressure formulation, the force equilibrium (3.2.1a) is written in the following form:

$$\vec{\nabla} \cdot \mathbf{s} - \vec{\nabla} p = 0 \quad \text{in} \quad \Omega_x \quad (3.2.17)$$

Application of variational formulation

The Virtual Power Principle is applied to find the weak or the weighted-residual form of the mixed velocity-pressure formulation of the problem. The adequate functional spaces for admissible kinematic velocities $\mathbb{V}^{\vec{v}}$, $\mathbb{V}_0^{\vec{v}}$ and for pressure \mathbb{V}^p are defined:

$$\mathbb{V}^{\vec{v}} = \left\{ \vec{v} \in [H^{1,0}(\Omega_x)]^3, \left\{ \begin{array}{l} (\vec{v} - \vec{v}^{tool}) \cdot \vec{n}^{tool} \leq 0 \quad \text{on} \quad \Gamma_c \\ \vec{v} \cdot \vec{n}^{sym} = 0 \quad \text{on} \quad \Gamma_{sym} \\ \vec{v} - (\vec{v} \cdot \vec{n}) \vec{n} = 0 \quad \text{on} \quad \Gamma_{in} \cup \Gamma_{out} \end{array} \right. \right\} \quad (3.2.18a)$$

$$\mathbb{V}_0^{\vec{v}} = \left\{ \vec{v} \in [H^{1,0}(\Omega_x)]^3, \left\{ \begin{array}{l} \vec{v} \cdot \vec{n}^{tool} \leq 0 \quad \text{on} \quad \Gamma_c \\ \vec{v} \cdot \vec{n}^{sym} = 0 \quad \text{on} \quad \Gamma_{sym} \\ \vec{v} - (\vec{v} \cdot \vec{n}) \vec{n} = 0 \quad \text{on} \quad \Gamma_{in} \cup \Gamma_{out} \end{array} \right. \right\} \quad (3.2.18b)$$

$$\mathbb{V}^p = \left\{ w \in L^2(\Omega_x), \quad \vec{\nabla} w \in [L^2(\Omega)]^3 \right\} \quad (3.2.18c)$$

Hence, the weak form of the mechanical problem (force equilibrium and incompressibility) is written as we look to solve for unknown $(\vec{v}, p) \in \mathbb{V}^{\vec{v}} \times \mathbb{V}^p$ such that, $\forall (\vec{v}^*, p^*) \in \mathbb{V}_0^{\vec{v}} \times \mathbb{V}^p$

$$\int_{\Omega_x} \vec{v}^* (\vec{\nabla} \cdot \mathbf{s} - \vec{\nabla} p) d\omega = 0 \quad (3.2.19a)$$

$$\int_{\Omega_x} p^* (\vec{\nabla} \cdot \vec{v}) d\omega = 0 \quad (3.2.19b)$$

Using Green's formula, we find the weighted-residual form with (3.2.20). The surface integral term is applicable on the contact surface Γ_c , which can be divided into respective normal and tangential terms. The former is a mixed Robin boundary condition representing the Signorini conditions in equation (3.2.13) and the latter is a Neumann condition representing the friction force in equation (3.2.15). Find $(\vec{v}, p) \in \mathbb{V}^{\vec{v}} \times \mathbb{V}^p$ such that, $\forall (\vec{v}^*, p^*) \in \mathbb{V}_0^{\vec{v}} \times \mathbb{V}^p$

$$\int_{\Omega_x} \mathbf{s}(\vec{v}) : \dot{\boldsymbol{\epsilon}}(\vec{v}^*) d\omega - \int_{\Gamma_c} \vec{\tau}_f(\vec{v}) \cdot \vec{v}^* ds - \int_{\Omega_x} p \vec{\nabla} \cdot (\vec{v}^*) d\omega - \int_{\Gamma_c} \sigma_n \vec{n} \cdot \vec{v}^* ds = 0 \quad (3.2.20a)$$

$$- \int_{\Omega_x} p^* \vec{\nabla} \cdot (\vec{v}) d\omega = 0 \quad (3.2.20b)$$

Discretization of variational formulation

The weak form of the problem is described in a space with infinite degrees of freedom, and we can bring it to finite dimension with the invocation of the finite element spaces. This calls for the

discretization of the spaces and use of a mesh representing the computation domain Ω_{x_h} which is now a union of disjointed subdomains $\Omega_{x_h}^e$

$$\Omega_{x_h} = \cup_e \Omega_{x_h}^e \quad (3.2.21)$$

Thus, the discretized mixed (\vec{v}, p) form of the problem can be written as: Find $(\vec{v}_h, p_h) \in \mathbb{V}_h^{\vec{v}} \times \mathbb{V}_h^p$ such that, $\forall (\vec{v}_h^*, p_h^*) \in \mathbb{V}_h^{\vec{v}} \times \mathbb{V}_h^p$

$$\int_{\Omega_{x_h}} \mathbf{s}(\vec{v}_h) : \dot{\mathbf{e}}(\vec{v}_h^*) d\omega_h - \int_{\Gamma_{c,h}} \vec{\tau}_f(\vec{v}_h) \cdot \vec{v}_h^* ds_h - \int_{\Omega_{x_h}} p_h \vec{\nabla} \cdot (\vec{v}_h^*) d\omega_h - \int_{\Gamma_{c,h}} \sigma_n \vec{n} \cdot \vec{v}_h^* ds_h = 0 \quad (3.2.22a)$$

$$- \int_{\Omega_{x_h}} p_h^* \vec{\nabla} \cdot (\vec{v}_h) d\omega_h = 0 \quad (3.2.22b)$$

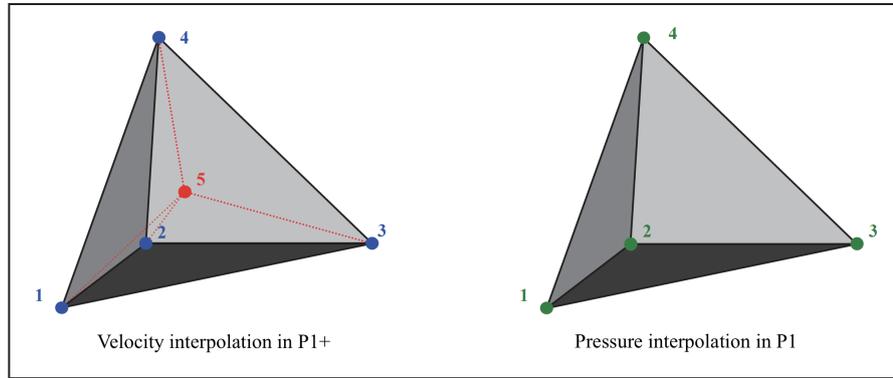


Figure 3.2.3: Quasi-linear P1+/P1 “mini-element” for \vec{v}, p interpolation

Since we are using hybrid finite elements with mixed (\vec{v}, p) formulation, the interpolation spaces for the elements chosen must comply to the Brezzi-Babuska compatibility condition [3]. In **Fig 3**[®], the weighted-residual form is discretized using tetrahedral finite element with the quasi-linear *P1+/P1 mini element* interpolation [2] satisfying the *inf-sup compatibility condition*. In this element, the pressure is interpolated with P1 linear functions, and their values are computed at the nodes. The velocity field is interpolated with P1+ function which is enriched with an additional degree of freedom at the center of the element. The enriched space uses the linear functions at the nodes and a bubble function for the center node. The latter is piece-wise continuous inside the tetrahedron element. The $(\Omega_{x_h}^e)_{i=1,4}$ define the four sub-elements of the tetrahedron element $\Omega_{x_h}^e$ (see Figure 3.2.3-left) and define the approximation spaces as:

$$\mathbb{V}_h^{\vec{v}} = \mathbb{V}_h^{\vec{v},l} \oplus \mathbb{V}_h^{\vec{v},b} \quad (3.2.23a)$$

$$\mathbb{V}_h^p = \{p_h \in C_0(\Omega_{x_h}) \text{ s.t. } e \in \tau_h, p_h|_{\Omega_{x_h}^e} \in \text{P1}(\Omega_{x_h}^e)\} \quad (3.2.23b)$$

$$\mathbb{V}_h^{\vec{v},l} = \{\vec{v}_h^l \in (C_0(\Omega_{x_h}))^3 \text{ s.t. } e \in \tau_h, \vec{v}_h^l|_{\Omega_h^e} \in (\text{P1}(\Omega_{x_h}^e))^3, \left. \begin{array}{l} \forall k \in \Gamma_c \quad \vec{v}_k \cdot \vec{n}_k^{tool} \leq 0 \\ \forall k \in \Gamma_{sym} \quad \vec{v}_k \cdot \vec{n}_k^{sym} = 0 \\ \forall k \in \Gamma_{in} \cup \Gamma_{out} \quad \vec{v}_k - (\vec{v}_k \cdot \vec{n}_k) \vec{n}_k = 0 \end{array} \right\} \quad (3.2.23c)$$

$$\mathbb{V}_h^{\vec{v},b} = \{\vec{v}_h^b \in (C_0(\Omega_{x_h}))^3 \text{ s.t. } e \in \tau_h, \forall i = 1, 4 \vec{v}_h^b|_{\Omega_{x_h}^e} \in (\text{P1}(\Omega_{x_h}^e))_i^3, \text{ and } \vec{v}_h^b = 0 \text{ on } \partial\Omega_{x_h}^e\} \quad (3.2.23d)$$

In the description above, the superscript l, b corresponds to the spaces corresponding to the linear and bubble interpolation functions of the velocity. The \vec{v}, p fields can be interpolated using the Finite Element form with equation (3.2.24), with nbn and nbe being the number of nodes and

elements in the mesh. N_k is the linear interpolation function associated with the nodes. At a reference node k , it is equal to 1 and 0 on the other nodes. N_e is the linear bubble interpolation function corresponding to the element e which is 1 at the centroid of the element and 0 at the boundary. The function N_e varies linearly on the sub-elements in the tetrahedron element e formed by the nodes and the centroid. The derivation of N_e inside the element is explained in Appendix A.1

$$\vec{v}_h = \vec{v}_h^l + \vec{v}_h^b = \sum_k^{nbn} N_k \vec{v}_k^l + \sum_e^{nbe} N_e \vec{v}_e^b \quad (3.2.24a)$$

$$p_h = \sum_k^{nbn} N_k p_k \quad (3.2.24b)$$

Treatment of contact in FE formulation

At the contact surface Γ_c , the Signorini conditions (3.2.14) are applied with a node-to-facet penalty formulation [9]. The unilateral condition ensures a free surface condition on the surface of the tool and also permits the possibility of a contact node to leave the contact surface. It is to be noted that this condition is applied only on the nodes which are already in contact. This zone must thus be predetermined, which is discussed in detail in Section 3.4.1. The contact condition is imposed with a penalty method, as done in ForgeNxt[®]inc. Hence, the resultant penalty functional together with the mechanical functional Φ is written as follows.

$$\Phi_p(\vec{v}) = \Phi(\vec{v}) + \frac{1}{2} \rho_c^{(1)} \sum_{k \in \Gamma_{h,c}} s_k [h(\vec{v}_k)]^{+2} \quad (3.2.25a)$$

$$s_k = \int_{\Gamma_{c,h}} N_k dS_h \quad (3.2.25b)$$

The operator $[\mathbf{a}]^+ = (\mathbf{a} + |\mathbf{a}|)/2$ denotes strictly positive values. s_k is the surface area associated to node k described with equation (3.2.25b), and $\rho_c^{(1)}$ is the penalty coefficient and the index (1) denotes the Step-1 of the algorithm. In principle, we can use gradient methods to find the unknown velocity field by minimizing this functional. The minimization of the functional corresponds to nullification of the derivative $\Phi'_p(\vec{v}) = 0$. Hence, the discretized residual from equation (3.2.22) is written as: Find $(\vec{v}_h^l, \vec{v}_h^b, p_h) \in \mathbb{V}_{h,0}^{v,l} \times \mathbb{V}_{h,0}^{v,b} \times \mathbb{V}_h^p$ such that, $\forall (\vec{v}_h^{l,*}, \vec{v}_h^{b,*}, p_h^*) \in \mathbb{V}_{h,0}^{v,l} \times \mathbb{V}_{h,0}^{v,b} \times \mathbb{V}_h^p$

$$\int_{\Omega_{xh}} \mathbf{s}(\vec{v}_h^l + \vec{v}_h^b) : \dot{\boldsymbol{\epsilon}}(\vec{v}_h^{l,*}) d\omega_h - \int_{\Omega_{xh}} p_h \vec{\nabla} \cdot (\vec{v}_h^{l,*}) d\omega_h - \int_{\Gamma_{c,h}} \vec{\tau}_{f,h}(\vec{v}_h^l) \cdot \vec{v}_h^{l,*} dS_h - \sum_{k \in \Gamma_{c,h}} s_k [h(\vec{v}_k)]^+ (\vec{n}_k^{tool} \cdot \vec{v}_k^{l,*}) = 0 \quad (3.2.26a)$$

$$\int_{\Omega_{xh}} \mathbf{s}(\vec{v}_h^l + \vec{v}_h^b) : \dot{\boldsymbol{\epsilon}}(\vec{v}_h^{b,*}) d\omega_h - \int_{\Omega_{xh}} p_h \vec{\nabla} \cdot (\vec{v}_h^{b,*}) d\omega_h = 0 \quad (3.2.26b)$$

$$- \int_{\Omega_{xh}} p_h^* \vec{\nabla} \cdot (\vec{v}_h^l + \vec{v}_h^b) d\omega_h = 0 \quad (3.2.26c)$$

From the orthogonality of the linear and bubble shape functions in the P1+ interpolation, we find that the following terms are nullified.

$$\int_{\Omega_{xh}} \mathbf{s}(\vec{v}_h^b) : \dot{\boldsymbol{\epsilon}}_h(\vec{v}_h^{l,*}) d\omega_h = \int_{\Omega_{xh}} \mathbf{s}_h(\vec{v}_h^l) : \dot{\boldsymbol{\epsilon}}_h(\vec{v}_h^{b,*}) d\omega_h = 0 \quad (3.2.27)$$

The system of equations (3.2.26) can be described with the residuals R_k^l, R_k^b, R_k^p as following: Find $(\vec{v}_h^l, \vec{v}_h^b, p_h) \in \mathbb{V}_h^{v,l} \times \mathbb{V}_h^{v,b} \times \mathbb{V}_h^p$, with $k \in [1, 4]$ such that:

$$R_k^l(\vec{v}_h^l, \vec{v}_h^b, p_h) = \int_{\Omega_h} \mathbf{s}(\vec{v}_h^l) : \vec{\mathbb{B}}_k^l d\omega_h - \int_{\Omega_h} p_h \vec{\nabla} \cdot (\vec{\mathbb{B}}_k^l) d\omega_h - \dots = 0 \quad (3.2.28a)$$

$$\mathbf{R}^b(\vec{v}_h^l, \vec{v}_h^b, p_h) = \int_{\Omega_h} \mathbf{s}(\vec{v}_h^b) : \vec{\mathbb{B}}^b d\omega_h - \int_{\Omega_h} p_h \vec{\nabla} \cdot (\vec{\mathbb{B}}^b) d\omega_h = 0 \quad (3.2.28b)$$

$$\mathbf{R}_k^p(\vec{v}_h^l, \vec{v}_h^b, p_h) = - \int_{\Omega_h} N_k \vec{\nabla} \cdot (\vec{v}_h^l + \vec{v}_h^b) d\omega_h = 0 \quad (3.2.28c)$$

The nonlinear system of equations is solved with Newton-Raphson iterative algorithm to compute the incremental velocity $\delta \vec{v}^{l,(i+1)}$, $\delta \vec{v}^{b,(i+1)}$ and pressure field $\delta p^{(i+1)}$ as following:

$$\vec{v}^{l,(i+1)} = \vec{v}^{l,(i)} + \delta \vec{v}^{l,(i)} \quad (3.2.29a)$$

$$\vec{v}^{b,(i+1)} = \vec{v}^{b,(i)} + \delta \vec{v}^{b,(i)} \quad (3.2.29b)$$

$$p^{(i+1)} = p^{(i)} + \delta p^{(i)} \quad (3.2.29c)$$

The system of equations is solved until $\vec{v}^{(i+1)} \approx \vec{v}^{(i)}$. As we look to solve the equations with the Newton-Raphson method, we must also describe the corresponding Hessian term with respect to each unknown variable (to be computed). In (3.2.28), the Residual vector was presented. This Hessian matrix H_{xy} is described with the derivative of the Residual R_x with respect to the variable y from the set of unknowns. $\forall k, \kappa \in [1, 4]$

$$\mathbf{R}_k^{l(i+1)} = \mathbf{R}_k^{l(i)} + \underbrace{\frac{\partial \mathbf{R}_k^{l(i)}}{\partial \vec{v}_\kappa^l}}_{H_{k\kappa}^{ll(i)}} \delta \vec{v}_\kappa^{l(i+1)} + \underbrace{\frac{\partial \mathbf{R}_k^{l(i)}}{\partial \vec{v}^b}}_{H_k^{lb(i)}} \delta \vec{v}^{b(i+1)} + \underbrace{\frac{\partial \mathbf{R}_k^{l(i)}}{\partial p_\kappa}}_{H_{k\kappa}^{lp(i)}} \delta p_\kappa^{(i+1)} = 0 \quad (3.2.30a)$$

$$\mathbf{R}^{b(i+1)} = \mathbf{R}^{b(i)} + \underbrace{\frac{\partial \mathbf{R}^{b(i)}}{\partial \vec{v}_\kappa^l}}_{H_\kappa^{bl(i)}} \delta \vec{v}_\kappa^{l(i+1)} + \underbrace{\frac{\partial \mathbf{R}^{b(i)}}{\partial \vec{v}^b}}_{H^{bb(i)}} \delta \vec{v}^{b(i+1)} + \underbrace{\frac{\partial \mathbf{R}^{b(i)}}{\partial p_\kappa}}_{H_\kappa^{bp(i)}} \delta p_\kappa^{(i+1)} = 0 \quad (3.2.30b)$$

$$\mathbf{R}_k^{p(i+1)} = \mathbf{R}_k^{p(i)} + \underbrace{\frac{\partial \mathbf{R}_k^{p(i)}}{\partial \vec{v}_\kappa^l}}_{H_{k\kappa}^{pl(i)}} \delta \vec{v}_\kappa^{l(i+1)} + \underbrace{\frac{\partial \mathbf{R}_k^{p(i)}}{\partial \vec{v}^b}}_{H_k^{pb(i)}} \delta \vec{v}^{b(i+1)} + \underbrace{\frac{\partial \mathbf{R}_k^{p(i)}}{\partial p_\kappa}}_{H_{k\kappa}^{pp(i)}} \delta p_\kappa^{(i+1)} = 0 \quad (3.2.30c)$$

At each Newton-Raphson iteration (i), the residual and the Hessian matrices and Residual vector are updated from the newly computed velocities \vec{v}^l , \vec{v}^b and pressure p^l . In the matrix below, even though the term $H_{k\kappa}^{pp(i)} = 0$ for the viscoplastic materials, it is retained for a generalistic representation, as this term is non-zero for elastoplasticity and referred to in Section 5.2.5.

$$\begin{bmatrix} H_{k\kappa}^{ll(i)} & 0 & H_{k\kappa}^{lp(i)} \\ 0 & H^{bb(i)} & H_\kappa^{bp(i)} \\ H_{k\kappa}^{pl(i)} & H_k^{pb(i)} & H_{k\kappa}^{pp(i)} \end{bmatrix} \begin{bmatrix} \delta \vec{v}_\kappa^{l(i+1)} \\ \delta \vec{v}^{b(i+1)} \\ \delta p_\kappa^{(i+1)} \end{bmatrix} = - \begin{bmatrix} \mathbf{R}_k^{l(i)} \\ \mathbf{R}^{b(i)} \\ \mathbf{R}_k^{p(i)} \end{bmatrix} \quad (3.2.31)$$

It is to be reminded here to the reader that the constitutive law for the viscoplastic material relates the deviatoric stress \mathbf{s} with the rate of deformation tensor $\dot{\boldsymbol{\epsilon}}$ with the equation (3.2.6c). With the stabilization approach, the viscoplastic viscosity μ does not have dependency on the bubble function, meaning:

$$\mu(\vec{v}) \approx \mu(\vec{v}^l) \quad \Rightarrow \mathbf{s}(\vec{v}) = \mu(\vec{v}^l) (\dot{\boldsymbol{\epsilon}}^l + \dot{\boldsymbol{\epsilon}}^b) \quad (3.2.32)$$

Hence, we can write the Hessian functions as following:

$$H_{k\kappa}^{ll} = \int_{\Omega_h} \frac{\partial \mu(\vec{v}^l)}{\partial \vec{v}_\kappa^l} : \vec{\mathbb{B}}_k^l d\omega_h \quad (3.2.33a)$$

$$H_{k\kappa}^{lp} = - \int_{\Omega_h} N_\kappa \vec{\nabla} \cdot (\vec{\mathbb{B}}_k^l) d\omega_h \quad (3.2.33b)$$

$$H^{bb} = \int_{\Omega_h} \mu(\vec{v}^l) \vec{\mathbb{B}}^b : \vec{\mathbb{B}}^b d\omega_h \quad (3.2.33c)$$

$$H_{\kappa}^{bp} = - \int_{\Omega_h} N_{\kappa} \vec{\nabla} \cdot (\vec{B}^b) d\omega_h \quad (3.2.33d)$$

$$H_{k\kappa}^{pl} = - \int_{\Omega_h} N_{\kappa} tr(\vec{B}_k^l) d\omega_h \quad (3.2.33e)$$

$$H_k^{pb} = - \int_{\Omega_h} N_k tr(\vec{B}^b) d\omega_h \quad (3.2.33f)$$

$$H_{k\kappa}^{pp} = 0 \quad (3.2.33g)$$

It is to be noted that the bubble function for velocity interpolation was introduced in order to stabilize the mixed \vec{v}, p formulation. However, the system of equations in (3.2.33) can be condensed in order not to increase the degrees of freedom of the system. This is done by removing the following unknown incremental velocity at bubble node $\delta \vec{v}^{b(i)}$ from the system of equations:

$$\delta \vec{v}^b = - (H^{bb})^{-1} (R^b - H_{\kappa}^{bp} \delta p_{\kappa}) \quad (3.2.34a)$$

$$\begin{bmatrix} H_{k\kappa}^{ll,(i)} & H_{k\kappa}^{lp,(i)} \\ H_{k\kappa}^{pl,(i)} & -H_k^{pb,(i)} (H^{bb,(i)})^{-1} H_{\kappa}^{bp,(i)} \end{bmatrix} \begin{bmatrix} \delta \vec{v}_{\kappa}^{l,(i)} \\ \delta p_{\kappa}^{(i)} \end{bmatrix} = - \begin{bmatrix} R_k^{l,(i)} \\ R_k^{p,(i)} - H_k^{pb,(i)} (H^{bb,(i)})^{-1} R^{b(i)} \end{bmatrix} \quad (3.2.34b)$$

3.2.4 Thermal Problem

3.2.4.1 Energy conservation

In the steady-state, the time-independent thermal equilibrium problem is described (3.2.35) with c_p is the specific heat capacity, K_{cond} is the thermal conductivity and P_{pl} is the fraction of plastic power dissipated into heat.

$$\rho c_p \frac{dT}{dt} = \rho c_p \left(\underbrace{\frac{\partial T}{\partial t}}_{=0} + \vec{v} \cdot \vec{\nabla} T \right) = \vec{\nabla} \cdot K_{cond} \vec{\nabla} T + P_{pl} (\boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}}) \quad (3.2.35)$$

3.2.4.2 Initial conditions

In the steady-state formulation, the initial condition is represented by the steady-state temperature of the workpiece before metal forming. This known temperature field T_{imp} (constant or mapped from former history) is applied at the input plane Γ_{in} as a Dirichlet boundary condition for the resolution of the steady-state thermal equation.

$$T = T_{imp} \quad \text{on} \quad \Gamma_{in} \quad (3.2.36)$$

3.2.4.3 Boundary conditions

There are various ways heat is exchanged from the boundary of the domain. These heat exchanges are applied with the Fourier conduction model:

$$\vec{q} \cdot \vec{n} = -K_{cond} \vec{\nabla} T \cdot \vec{n} = \phi_{imp} \quad (3.2.37)$$

Symmetry and output planes

The symmetry and output planes are assumed to be adiabatic and flux $\phi_{imp} = 0$ is imposed on these surfaces.

At the contact surface

There are two main elements of energy exchange at the contact surface Γ_c : (i) Conduction of heat at the contact with tool, and (ii) partition of the heat generation due to friction between the workpiece and the tool. In the equation (3.2.38), h_{cond} is the thermal exchange coefficient between the workpiece and the tool, and b , b_m and b_{tool} are the respective averaged, workpiece and tool thermal effusivity.

$$\vec{q} \cdot \vec{n} = -K_{cond} \vec{\nabla} T \cdot \vec{n} = h_{cond} (T - T_{tool}) \quad (3.2.38a)$$

$$\phi_f = -\frac{b_m}{b_m + b_{tool}} (\vec{\tau}_f \cdot \Delta \vec{v}_t) \text{ with } b_m = \sqrt{\rho c_p K_{cond}} \quad (3.2.38b)$$

At the free-surface

The heat exchange at free-surface Γ takes place in two ways: (i) Convection and (ii) Radiation. In the equation, h_{conv} is the thermal exchange coefficient between workpiece and air, T_{ext} is the ambient temperature, σ_r is the Stefan's constant and ϵ_r is the emissivity.

$$\vec{q} \cdot \vec{n} = -K_{cond} \vec{\nabla} T \cdot \vec{n} = h_{conv} (T - T_{ext}) + \epsilon_r \sigma_r (T^4 - T_{ext}^4) \quad (3.2.39)$$

The convection and radiation terms can be combined to describe a unique thermal exchange coefficient h_{cr} which is computed from the Temperature known at the preceding iteration. With the fixed-point iterative solver, when the temperature field is converged, the h_{cr} is computed from the converged Temperature field.

$$\vec{q} \cdot \vec{n} = h_{cr} (T - T_{ext}) \quad (3.2.40a)$$

$$h_{cr} = h_{conv} + \epsilon_r \sigma_r (T + T_{ext}) (T^2 + T_{ext}^2) \quad (3.2.40b)$$

3.2.4.4 Weak form of the Thermal problem

The steady-state energy equilibrium equation is convection dominated, and hence the temperature field test function is discretized with SUPG finite elements. The strong form of the steady-state thermal equation (3.2.37) is written in weighted residual form (3.2.41d) and solved to compute the unknown temperature field T_h .

$$T_h = \sum_k^{nb_n} N_k T_k \quad T_h^* = \sum_k^{nb_n} N_k^{SUPG} T_k^* \quad (3.2.41a)$$

$$\mathbb{V}_h^T = \left\{ T_h \in C^0(\Omega_{x_h}) \text{ such that } \forall e \in \tau_h, T_h|_{\Omega_{x_h}^e} \in P1(\Omega_{x_h}^e) \quad \text{and } T = T_{imp} \text{ on } \Gamma_{in} \right\} \quad (3.2.41b)$$

$$\mathbb{V}_{h,0}^T = \left\{ T_h \in \mathbb{V}_h^T \text{ such that } T = 0 \text{ on } \Gamma_{in} \right\} \quad (3.2.41c)$$

$$\begin{aligned} \int_{\Omega_{x_h}} T_h^* (\rho c_p \vec{v}_h \cdot \vec{\nabla} T_h) d\omega + \int_{\Omega_{x_h}} K_{cond} (\vec{\nabla} T_h^* \cdot \vec{\nabla} T_h) d\omega + \int_{\Gamma_h} h_{cr} (T - T_{ext}) T^* ds \\ + \int_{\Gamma_{c,h}} h_{cond} (T - T_{tool} + \phi_f) T^* ds = \int_{\Omega_{x_h}} T_h^* P_{pl}(\boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}}) d\omega \end{aligned} \quad (3.2.41d)$$

3.2.5 Equivalent strain

3.2.5.1 Strong form

In the incremental formulation, the equivalent strain is computed with the integration of equation (3.2.5). The said equation is time-dependent, and at steady-state the time $t = \infty$. So the steady-state equation is given as (3.2.42a). At the input plane, the equivalent strain field is imposed as the

initial condition (3.2.42b) similar to the thermal problem.

$$\dot{\bar{\epsilon}} = \frac{d\bar{\epsilon}}{dt} = \underbrace{\frac{\partial \bar{\epsilon}}{\partial t}}_{=0} + \vec{v} \cdot \vec{\nabla} \bar{\epsilon} \quad \text{in } \Omega_x \quad (3.2.42a)$$

$$\bar{\epsilon} = \bar{\epsilon}_{imp} \quad \text{in } \Gamma_{in} \quad (3.2.42b)$$

The equation (3.2.42a) can be solved either with *streamline integration* or *global resolution* methods explained in the Section 2.3.5.2. With unstructured meshes, the latter seems a more practical choice.

3.2.5.2 Weak form

For the global resolution, the weighted residual form of the steady-state equivalent strain equation is written as following:

$$\int_{\Omega_x} (\vec{v} \cdot \vec{\nabla} \bar{\epsilon}) \bar{\epsilon}^* d\omega = \int_{\Omega_x} \dot{\bar{\epsilon}} \bar{\epsilon}^* d\omega \quad (3.2.43)$$

It can be remarked here that the state-variable is a P0 field which is constant in an element. Thus, for keeping the gradient definition meaningful in equation (3.2.43), a smoothed equivalent-strain field $\tilde{\epsilon}_k$ is used. The smoothing is done by averaging the P0 field at the mesh nodes.

$$\tilde{\epsilon}_h(\vec{x}) = \sum_{k=1}^{nbn} N_k(\vec{x}) \tilde{\epsilon}_k \quad (3.2.44)$$

The equations (3.2.43) denote convection dominant problem (large Peclet number), thus a SUPG test function (see equation (2.3.35)) is used for the stabilization of the finite element solution.

$$\sum_{e=1}^{nbe} \int_{\Omega_{xe}} N_k^{SUPG} (\vec{v} \cdot \vec{\nabla} \tilde{\epsilon}_h) d\omega_e = \sum_{e=1}^{nbe} \int_{\Omega_{xe}} N_k \dot{\bar{\epsilon}} d\omega_h \quad (3.2.45)$$

3.2.6 Resolution of the thermo-mechanical problem in the iterative algorithm

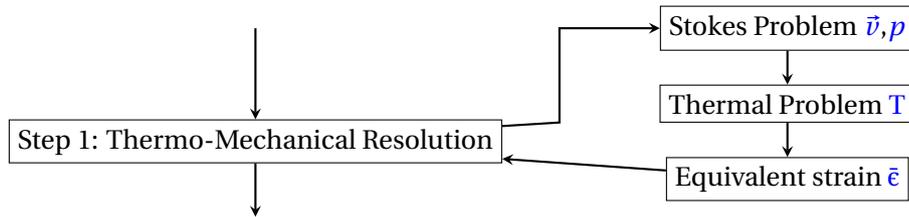


Figure 3.2.4: Details of the thermo-mechanical resolution in the iterative algorithm

The thermo-mechanical problem is weakly coupled as solved in the sequence shown in Figure 3.2.4. Firstly, the mechanical problem is solved from the system explained in equation (3.2.34b) to compute the unknown velocity and pressure field increments with the Newton-Raphson method. The convergence of the velocity field is sought before the resolution of the thermal problem from equation (3.2.41d). The third sub-step corresponds to the computation of equivalent strain from equation (3.2.45). It is to be impressed here that the thermo-mechanical problem is weakly coupled, hence the velocity field is not updated using the new material consistency \mathbf{K} found from equation (3.2.8) using the new equivalent-strain and temperature field. This update is done in the following fixed-point iteration.

3.3 Free-surface resolution formulation

Among the different possibilities for the free-surface resolution discussed in Section 2.3.5, ForgeNxt[®]2016 uses the global resolution method for finding the steady-state shape. The free-surface resolution aims to calculate the steady-state boundary shape by solving a velocity shape compatibility condition. The surface mesh nodes are corrected to conform to the new boundary surface, which also necessitates the regularization of the volume mesh elements to avoid their stretching. The present section is focused on the detailed description of the global resolution method for free surface computation.

3.3.1 Basic equations

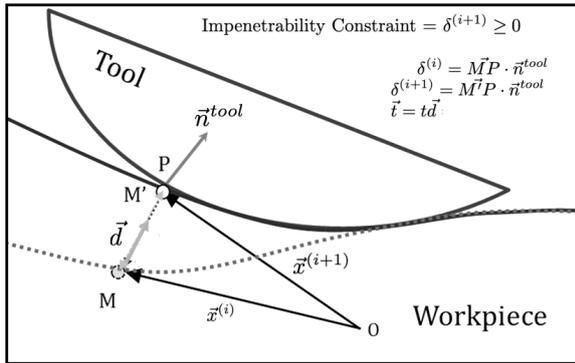
The free-surface boundary Γ is updated from the velocity field computed in the Mechanical resolution Step-1 by solving a velocity-shape compatibility condition. This condition is applied with the *global form* of the free-surface equation (2.3.32), and the unknown shape is computed using Finite element method. The global-form of free-surface equation (3.3.1a) finds the unknown shape $\vec{x}^{(i+1)}$ by enforcing the material flux across the boundary to be nullified. In the said equation, $\vec{x}^{(i+1)}$ represents the unknown shape in current configuration which is computed from the previous configuration $\vec{x}^{(i)}$ and the unknown correction vector \vec{t} as presented in equation (3.3.1b). The correction vector can be described with the scalar t in the direction \vec{d} in equation (3.3.1c). An important assumption in this step is that the velocity field is assumed to be constant as given in equation (3.3.1d). This assumption holds true due to the fixed-point choice for the algorithm and hence updated only in the Step-1. Thus, the free-surface of the steady-state domain is computed from the resolution of a scalar equation, which is trivial in terms of computation cost.

$$\vec{v}^{(i+1)} \cdot \vec{n}(\vec{x}^{(i+1)}) = 0 \quad \text{on } \Gamma \quad (3.3.1a)$$

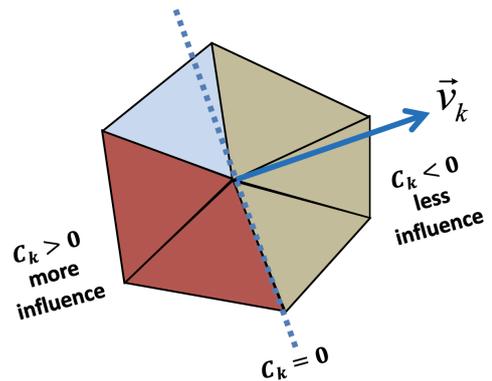
$$\vec{x}^{(i+1)} = \vec{x}^{(i)} + \vec{t}^{(i+1)} \quad \text{on } \Gamma \quad (3.3.1b)$$

$$\vec{t} = t\vec{d} \quad (3.3.1c)$$

$$\vec{v}(\vec{x}^{(i+1)}) = \vec{v}(\vec{x}^{(i)}) \quad (3.3.1d)$$



(a) Impenetrability constraint on the correction field \vec{t} at the contact surface



(b) Definition of C_k : weightage of elements with respect to position of elements with reference to node k

Figure 3.3.1: free-surface correction: Contact condition and C_k in SUPG test function

The free-surface correction is applied to the entire free-surface boundary Γ , however, in the contact region, the correction could result in the penetration of the workpiece material inside the contact surface. To avoid this, an impenetrability condition, represented with the unilateral constraint on the contact distance or the gap-function δ (on the current configuration), is applied

with equation (3.3.2). Here, δ represents the shortest normal distance between the node on the free-surface of the workpiece and the tool surface (see Figure 3.3.1a). In practice, this condition is applied with the constraint $\bar{h}(t)$ described on the previous configuration $\bar{\mathbf{x}}^{(i)}$ and correction vector $\bar{\mathbf{t}}$ as shown in equation (3.3.2b).

$$\delta(\bar{\mathbf{x}}^{(i+1)}) = \overrightarrow{M'P} \cdot \bar{\mathbf{n}}^{tool} \geq 0 \quad (3.3.2a)$$

$$\bar{h}(t) = t\bar{\mathbf{d}} \cdot \bar{\mathbf{n}}^{tool} - \delta(\bar{\mathbf{x}}^{(i)}) \leq 0 \quad (3.3.2b)$$

3.3.2 Weak formulation of free-surface equation

The strong form of free-surface equation (3.3.1) is written in the weighted residual form with residual S_k . Numerical experiments have shown that the free-surface equation is of convection nature, and destabilizes with standard Galerkin functions. So, a PG type test function is necessary for the free-surface computation. We can choose from a variety of PG test functions presented in the Section 2.3.5.2, but ForgeNxt[®]2016 uses SUPG test function as it results in accurate geometry prediction in comparison to the other functions presented. The SUPG function N_k^{SUPG} proposed by [1] provides an accurate multidimensional generalization of optimal one-dimensional upwind schemes. The SUPG method introduces an artificial diffusion only in the streamline flow direction, by virtue of which its stability is stronger than that of the Galerkin finite element method. The SUPG element can be described as a modification of the Galerkin element with a stabilization term in the rhs of (3.3.3b). This stabilization term permits a bias to the elements in the upwind direction from the reference of node k in comparison to the Galerkin hat function N_k . Here C_k is a cosine determining the angular location of a facet element from the point of reference of node k (see Figure 3.3.1b) such that $-1 \leq C_k \leq 1$ and α is a non-dimensional factor such that $0 < \alpha \leq 1$. Note that $\alpha = 0$ means that the stabilization term is zero. The standard value of α used for 1D element in literature [5, 8] is $\frac{1}{2}$. The material impenetrability condition (3.3.2b) is applied with the penalty method in node-to-facet form on the free-surface boundary. $\rho_c^{(2)}$ is the penalty coefficient for the impenetrability constraint. Here, s_k uses the same definition as described in equation (3.2.25b).

$$\forall k \in \Gamma_h, \quad S_k(\bar{\mathbf{v}}^{(i+1)}, t^{(i+1)}) = \int_{\Gamma_h} N_k^{SUPG}(\bar{\mathbf{v}} \cdot \bar{\mathbf{n}}(\bar{\mathbf{x}}^{(i+1)})) ds + \rho_c^{(2)} s_k [\bar{h}(t_k)]^+ \bar{\mathbf{d}}_k \cdot \bar{\mathbf{n}}_k^{tool} = 0 \quad (3.3.3a)$$

$$N_k^{SUPG} = N_k + \alpha C_k; \quad ; C_k = \frac{\bar{\nabla} N_k \cdot \bar{\mathbf{v}}}{|\bar{\nabla} N_k| |\bar{\mathbf{v}}|} \quad (3.3.3b)$$

3.3.3 Finite Element resolution of free-surface equation

The weak form of the free-surface correction equation (3.3.3a) is discretized with triangular shaped SUPG finite elements. The discretized residual is written in matrix form (3.3.4a). Here, m is the index of the node on any facet element $f \in \mathfrak{f}_k$, where \mathfrak{f}_k represents the element patch about node k . $\bar{\mathbf{n}}(\bar{\mathbf{x}}^{(i+1)})$ is the unit surface normal defined on the facets f in the current configuration. Here, $\bar{\mathbf{U}}_{km}$ in equation (3.3.4b) represents a mass-like matrix used for computing the normal velocity at node k .

$$\forall k \in \Gamma_h, \quad \int_{\Gamma_h} N_k^{SUPG}(\bar{\mathbf{v}} \cdot \bar{\mathbf{n}}(\bar{\mathbf{x}}^{(i+1)})) ds = \sum_{m \in \mathfrak{f}_k} \bar{\mathbf{v}}_m \cdot \bar{\mathbf{U}}_{km} \quad (3.3.4a)$$

$$\bar{\mathbf{U}}_{km} = \int_{\Gamma_h} N_k^{SUPG} N_m(\bar{\mathbf{n}}(\bar{\mathbf{x}}^{(i+1)})) ds \quad (3.3.4b)$$

The discretized system is solved with the Newton's iterative resolution method to compute the unknown correction field t at the free-surface boundary to find the updated shape $\bar{\mathbf{x}}^{(i+1)}$ from equations (3.3.1b) and (3.3.1c).

3.3.4 Extension of the free-surface correction to multi-DoF (with edges)

The description of scalar form of free-surface correction equation (3.3.1) is efficient as long as the geometry of the workpiece does not have any topological singularities like edges or corners. In the case with singularities, two different approaches are used: (i) **Free-surface correction with surface regularization (SUPG-reg)**, in which the surface mesh is regularized to follow the singularity (ii) **SUPG-dif**, in which a vectorial form of the free-surface equation is envisaged.

3.3.4.1 SUPG-reg

In the **SUPG-reg** technique, the free-surface boundary mesh is regularized with the Laplacian of the correction field t computed from equation (3.3.5a). Hence, the technique takes into account several degrees of freedom or the singularities of the mesh without any particular treatment.

$$\Delta t = 0 \quad \text{on} \quad \Gamma \quad (3.3.5a)$$

$$\mathbf{H}^{(i)} + w_r \mathbf{H}^{r(i)} \delta \vec{t}^{(i)} = -r^{(i)} \quad (3.3.5b)$$

The index r represents the regularization term with H_r is the updated Hessian for the resolution of correction t with the iterative method. The surface mesh is regularized simultaneously with the iterative resolution of the correction field, but this regularization proves to be antagonistic to the free-surface correction. In order not to constrain the displacements too much, only the matrix of the system coming from the Newton-Raphson resolution is regularized (3.3.5b). However, the weighting coefficient w_r of the regularization is difficult to calibrate and the speed of convergence of the free-surface calculation is extremely sensitive. Among the various methods tested by [7], the original least squares method converges slowly such that the solution obtained after a fixed number of iterations is not always correct. The **LS-SUPG** method on the other hand is more robust, and yields good results.

3.3.4.2 SUPG-dif

In the **SUPG-dif** formulation, differential processing is done for the edge nodes and those of the plane surfaces. For edges, two degrees of freedom are needed, hence, a vectorial (2-DoF) form of the free-surface equation for cases is proposed. It aims at finding the corrections $t^{[j]}$ in the respective linearly independent directions $\vec{d}^{[j]}$ defined at an edge of the free-surface Γ_{edge} (see Figure 3.3.2a).

$$\vec{x}^{(i+1)} = \vec{x}^{(i)} + \sum_{j=1,2} t^{[j]} \vec{d}^{[j]} \quad (3.3.6)$$

The Least Square method is easily extensible to multiple-DoF [7] as the derivative of the normal results in an independent residual in each direction that computes the scalar corrections in the respective direction \vec{d} . On the other hand, the **SUPG** shape function results in only a scalar equation to be resolved. In order to find a multi-DoF formulation, we assume linear independence of the directions $\vec{d}^{[j]}$, with $j = 1, 2$ for the edge singularities. Then, the residual is independently written in each direction. From this assumption, we obtain the vectorial form of the free-surface equation residual which is applied on the free-surface edge boundary Γ_{edge} .

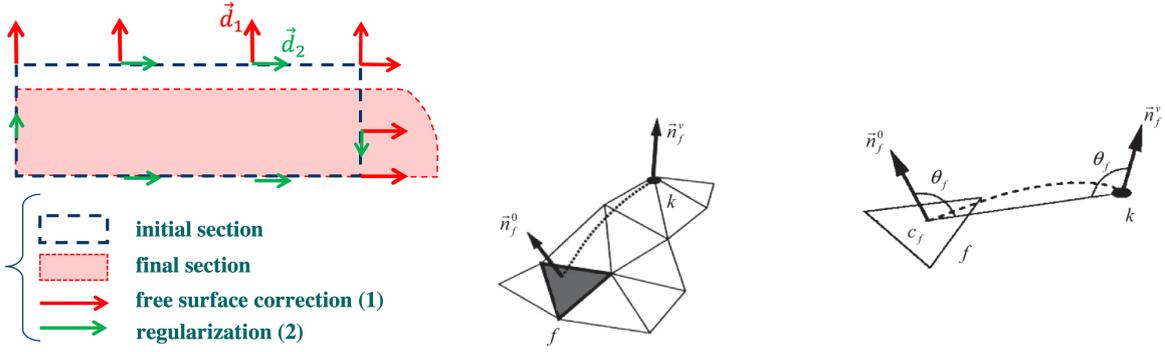
$$\forall k \in \Gamma_{edge}, \quad S_k^{[j]}(\vec{v}, t^{[j]}) = \int_{\Gamma_{edge}^{[j]}} \mathbf{N}_k^{\text{SUPG}}(\vec{v} \cdot \vec{n}^{[j]}(\vec{x}^{(i+1)})) ds + \rho_c^{(2)} s_k [\bar{h}(t_k)]^+ \vec{d}^{[j]} \cdot \vec{n}_k^{tool} = 0 \quad (3.3.7)$$

The proposed method works perfectly well and is much more robust than **SUPG-reg**, but the free-surface correction at the edges results in the stretching of the corresponding elements in its support. So a regularization (see Figure 3.3.2a) of the surface mesh is adopted in the tangential direction perpendicular to the correction and the rolling direction. A Laplacian operator is introduced in the correction residual (weak form) for the correction t in the lateral direction \vec{e}_2 , which restricts

the stretching of element shape.

$$\forall k \in \Gamma_h, \quad S_k^b(t) = \int_{\Gamma_h} \vec{\nabla} N_k \cdot \vec{\nabla} (\vec{t} \cdot \vec{e}_2) ds = 0 \quad (3.3.8)$$

SUPG-dif is envisaged only on the edge geometry, whereas on the rest of the domain, only one degree of freedom is enough and is optimal. However, to keep a homogeneous number of unknowns per node, the 2-DoF formulation is used for the entire free-surface boundary Γ and not just the edge Γ_{edge} . The second dof of the free-surface mesh is used for the mesh regularization in the \vec{e}_2 direction.



(a) Choice of directions \vec{d}_1 and \vec{d}_2 in the boundary of domain Ω

(b) Detection of topological singularities (edges/corners) on the surface mesh using normal "voting" method

Figure 3.3.2: Vectorial form of free-surface computation with **SUPG-dif** formulation

3.3.5 Edge Detection

For the detection of geometric singularities, like edge or corners, the normal voting method [6] for piece-wise smooth surfaces is used. The proposed method uses information from the group of surface elements in the geodesic neighborhood of a node k to find out mainly two characteristics of a node (i) the orientation (ii) the curvature of the surface. These characteristics could be used to find out if the said node lies on a smooth surface or a discontinuity. In an element patch \mathcal{J}_k , each surface element f with normal \vec{n}_f^0 yields a voted normal \vec{n}_f^v at node k from the translation about a surface curve. This surface curve is approached by the geodesic distance, defined by the angle θ_f is expressed as:

$$\forall f \in \mathcal{J}_k, \quad \vec{n}_f^v = \vec{n}_f^0 + 2\cos(\theta_f) \frac{S_f}{S_{max}} \frac{\vec{X}_k c_f}{\|\vec{X}_k c_f\|}, \quad \text{with } \cos(\theta_f) = \frac{\vec{n}_f^0 \cdot \vec{X}_k c_f}{\|\vec{X}_k c_f\|} \quad (3.3.9)$$

In this expression, c_f is the centroid of facet f , \vec{X}_k is the coordinates of node k , S_f is surface area of f , and S_{max} is the maximum surface area of facet on patch \mathcal{J}_k . Using this voted normal, a covariance matrix is constructed which is used to find the principal directions (eigen vectors) at the node k from the eigen values of this matrix. The magnitude of each of the principal eigen values (1, 2, or 3) are tested on a geometric criteria to classify whether the given free-surface node k belongs to a pure surface, an edge or a corner. The geometric criteria chosen for this classification are taken from [4].

3.3.6 Volume Mesh Regularization and Mesh Update

In the previous sections 3.3.1-3.3.5, we developed the methodology to compute the steady-state shape position vectors $\vec{x}^{(i+1)}$ numerically, but only on the boundary Γ of the workpiece domain Ω_x . This boundary correction results in the stretching of the volume tetrahedral elements due to the boundary update (see Figure 3.3.3). In order to avoid this stretching, a volume mesh regularization

is required (not to be confused with surface mesh regularization in the equation (3.3.8)). This is achieved by solving the Laplacian of the correction t on the discretized domain with equation (3.3.10a). The free-surface correction field \vec{t}_{comp} known on the boundary Γ is used as the Dirichlet condition to the Laplacian problem. Hence, the mesh coordinates $\vec{x}^{(i+1)}$ are updated with the global correction field using equation (3.3.6).

$$\Delta t_k^{(i+1)} = 0 \quad \text{in} \quad \Omega_h^{(i)} \quad (3.3.10a)$$

$$t_k = t_{comp} \quad \text{in} \quad \Gamma_h \quad (3.3.10b)$$

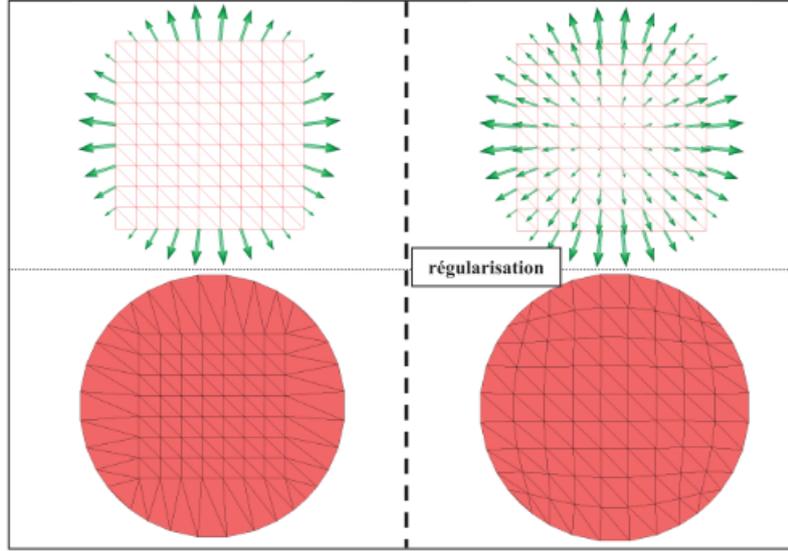


Figure 3.3.3: Regularisation of the volume mesh [7]

3.3.7 Free-surface Resolution in the Iterative Algorithm

The free-surface resolution entails finding the steady-state domain at the boundary and then regularizing the internal volume nodes. Thus it corresponds to the three steps as shown in the Figure 3.3.4.

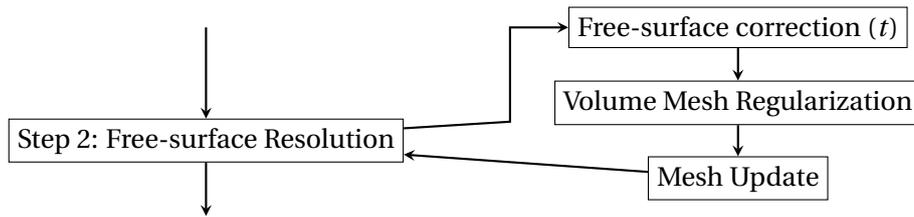


Figure 3.3.4: Details of the free-surface resolution step in the iterative algorithm

3.4 Contact treatment in a steady-state frame

3.4.1 Contact coupling

A decoupled, fixed-point iterative algorithm is chosen to solve the strong coupling in velocity and shape of the steady-state forming problem. The chosen algorithm ensures flexibility from the point of view of numerical implementation. The unilateral contact constraints need to be satisfied with both the velocity (3.2.14) and the free-surface shape (3.3.2b). However, the contact surface Γ_c on which these constraints must be applied is unknown. Taking advantage of the fixed-point algorithm, even the contact surface is chosen as a fixed-point variable which is assumed at

initialization from (3.4.1) and gradually improved until the algorithm converges. Given the strong coupling of the formulation, the contact surface is critical to the convergence of the steady-state algorithm.

$$\Gamma_c^{(1)} = \left\{ \forall k \in \Gamma, \quad \delta_k \left(\bar{x}^{(i=0)} \right) < 0 \right\} \quad (3.4.1)$$

The indices (1) and (2) correspond to the **Step-1** and the **Step-2** of the fixed-point iterative algorithm respectively. The velocity is computed from equation (3.2.26a) with the contact constraint (3.2.14) applied on the contact surface $\Gamma_c^{(1)}$. However, after the mechanical resolution, the contact surface $\Gamma_c^{(2)}$ is updated using the Lagrange multiplier λ_k from equation (3.4.2b), which is an approximation of the normal contact stress from the velocity field.

$$\lambda_k = -\rho_c^{(1)} s_k [\bar{v}_k \cdot \bar{n}_k]^+ \quad (3.4.2a)$$

$$\Gamma_c^{(2)} = \{ \forall k \in \Gamma, \lambda_k < 0 \} \quad (3.4.2b)$$

[8] remarked that if a unilateral contact condition (3.3.2b) is applied at the entire free-surface, it results in complete loss of contact surface. Therefore, it is proposed to apply only a bilateral condition (in normal direction) on contact surface $\Gamma_c^{(2)}$ from (3.4.2b) and unilateral condition only on the free-surface $\Gamma \setminus \Gamma_c^{(2)}$ which is not in contact as represented in equation (3.4.3).

$$\begin{aligned} \forall k \in \Gamma_h, \quad S_k \left(\bar{v}^{(i+1)}, t^{(i+1)} \right) = \int_{\Gamma_h} N_k^{\text{SUPG}} \left(\bar{v} \cdot \bar{n} \left(\bar{x}^{(i+1)} \right) \right) ds + \rho_c^{(2)} s_k \mathbb{1}_{(k \in \Gamma_c^{(2)})} (\bar{h}(t_k)) \bar{d}_k \cdot \bar{n}_k^{\text{tool}} \\ + \rho_c^{(2)} s_k \mathbb{1}_{(k \in \Gamma \setminus \Gamma_c^{(2)})} [\bar{h}(t_k)]^+ \bar{d}_k \cdot \bar{n}_k^{\text{tool}} = 0 \end{aligned} \quad (3.4.3)$$

After the free-surface computation, the contact surface $\Gamma_c^{(1)}$ is again updated with another Lagrange multiplier μ_k from (3.4.4b) using the condition (3.4.4a). Here, ϵ is a small positive coefficient, introduced for numerical pseudo-adhesion for the definition of contact surface and is empirically computed as 2% of the mesh size.

$$\mu_k = -\rho_c^{(2)} \bar{h}(t_k) \quad (3.4.4a)$$

$$\Gamma_c^{(1)} = \{ \forall k \in \Gamma, \mu_k < \epsilon \} \quad (3.4.4b)$$

The contact surface $\Gamma_c^{(1)}$ is where the conditions (3.2.14) and (3.2.15) are applied in the **Step-1** for velocity computation. At the convergence of fixed-point iterative loop, the final contact surface is found as:

$$\Gamma_c = \Gamma_c^{(1)} \cap \Gamma_c^{(2)} \quad (3.4.5)$$

3.4.2 Summary of the ForgeNxt[®]2016 Fixed-point resolution algorithm

A general form of the iterative resolution algorithm for the steady-state computation of the rolling problem was presented in Figure 3.1.1. The detailed substeps in the respective thermo-mechanical and free-surface resolution steps were presented in Figure 3.2.4 and Figure 3.3.4. The detailed complete algorithm, with the contact coupling described in Section 3.4.1, is presented in Figure 3.4.1. The convergence of the velocity field, tool force, temperature, equivalent strain, and geometry need to be assured for the convergence of the fixed-point algorithm. Using this algorithm, the contact area is reported to be correctly simulated. It is important to mention that a relaxation of the free-surface correction is used during the first iterations of the simulation. It consists of keeping only half of the values of the correction calculated to correct the domain. Thus, the nodes are gradually detached without showing strong discontinuities upstream of the contact zone. For the steady-state resolution, the convergence criteria for (i) Geometry t_{conv} (ii) Temperature T_{conv} (iii)

Equivalent strain $\bar{\epsilon}_{conv}$ (iv) Tool Force F_{conv} are defined as following:

$$t_{conv} = \forall k \in \Gamma_{out}, \quad \max_k \frac{t_k}{h_k^c} \leq \epsilon_{geom} \approx 0.05 \quad (3.4.6a)$$

$$T_{conv} = \forall e \in \Omega_h, \quad \frac{\max_e |T_e^{(i+1)} - T_e^{(i)}|}{\max_e (T_e^{(i+1)}) - \min_e (T_e^{(i+1)})} \leq \epsilon_T \approx 0.1 \quad (3.4.6b)$$

$$\bar{\epsilon}_{conv} = \forall e \in \Omega_h, \quad \frac{\max_e |\bar{\epsilon}_e^{(i+1)} - \bar{\epsilon}_e^{(i)}|}{\max_e (\bar{\epsilon}_e^{(i+1)}) - \min_e (\bar{\epsilon}_e^{(i+1)})} \leq \epsilon_\epsilon \approx 0.1 \quad (3.4.6c)$$

$$F_{conv} = 2 \times \frac{|F_{tool}^{(i+1)} - F_{tool}^{(i)}|}{F_{tool}^{(i+1)} + F_{tool}^{(i)}} \leq \epsilon_F \approx 0.01 \quad (3.4.6d)$$

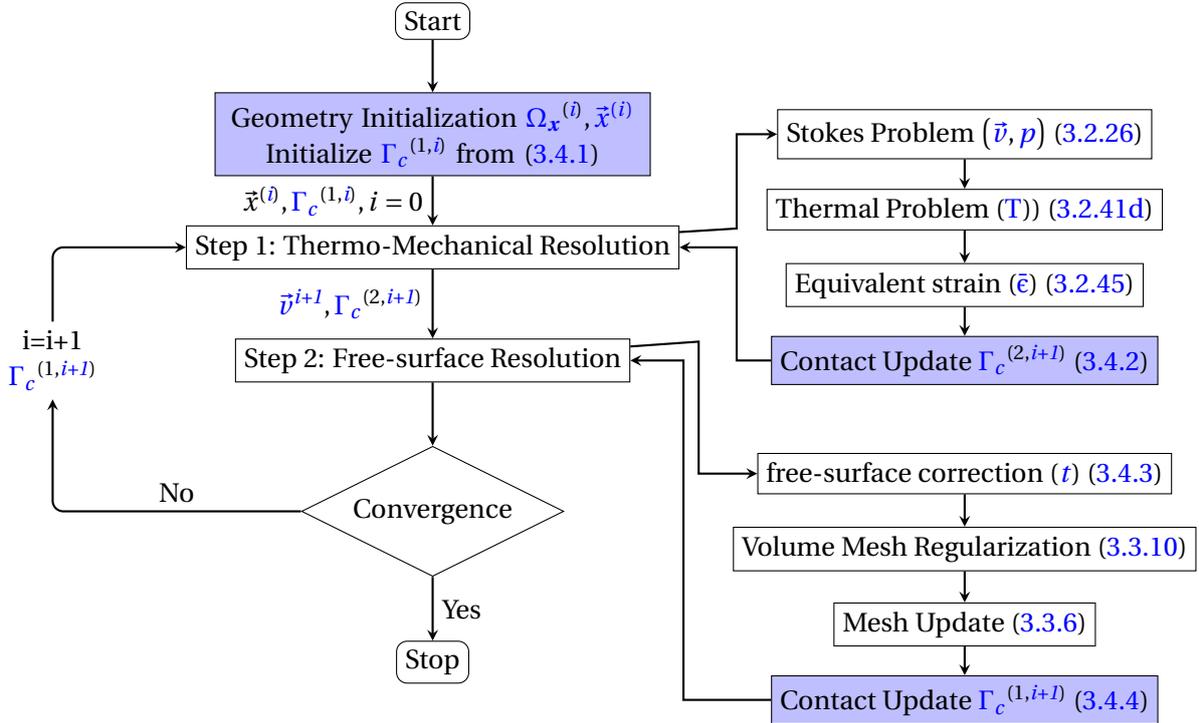


Figure 3.4.1: Complete ForgeNxt[®] 2016 fixed-point algorithm with contact coupling

It can be noticed that the convergence of the unknown fields of the steady-state formulation is ensured with the strict infinite norm, which aims to keep the maximum difference between two iterations below the defined criterion for each variable.

3.5 Application to hot-rolling test-cases

A large number of relatively different test-cases were envisaged for the **Forge Stationnaire Bis** project. They correspond to the simulation of steady-state forming processes (mainly rolling) proposed by a consortium of industries interested in reducing the computation time for simulation. To understand better the underlying problems with the existing steady-state formulation presented in Section 3.4.2, the same is tested on some of these test-cases. We focus on two of the most representative cases: thick sheet and shape rolling with a high reduction rate. The first test-case is a flat sheet subjected to hot rolling under a pair of symmetric rigid, flat rolls. The principle attributes of this test are a uniform contact surface with the tool in the transverse direction and no large changes in shape after rolling. The second test-case is a shape rolling problem with an oval section workpiece transformed into a square section after hot rolling. In both the test-cases, the

respective workpiece has two symmetry planes, so, only one-fourth of the domain is modeled. The steady-state solution from the ForgeNxt[®]2016 is validated with the steady-state solution obtained from the time dependent solution with ForgeNxt[®]inc using similar mesh size. The geometry for the steady-state resolution is initialized using the *extrusion method* as shown in 3.2.2a. Following rheology and tribology properties are used for both the test cases:

S.No.	Rheological Parameter	Value
1	K	30
2	m	0.15
3	α_f	0.3
4	p	0.15

Table 3.5.1: Rheological and tribological properties of the test-cases

A constant initial temperature of 250 °C is imposed for the workpiece and 150 °C is imposed for the rolls. *It is to be noted here that the primary objective of this study is to understand a problem related to the contact in the mechanical resolution, which perturbs the thermal as well as the mechanical solution. Hence, it is important to remove the thermal computation from the problem to isolate the problem due to the thermal coupling. Hence, adiabatic interface conditions are applied in order to prevent the heat exchange with the die (heat transfer coefficient $h_{cond} = 0.0$, thermal effusivity $b = 0.0$), with the surroundings (heat transfer coefficient $h_{conv} = 0.0$) and within the workpiece (Thermal conductivity $K_{cond} = 0.0$).* For the free-surface computation, the two test functions, namely (i) LS-SUPG (equation (2.3.34)) and (ii) SUPG (equation (3.3.3b)), as used by [7], have been tested. The SUPG-dif formulation along with mesh regularization in the \tilde{e}_2 direction is used, which was concluded (by [7]) to be the best in terms of resolution time and robustness. Each of the simulations is run on 12 cores of Intel(R) Xeon(R) CPU E5-1650 v3 @ 3.5GHz processor. For the validation of the steady-state results, two main comparisons are made for each of the test-case with the two free surface formulations. Firstly, the global results like tool force, torque, computation time, the number of iterations for the resolution and the deformed shape of the workpiece are compared. Next, the solution contours like equivalent stress, equivalent strain along with velocity/contact are compared. In addition, the convergence of the three of the four convergence parameters, viz. (i) Geometry t_{conv} (ii) Equivalent strain $\bar{\epsilon}_{conv}$ (iii) Tool Force F_{conv} , are also presented. These results help to explain the stability of the solution and hence compare different steady-state formulations. Another criteria chosen to compare the steady-state formulations is the loss of flux or the loss of material volume. This loss must ideally be zero because of the isochoric conditions applicable with the plastic deformation in the formulation, however may differ due to the assumptions in the free surface shape computation. The flux loss is measured as the difference between the flow of material into and out of the control volume Ω_x and given with the following relation (3.5.1).

$$\text{flux loss function} = \frac{\sum_{f \in \Gamma_{in}} \int_f \vec{v}_f \cdot \vec{n}_f ds_f + \sum_{f' \in \Gamma_{out}} \int_{f'} \vec{v}_{f'} \cdot \vec{n}_{f'} ds_{f'}}{\sum_{f \in \Gamma_{in}} \int_f \vec{v}_f \cdot \vec{n}_f ds_f} \times 100 \quad (3.5.1)$$

3.5.1 Flat rolling problem: VP-Test-Case-01

3.5.1.1 Model Details

The VP-Test-Case-01 consists of a rectangular section (80 mm x 25 mm) workpiece and is deformed under a pair of flat rolls (see Figure 3.5.1) without any constraint of material flow on the lateral side. This test case is proposed by Constellium. The rolls are rigid, with diameter 600 mm, and rotate at 27.5 rpm. After the rolling operation, the thickness of the plate is reduced to 18 mm, and the section does not remain rectangular due to the lateral spread phenomenon.

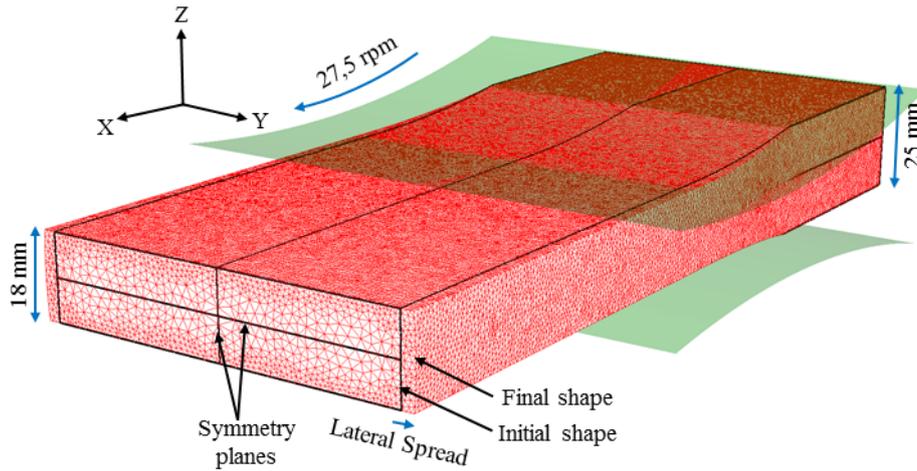


Figure 3.5.1: Simulation model for VP-Test-Case-01 from Constellium

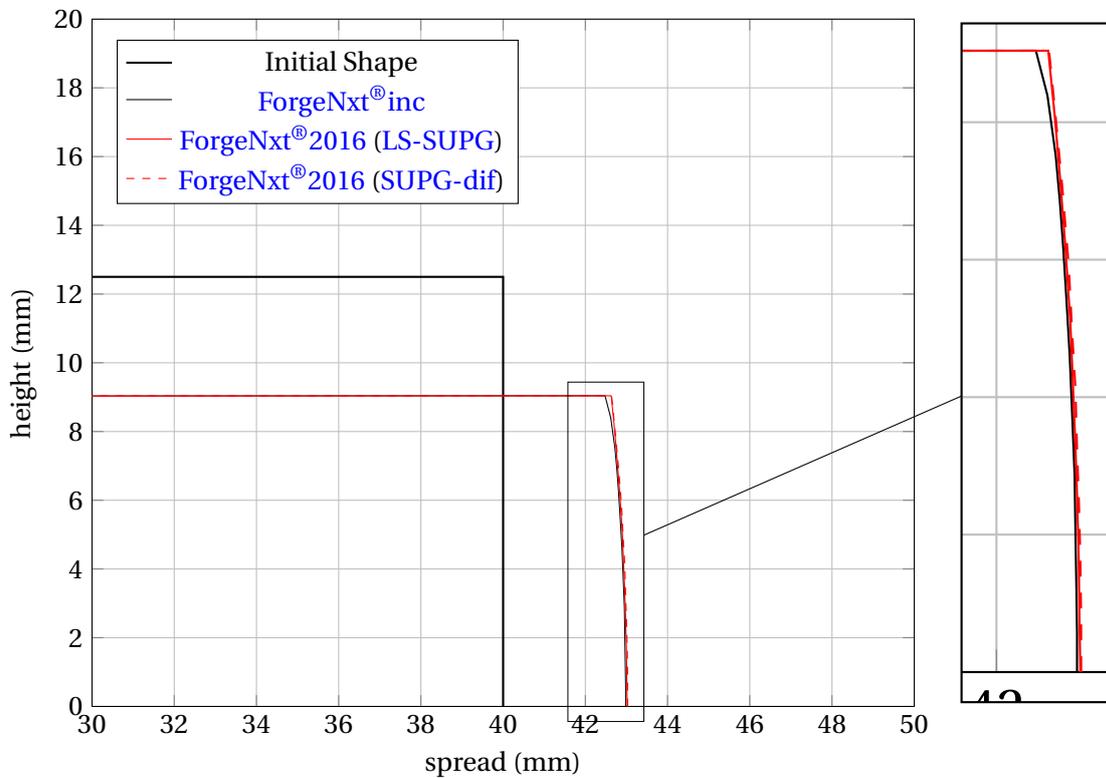


Figure 3.5.2: Comparison of deformed section (steady-state) shape of the VP-Test-Case-01 with the steady-state and incremental formulations

3.5.1.2 Comparison of solution

The global results for the VP-Test-Case-01 are presented in Table 3.5.2. The respective deformed (steady-state) sections measured at the output plane Γ_{out} (for the steady-state solution) are compared in Figure 3.5.2. The deformed section for the incremental solution is taken far enough from the tool, such that the velocity field, the state-variables, and the deformations are constant in the rolling direction. Even for the existing steady-state formulation ForgeNxt® 2016, certain end effects are present and need improvement of the boundary condition at the output plane Γ_{out} . So, the section results are from an axial location (rolling direction) few elements before the output plane. It can be seen that the ForgeNxt® 2016 algorithm finds the steady-state solution in 5 iterations with both the free surface formulations. The steady state solver predicts both the tool force and the

torque from the incremental solution under a margin of 2%. The deformed section computed with the steady-state algorithm also has a very good agreement with that found from the incremental resolution. There is a minor over-prediction of the lateral spread at the edge with both steady-state simulations, whereas, at the symmetry plane, the spread has a very good agreement with the ForgeNxt[®]inc solution. Since this observation is idem with both the free surface formulations, it is less likely that the SUPG-dif formulation by [7] is the reason for this difference in the lateral spread. This phenomenon could be linked to the contact formulation which may result in larger lateral velocities and hence result in larger spread calculation with both free surface formulations.

Shape Correction method	Conv. inc/iters	Comp. Time (min)	Tool Force (T)	Tool Torque (kN-m)
ForgeNxt [®] 2016 (LS-SUPG)	5	3.9	17.14	4.12
ForgeNxt [®] 2016 (SUPG-dif)	5	5.1	17.2	4.13
ForgeNxt [®] inc	541	300	17.6	4.26

Table 3.5.2: Global performance of the VP-Test-Case-01 with the ForgeNxt[®]2016 algorithm in comparison to the ForgeNxt[®]inc solution (on 12 cores)

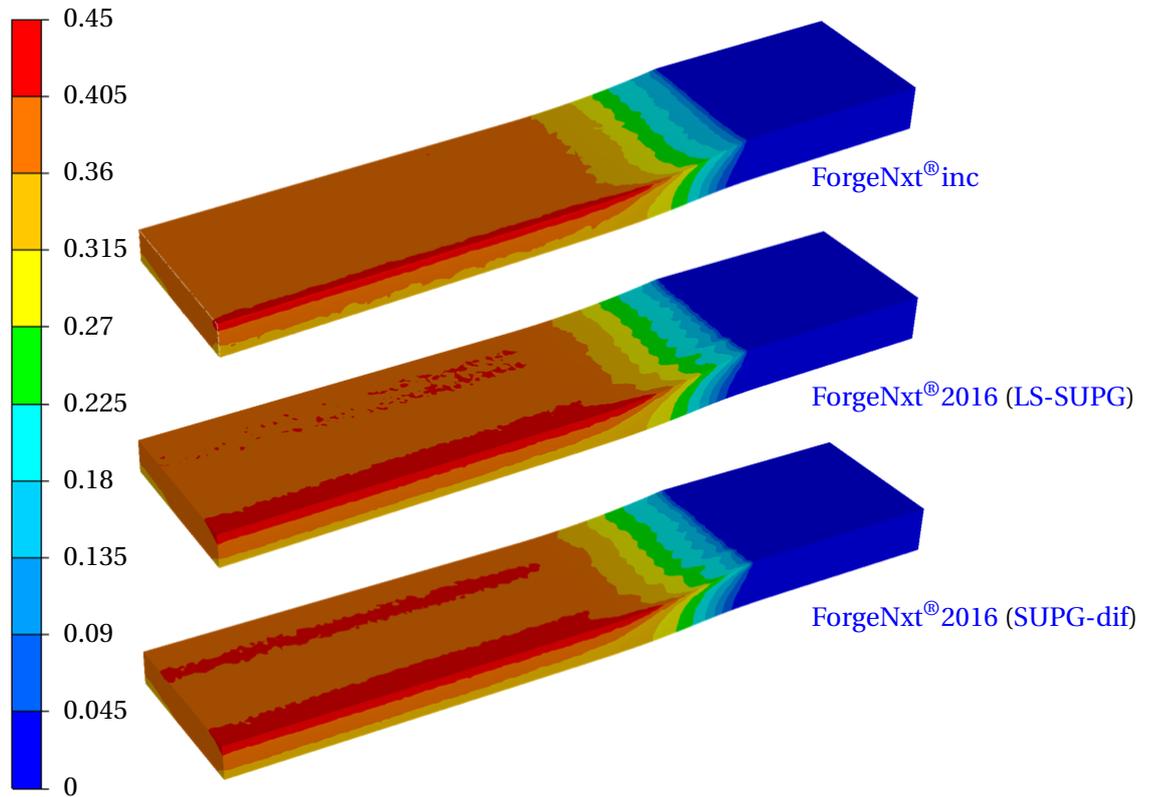


Figure 3.5.3: Comparison of equivalent strain for VP-Test-Case-01

The most attractive part of the obtained results is the number of iterations for the resolution of the test-case with the steady-state solver, which are 108 times lesser than the incremental formulation. Also, the computation time is reduced by 60-75 times with the steady-state resolution. The solution contours for equivalent strain $\bar{\epsilon}$ and the von Mises (equivalent) stress $\bar{\sigma}$ with the incremental and steady-state formulations are presented in Figure 3.5.3 & Figure 3.5.4 respectively. Globally the result contours seem to be predicted well with the steady-state formulations. However, if observed carefully, large undulations of deformation below the tool contact is observed with the steady-state formulations. This is an indication of an unstable contact behavior. These

instabilities are also seen to be convected downstream and seen as bubbles of large strains on the free surface boundary. These oscillations are also observed in the equivalent stress but only under the tool as the material reaches stress-free state as soon as it exits the tool. These oscillations are observed with both the free surface formulations tested, thus indicating an alternative phenomenon than the free surface computation method resulting in unstable solution. The convergence parameters for Force, equivalent strain and correction are presented in the Figure 3.5.5a, Figure 3.5.5b and Figure 3.5.6a respectively. In this exercise, we look to observe the stability of the solution if it is continued to be iterated until the maximum number of iterations are reached by choosing a very small geometry convergence criterion (for example $\epsilon_{geom} = 10^{-5}$). The stability test would only be met if all these parameters do not diverge as the computation progresses.

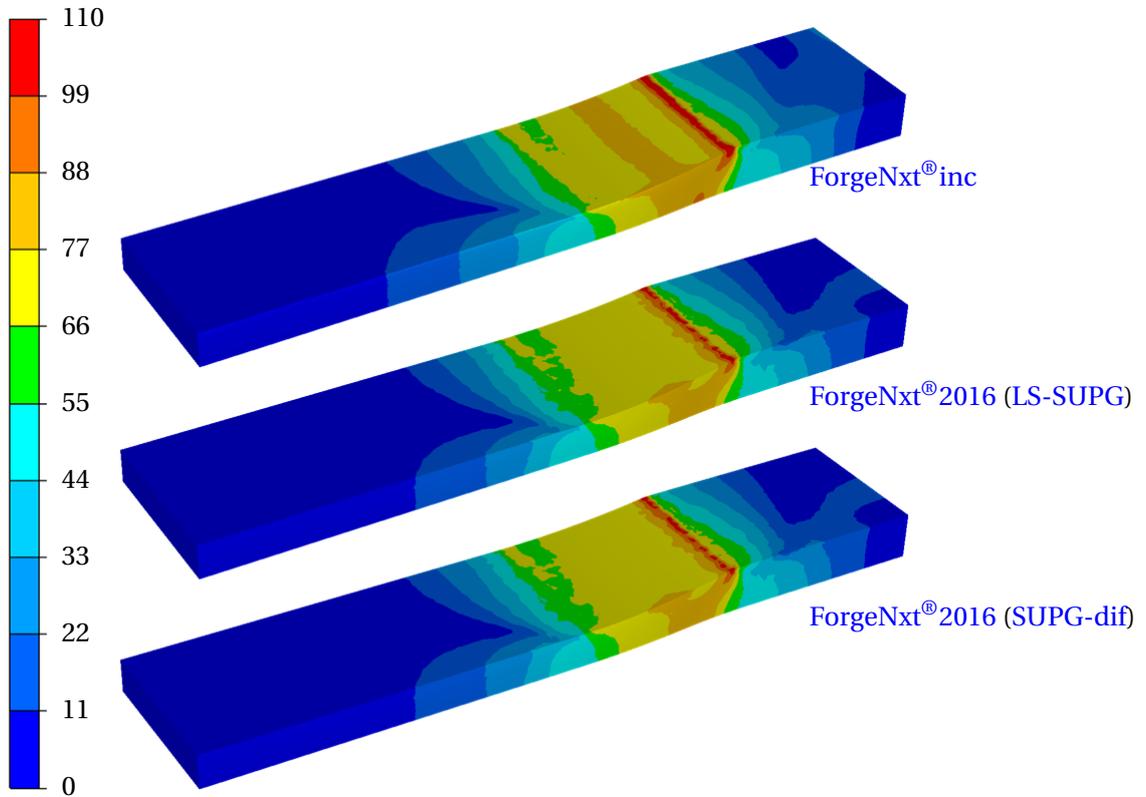


Figure 3.5.4: Comparison of von Mises stress (in MPa) for VP-Test-Case-01

It can be seen that all the tested parameters meet their respected convergence criteria in the same number of iterations with both the steady-state formulations. However, if we look closely at each of the parameters, generally we observe better convergence with the SUPG-dif formulation as the magnitude of the difference between consecutive iterations is much smaller. In other words, for all the convergence parameters, the LS-SUPG formulation results in larger changes in force, max. equivalent strain and max correction at the exit plane, between two iterations. The flux loss with the two formulations is compared in Figure 3.5.6b. This loss is nearly zero with the SUPG-dif formulation, but around 0.05% with the LS-SUPG formulation, which is insignificant from an industrial viewpoint.

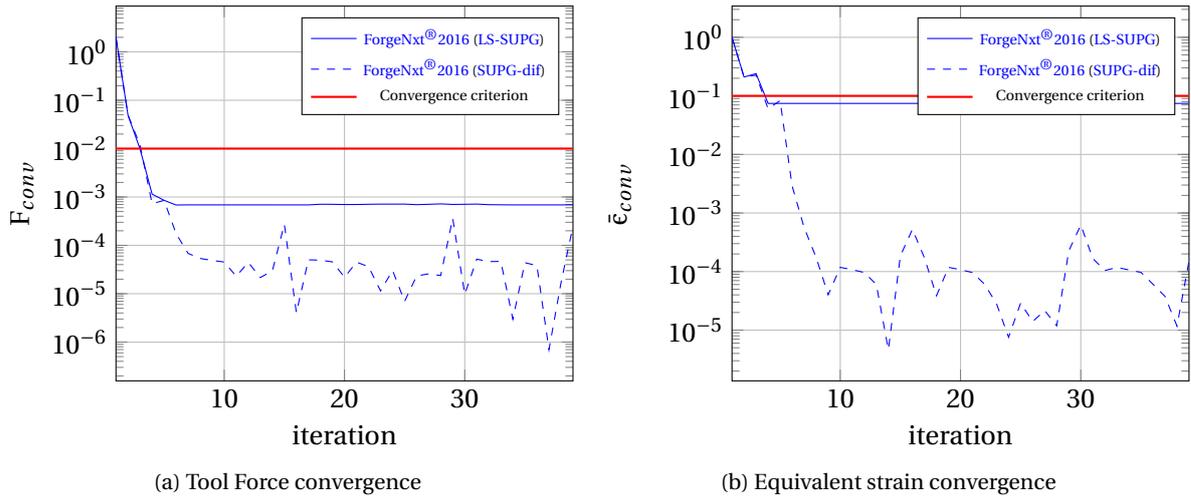


Figure 3.5.5: Comparison of Tool Force and Equivalent strain convergence for **VP-Test-Case-01** with the **ForgeNxt®2016** formulation

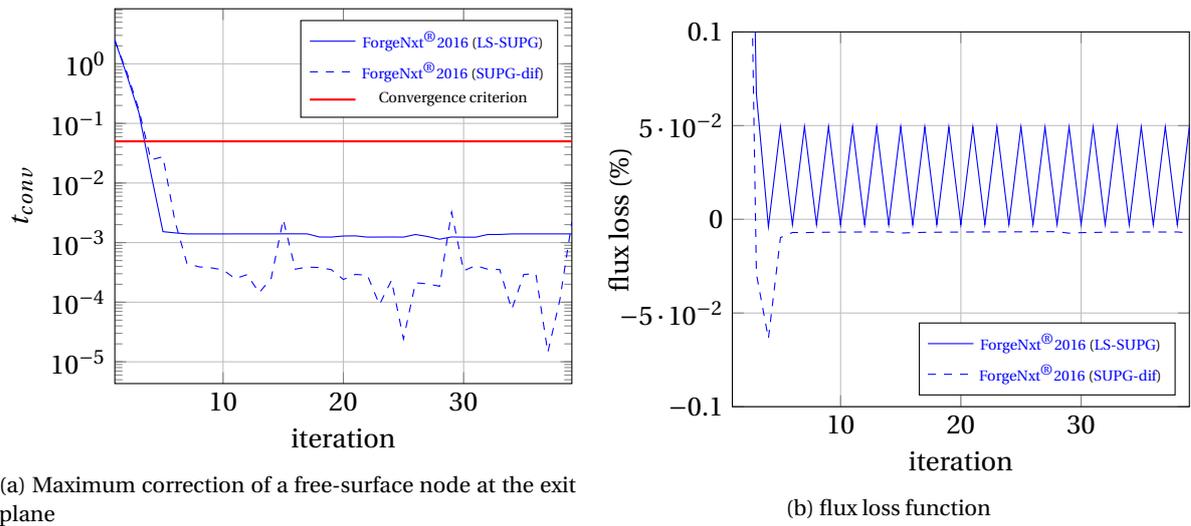


Figure 3.5.6: Comparison of free-surface correction convergence and flux loss for **VP-Test-Case-01** with the **ForgeNxt®2016** formulation

3.5.2 Shape rolling problem: **VP-Test-Case-02**

3.5.2.1 Model Details

In this test-case **VP-Test-Case-02**, an oval section workpiece is rolled under concave dies to a square shape (see Figure 3.5.7). The principal axes of the oval cross-section of the workpiece measure 108 mm x 36 mm. The rolling reduction and the lateral constraint (due to a not flat roll) brings the maximum height and the width to 56 mm with filleted corners. These rolls have a 600 mm diameter, are separated by 3.5 mm and have a rotational velocity of 48 rpm. Compared to the previously presented, this test-case observes a large shape change. The workpiece is engaged between two non flat tools that results in an oval shaped curved contact, and hence only a few nodes of the workpiece come in contact with the tool in the roll-bite.

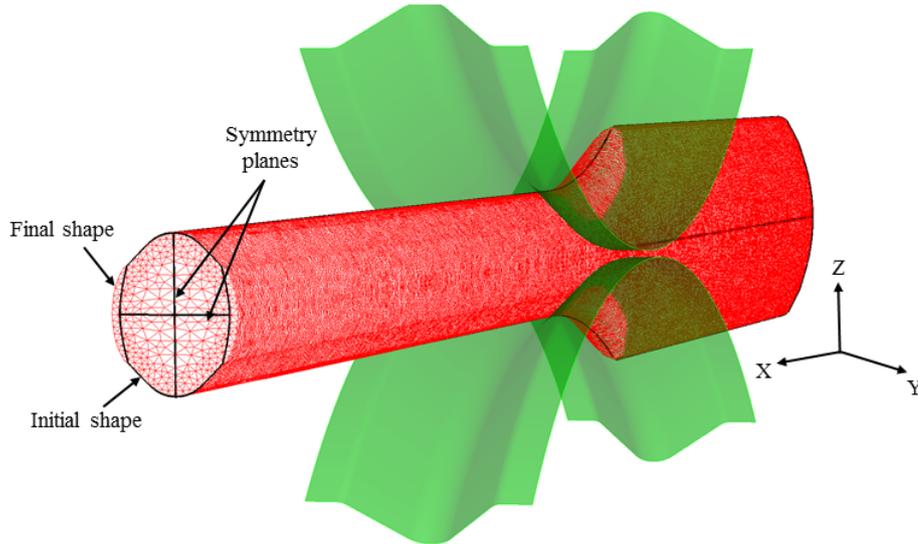


Figure 3.5.7: *VP-Test-Case-02* from Aubert & Duval tested with the steady-state algorithm

3.5.3 Comparison of solution

It is observed that the steady-state algorithm, the problem solution is approached but fails to reach to a stable solution. This is observed with both the free surface formulations *LS-SUPG* and *SUPG-dif* formulations. The steady-state iterative algorithm looks to find a solution with parameters converged, but is not successful. However, it is to be impressed that the unstable solution found with the steady-state resolution is not far from the incremental solution. The prediction of tool force and torque (see Table 3.5.3) is accurate in comparison to the *ForgeNxt[®]inc* solution. However, minor oscillations are observed, which is common with fixed-point iterative solvers. For instance, the difference of Force between two successive iterations is $\pm 0.12\%$, which is much less than the convergence criteria of 1%. Oscillations of such magnitude were observed as well with the *LS-SUPG* formulation in the *VP-Test-Case-01* test-case.

Shape Correction method	Conv. inc/iters	Comp. Time (min)	Tool Force (T)	Tool Torque (kN-m)
<i>ForgeNxt[®]2016 (LS-SUPG)</i>	40	not converged	15.81 ± 0.02	9.36 ± 0.03
<i>ForgeNxt[®]2016 (SUPG-dif)</i>	40	not converged	15.94 ± 0.02	9.42 ± 0.04
<i>ForgeNxt[®]inc</i>	512	132	15.49	9.20

Table 3.5.3: Global performance of the *VP-Test-Case-02* with the *ForgeNxt[®]2016* algorithm in comparison to the *ForgeNxt[®]inc* solution (on 12 cores)

Like the previous test-case, the convergence parameters for *VP-Test-Case-02* are analyzed for understanding the reason behind the non-convergence of the *ForgeNxt[®]2016* algorithm. The Figure 3.5.8a, Figure 3.5.8b, Figure 3.5.9a and Figure 3.5.9b present the respective convergence of force, equivalent strain, correction and flux loss. We can observe that the force and geometric convergence criteria are met within 7 iterations and remain converged until the maximum number of iterations are reached. The stability of the *SUPG* formulation is again seen to be better in comparison to the *LS-SUPG* formulation due to over all lower magnitude of change in these parameters with consecutive iterations. Minor flux loss $< 0.5\%$ is also observed with both the free surface formulations, even though this is not problematic from the convergence point of view. The source of the non-convergence is however the equivalent strain which fails to converge and observes large fluctuations throughout the resolution. The reason behind this fluctuation is not obvious from the presented results and is attempted to be understood in the following section. However, its convergence is critical as the equivalent strain is a history variable in the viscoplastic materials and used

for computing the local material consistency K . This consistency effects the velocity which is used for the computation of free surface. Hence large changes in equivalent strain correspond to large fluctuations in velocity and the free surface shape, which is also seen as an large fluctuations in force and correction after 20 iterations.

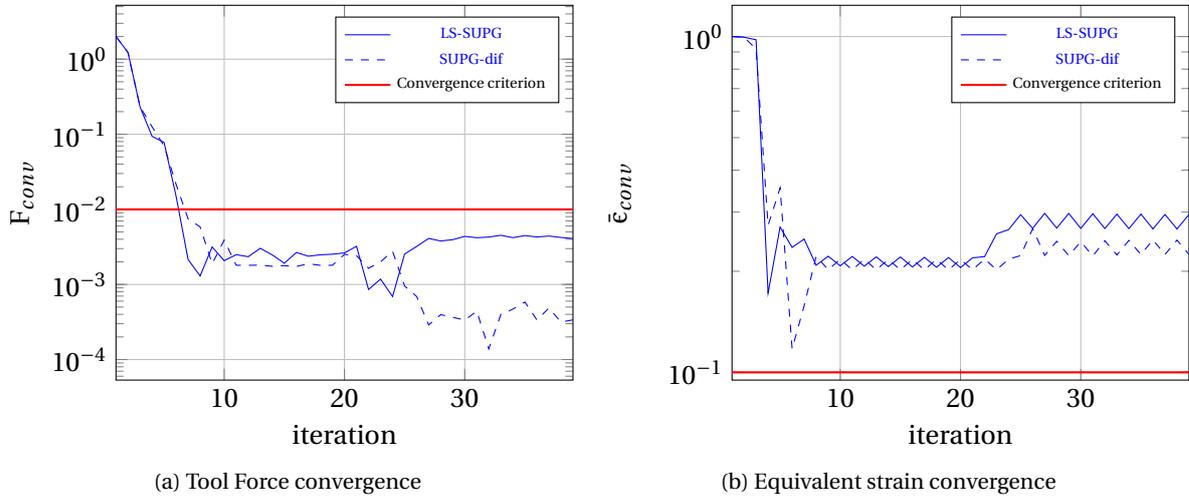


Figure 3.5.8: Comparison of tool force and equivalent strain convergence and flux loss for **VP-Test-Case-01** with the **ForgeNxt®2016** formulation

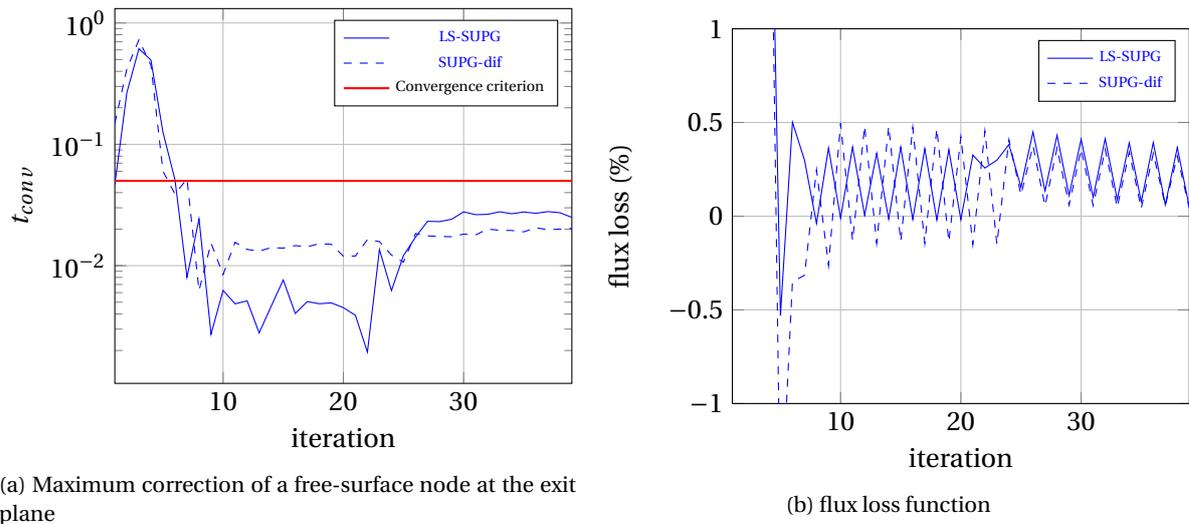


Figure 3.5.9: Comparison of free-surface correction convergence and flux loss for **VP-Test-Case-02** with the **ForgeNxt®2016** formulation

3.6 Reasoning for instabilities observed with **VP-Test-Case-02**

The **VP-Test-Case-01** is a simple rolling test case with a uniform contact surface with flat dies, which results in a contact surface with a nearly straight first contact line. However, the unstable contours under the contact surface signaled an unstable contact. However, the simplicity of the geometry and the low magnitude of oscillations ensured that the steady-state algorithm converged to a solution which didn't vary more than the used convergence criteria. In simple words, the algorithm presented in Section 3.4.2 is robust enough to handle such a problem. However, as soon as we are in the domain of complex geometries, the algorithm is unable to converge to a unique solution. We observed that this instability in the solution is due to the non-convergence of the equivalent strain. If we compare the equivalent strain field at the penultimate and last iteration

(see Figure 3.6.1), we can see that there is a change of the field under the tool contact, which was as well indicated by *VP-Test-Case-01* solution. The equivalent strain is directly linked to the equivalent rate of deformation, which observed a similar change, it does not take a lot of investigation to observe this change phenomenon in the contact status (a contact binary test with 0: nodes not in contact, 1: nodes in contact) for certain nodes under the tool. This highlights that the observed problem is linked to a non-converged contact surface Γ_c , which is known to be the most challenging problem with the steady-state resolution. Thus, the first objective of this work is targeted to improve the robustness of the steady-state algorithm in order to provide a stable steady-state solution.

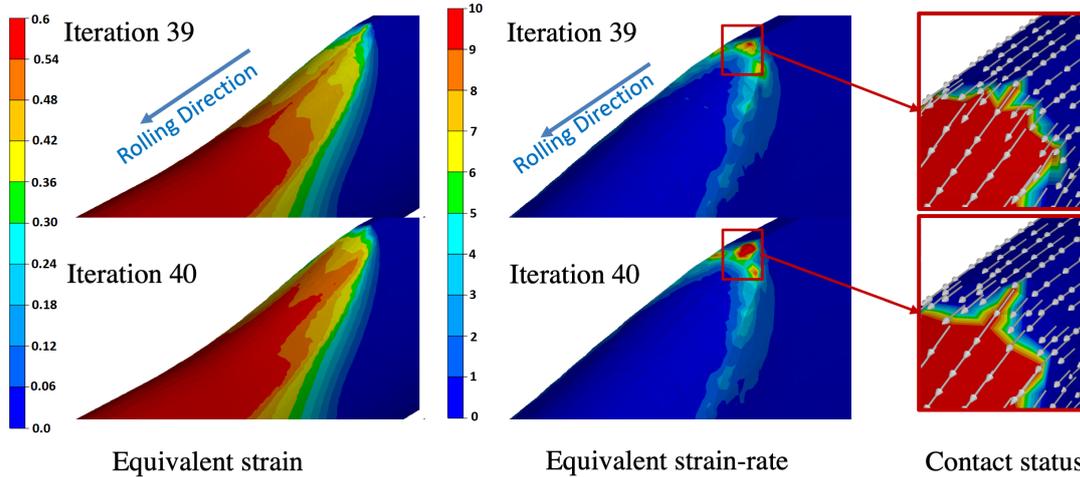


Figure 3.6.1: Comparison of equivalent strain, equivalent rate of deformation and contact-status near the tool contact area for *VP-Test-Case-02* at two consecutive iterations

3.7 Conclusions from the chapter

The present chapter focuses on the discussion of the previously developed *ForgeNxt*[®]2016 algorithm [7] and testing it for hot rolling problems. A fixed-point iterative algorithm is chosen for resolution since the problem of velocity-shape is strongly coupled. In the *Step-1*, the thermo-mechanical problem of the metal forming process is solved. It entails the computation of velocity field \vec{v} on an approximated steady-state domain Ω_x using the mixed (\vec{v}, p) formulation. From the computed velocities, we solve the steady-state energy equilibrium and equivalent strain equations to compute the respective unknown Temperature T and equivalent strain $\bar{\epsilon}$ fields. Given the convection dominance of these problems, SUPG stabilized finite elements are used. In the *Step-2*, the steady-state geometry $\vec{x}^{(i+1)}$ is updated from a free-surface scalar equation. The shape computation problem is also convection dominant, hence, two PG functions are proposed LS-SUPG and SUPG. For problems with geometric singularities, two formulations are proposed: (i) SUPG-reg (ii) SUPG-dif. The former does not change the formulation but requires regularization of the surface mesh after each Newton-Raphson iteration. The latter proposes a vector form of the free-surface equation for free-surfaces with edges and aims to compute the correction field in two linearly independent directions (for the edge). The vector form of free-surface equation is straightforward to develop with LS-SUPG test function in comparison to the SUPG-dif test function. The latter is however much better in terms of robustness as a solution is ensured within a few iterations for most of the tested rolling cases. For the SUPG-dif test function, the free-surface equation is independently resolved in the linearly independent directions and thus makes it possible to find a robust and accurate formulation.

An important aspect of the steady-state formulation with regard to the robustness is the contact formulation and its coupling in the iterative algorithm. Both the velocity field as well as the free-surface shape correction invoke constraints at the contact with the tool surface. The two con-

ditions defining the contact constraints are applied with the penalty form in the weak formulation of mechanical and free-surface problems. Consequently, the contact is updated at the end of each step. The convergence of the fixed-point iterative algorithm is tested with four convergence parameters for force, equivalent strain, temperature, and shape correction. In doing so the algorithm also aims to ensure the convergence of the contact surface under the aegis of the coupled equations.

Two hot rolling problems are tested with the [ForgeNxt[®]2016](#) algorithm. The [VP-Test-Case-01](#) is a simple hot rolling problem without large shape changes and has a uniform contact surface with the tool. The free-surface is computed with the [LS-SUPG](#) and [SUPG-dif](#) formulations. The [ForgeNxt[®]2016](#) is found to be 60-75 times faster in comparison to the incremental resolution of the same problem. Even though the solution converges with both methods, the solution contours hint oscillations under the tool contact. The [VP-Test-Case-02](#) is a shape rolling problem with a product that observes a shape change from an oval section to a square section. This steady-state algorithm fails to converge for this test-case and the solution is found to be unstable with both the test functions. It is found that the equivalent strain fails to converge for the problem, and careful analysis of the results hints a link with the contact surface (with the tool). This observation is the primary motivation to look into detail the contact formulation and provide other possibilities to improve the robustness of the algorithm.

Résumé

Ce chapitre se concentre sur la présentation de l'algorithme *ForgeNxt*[®] 2016 et son évaluation pour des problèmes de laminage à chaud. Un algorithme itératif de point-fixe est choisi pour la résolution car le problème vitesse/forme est fortement couplé. Dans une première étape, le problème thermo-mécanique de mise en forme des métaux est résolu. Il implique le calcul du champ de vitesse \vec{v} sur un domaine approximatif à l'état stationnaire. Ensuite, on calcule la température et la déformation équivalente avec la vitesse calculée à partir de l'équilibre énergétique et les équations de déformation équivalentes. Étant donné la dominance de convection de ces problèmes, des éléments finis stabilisés *SUPG* sont utilisés. Dans la deuxième étape, la géométrie de l'état stationnaire est calculée à partir d'une équation scalaire de surface libre. Le problème de calcul de surface libre est également dominante en convection, et donc deux fonctions de Petrov-Galerkin sont proposées *LS-SUPG* et *SUPG*. Pour traiter les géométries avec singularités comme les arêtes, deux formulations sont proposées: (i) *SUPG-reg* (ii) *SUPG-dif*. La première ne change pas la formulation, mais nécessite une régularisation du maillage de surface après chaque itération de Newton-Raphson. La seconde propose une forme vectorielle pour les surfaces libres avec arêtes, et vise à calculer le champ de correction dans deux directions linéairement indépendantes (pour l'arête). La forme vectorielle de l'équation de surface libre est simple à développer avec la fonction de test *LS-SUPG* en comparaison avec la fonction de test *SUPG-dif*. La seconde est cependant meilleure en terme de précision et robustesse car une solution est assurée en quelques itérations pour la plupart des cas de laminage testés. Pour la méthode *SUPG-dif*, l'équation de surface libre est résolue indépendamment dans les directions linéairement indépendantes. Un aspect important de la formulation en état stationnaire, ce qui concerne la robustesse de l'algorithme itératif, est la formulation de contact et son couplage. Le champ de vitesse ainsi que la correction de la forme de la surface libre doivent invoquer des conditions de contact à la surface de l'outil. Les deux conditions définissant les contraintes de contact sont appliquées avec la forme de pénalisation dans la formulation faible des problèmes mécaniques et de surface libre. Par conséquent, le contact est actualisé à la fin de chaque étape à partir des multiplicateurs de Lagrange respectifs de chaque problème. La convergence de l'algorithme itératif de point-fixe est testée avec quatre paramètres de convergence pour l'effort d'outil, la déformation équivalente, la température et la forme de la pièce à la sortie. Ce faisant, l'algorithme vise également à assurer la convergence de la surface de contact avec des équations couplées. Deux problèmes de laminage à chaud sont présentés avec l'algorithme *ForgeNxt*[®] 2016. Le *VP-Test-Case-01* est un problème de laminage à chaud d'une tôle épaisse avec une surface de contact uniforme avec l'outil. L'algorithme *ForgeNxt*[®] 2016 s'avère une solution convergente et 60 à 75 fois plus rapide par rapport à la résolution incrémentale du même problème. Le *VP-Test-Case-02* est un problème de laminage à chaud de forme et présente un grand changement de forme: section d'entrée ovale transformée en une section carrée. Cet algorithme stationnaire trouve une solution non-unique et on observe de légères oscillations dans l'historique d'effort d'outil. On constate que la déformation équivalente ne parvient pas à converger pour le problème, et des résultats suggèrent un lien avec la surface de contact avec l'outil. Cette observation est la principale motivation pour la proposition d'amélioration de la formulation du contact et de la robustesse de l'algorithme.

Bibliography

- [1] A.N. Brooks and T.J.R. Hughes. Streamline Upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer methods in applied mechanics and engineering*, 32(1):199–259, 1982.
- [2] T. Coupez. *Grandes transformations et remaillage automatique*. PhD thesis, Paris, ENSMP, 1991.
- [3] M. Fortin and F. Brezzi. *Mixed and hybrid finite element methods*, volume 734. New York: Springer-Verlag, 1991.
- [4] M. Hachani. *Amélioration de la modélisation de contact pour les procédés à faible zone de contact*. PhD thesis, ENSMP, 2011. URL <http://www.theses.fr/2011ENMP0078>.
- [5] P. Knobloch. On the choice of the SUPG parameter at outflow boundary layers. *Adv. Comput. Math.*, 31:369–389, 10 2009. doi: 10.1007/s10444-008-9075-6.
- [6] D.L. Page, Y. Sun, A.F. Koschan, J. Paik, and M.A. Abidi. Normal vector voting: Crease detection and curvature estimation on large, noisy meshes. *Graphical Models*, 64(3):199 – 229, 2002. ISSN 1524-0703. doi: <https://doi.org/10.1006/gmod.2002.0574>. URL <http://www.sciencedirect.com/science/article/pii/S1524070302905746>.
- [7] U. Ripert. *Méthode itérative de recherche de l'état stationnaire des procédés de mise en forme : application au laminage*. PhD thesis, ENSMP, 2014. URL <http://www.theses.fr/2014ENMP0006>.
- [8] U. Ripert, L. Fourment, and J.L. Chenot. An upwind least square formulation for free surfaces calculation of viscoplastic steady-state metal forming problems. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1):1, 2015.
- [9] P. Wriggers and G. Zavarise. Computational contact mechanics. *Encyclopedia of computational mechanics*, 2004.

Chapter 4

A stabilized contact constraint for fixed-point, multi-field, steady-state formulations

Contents

4.1	Problem with contact coupling	76
4.2	Consistent contact coupling	77
4.3	Description of the nodal form of free-surface equation	78
4.3.1	Nodal condensation SUPG-NC	78
4.3.2	2-DoF form of nodal free-surface equation (SUPG-NC-dif)	78
4.3.3	Optimization of SUPG stabilization term α	78
4.3.4	Free-surface boundary condition for the symmetry plane nodes	79
4.4	Test of new free-surface formulation with analytical test-cases	80
4.4.1	Principle of the analytical test-case	80
4.4.2	Description of the analytical tests	81
4.4.3	Test problems with flat sheet	81
4.4.4	Test problems with 3D tubes	84
4.4.5	Conclusions from analytical test-cases	88
4.5	Updated contact coupling	89
4.5.1	Explicit form of contact constraint	89
4.5.2	Implicit form of contact constraint	91
4.5.3	Updated steady-state algorithm	96
4.6	Performance with hot-rolling test-cases	96
4.6.1	Flat rolling problem: VP-Test-Case-01	96
4.6.2	Shape rolling problem: VP-Test-Case-02	101
4.6.3	Shape rolling problem: VP-Test-Case-03	105
4.7	Conclusions from the chapter	116

The existing steady-state algorithm is proven to be fast for a simple hot rolling test-case. However, the formulation lacks robustness as instabilities/oscillations are observed in the fields (Force, Torque, strain, etc.). A careful analysis of the solution demonstrates a link between observed oscillations and the contact surface with the tool (see Section 3.6). It is thus hypothesized that the reason behind the unstable solution is the contact coupling explained in Section 3.4.1. The present chapter is mainly focused on the investigation and development of contact formulation in a velocity-based steady-state algorithm, which is not well studied in the limited literature with simple configurations [3]. Even though the contact is not the primary convergence variable of the algorithm, a non-stable contact is seen to have a large impact on the stability of other variables that are critical to convergence. This motivates the present investigation for finding a better contact formulation for the steady-state algorithm and hence improve its robustness.

4.1 Problem with contact coupling

The first step is to identify the problem with the existing formulation: we learned that the impenetrability conditions are applied during the computation of the velocity field with the equation (3.2.14) and the computation of the shape correction with the equation (3.3.2b). The contact is updated at the end of each step and it is hoped that the fixed-point iterative algorithm converges to a unique contact surface at the convergence of the algorithm. It means that in the hindsight, the contact conditions applied in the two steps complement and are consistent with each other and result in a unique contact surface. This is however not truly seen in the test-cases studied in the Section 3.5. From the general overview of the conditions applicable on contact, it is understood that the material impenetrability condition applied on the coordinate correction during the free-surface correction (3.3.2b) is a more comprehensible contact condition. The contact condition on the velocity field is necessary but needs to be investigated. The contact impenetrability condition, applied in the penalty form, along with the free-surface equation in the weak form (3.3.3) are invoked again:

$$\forall k \in \Gamma_h, \quad S_k(\vec{v}^{(i+1)}, t^{(i+1)}) = \int_{\Gamma_h} N_k^{\text{SUPG}}(\vec{v} \cdot \vec{n}(\mathbf{x})) ds_h + \rho_c^{(2)} s_k [\bar{h}(t_k)]^+ \vec{d}_k \cdot \vec{n}_k^{\text{tool}} = 0 \quad (4.1.1)$$

In the free surface residual, the term in $[\]^+ \geq 0$ defines the unilateral contact condition on a node k . This implies that there are following possibilities for the free-surface residual:

$$\text{if } \bar{h}(t_k) < 0 \quad \int_{\Gamma_h} N_k^{\text{SUPG}}(\vec{v} \cdot \vec{n}(\mathbf{x})) ds_h = 0 \quad (4.1.2a)$$

$$\text{otherwise } \bar{h}(t_k) \geq 0 \quad \int_{\Gamma_{c,h}} N_k^{\text{SUPG}}(\vec{v} \cdot \vec{n}(\mathbf{x})) ds_h \leq 0 \quad (4.1.2b)$$

The (4.1.2a) represents that for a free surface node k , far away from the contact, only the free-surface condition is applicable and the penalty part is nullified as the gap function $\delta_k > 0$. On the other hand, the condition (4.1.2b) results when the node k is in contact. This means that the gap function $\delta_k \leq 0$ or the constraint is active $\bar{h}(t_k) \geq 0$. In such scenarios, the large value of the penalty coefficient $\rho_c^{(2)}$ dominates the free surface residual, and forces the projection of the node k onto the contact surface. This means that the free-surface condition may not be verified. As a consequence, the node $k \in \Gamma_c$ in contact must respect the following condition:

$$\forall k \in \Gamma_{c,h}, \quad \int_{\Gamma_{c,h}} N_k^{\text{SUPG}}(\vec{v} \cdot \vec{n}(\mathbf{x})) ds_h \leq 0 \quad (4.1.3)$$

This condition may be understood such that the node k may have a zero normal velocity, which lets the node slide on the contact surface, or negative normal velocity that pushes the node away from the contact surface in the next iteration of the free surface computation. In other words, the said condition is an impenetrability condition. A positive normal velocity would mean that the

velocity vector at the said contact node must be reoriented such that it conforms to the impenetrability condition. In short, we see that the constraint (4.1.3) has the same effect as the Signorini conditions (3.2.14) applicable at the contact surface during the mechanical resolution. For the convenience of following-up, we rewrite the existing velocity contact constraint below:

$$\forall k \in \Gamma_{c,h}, \quad \vec{v}_k \cdot \vec{n}_k^{tool} \leq 0 \quad (4.1.4)$$

It is seen that the two constraints are different in the following manner:

1. The Signorini condition is described with the tool normal \vec{n}^{tool} , whereas the free-surface contact condition is described with the mesh normal \vec{n}
2. The second difference between the two is the type of weak form used. In the former, the constraint is applied in a nodal form whereas in the latter a weighted residual form or the integral form is applied.

The contact conditions for both the mechanical resolution and free-surface computation must complement each other in order to obtain a robust fixed-point algorithm. However, this is not the case with the existing formulation as it is observed that the intuitively found Signorini constraint is not consistent with the contact condition from the free-surface correction.

4.2 Consistent contact coupling

It is hypothesized that the problem observed with instabilities in the solution with the existing steady-state algorithm is because of the inconsistent description of the constraints on the contact surface in the two steps of the algorithm. Thus, the objective of this section is to find a way to improve the consistency of the contact formulation for the steady-state algorithm. Following options are envisaged for improving the formulation:

Option A

The easiest choice to improve the contact consistency is to use equation (4.1.3) as the contact constraint for mechanical resolution. Hence, the corresponding penalty functional in equation (3.2.25a) is updated with the new constraint, and we compute the unknown velocity field from the corresponding residual.

$$\min_{\vec{v}} \left(\Phi_p'(\vec{v}) \right), \quad \Phi_p(\vec{v}) = \Phi(\vec{v}) + \frac{1}{2} \rho_c^{(1)} \sum_{k \in \Gamma_{c,h}} \left[\int_{\Gamma_{c,h}} N_k^{SUPG}(\vec{v} \cdot \vec{n}(\mathbf{x})) ds_h \right]^2 \quad (4.2.1)$$

The residual is found with the minimization of the functional Φ_p and discretized to resolve the mechanical problem with the FE method to find the unknown velocity field. Here we assume that the derivative of the normal with the velocity $d\vec{n}/d\vec{v} = \mathbf{0}$ due to the fixed-point assumption.

$$\Phi_p'(\vec{v}) = \Phi'(\vec{v}) + \rho_c^{(1)} \sum_{k \in \Gamma_{c,h}} \left[\int_{\Gamma_{c,h}} N_k^{SUPG}(\vec{v} \cdot \vec{n}(\mathbf{x})) ds_h \right]^+ \left(\int_{\Gamma_{c,h}} N_k^{SUPG} \left(\frac{d\vec{v}}{d\vec{v}_l} \cdot \vec{n}(\mathbf{x}) \right) ds_h \right) \quad (4.2.2)$$

The integral term is written with Finite element discretization as equation (4.2.3a). However, this residual consists of the product of two mass matrices $\vec{U}_{km} \times \vec{U}_{km}$ each containing the term \vec{U}_{km} , each term defined with equation (4.2.3b). This product results in doubling of the mass matrix bandwidth, which poses a problem with its resolution in ForgeNext[®] solver as such a term cannot be accepted in the discretized system. Thus, this option remains impracticable from the implementation point of view.

$$\forall k \in \Gamma_h, \quad \int_{\Gamma_h} N_k^{SUPG}(\vec{v} \cdot \vec{n}(\mathbf{x})) ds = \sum_{m \in \mathcal{I}_k} \vec{v}_m \cdot \vec{U}_{km} \quad (4.2.3a)$$

$$\vec{U}_{km}(\vec{x}) = \int_{\Gamma_h} N_k^{SUPG} N_m \vec{n}(\mathbf{x}) ds_h \quad (4.2.3b)$$

Option B

In another possibility, we derive a nodal form of the contact constraint from the free-surface equation by invoking an operation called the *nodal condensation*. This option is of primary interest for improving the contact coupling while keeping its implementation practicable. The derivation of the constraint is discussed in the following section 4.3.

4.3 Description of the nodal form of free-surface equation

The first step for finding a consistent contact formulation is to derive the nodal form of the free-surface equation.

4.3.1 Nodal condensation SUPG-NC

The Finite Element discretization of the integral form of the free-surface equation (4.2.3a) is invoked. We apply the *nodal condensation* to the matrix $\tilde{\mathbf{U}}_{km}$ that entails the summation of the extra-diagonal terms to its diagonal as shown in equation (4.3.1a). The resultant vector $\tilde{\mathbf{U}}_{kk}(\vec{x})$ is described with equation (4.3.1b). Here, $\vec{\mathbf{u}}_k$ is a surface normal average computed at the node k shared by a patch of elements \mathfrak{N}_k . However, this average is done using an upwind-biased SUPG function, so we term this normal as the *upwind normal*.

$$\tilde{\mathbf{U}}_{km}(\vec{x}) = \begin{cases} \tilde{\mathbf{U}}_{kk}(\vec{x}) = \sum_m \tilde{\mathbf{U}}_{km} & \text{if } k = m \\ 0 & \text{if } k \neq m \end{cases} \quad (4.3.1a)$$

$$\tilde{\mathbf{U}}_{kk}(\vec{x}) = \vec{\mathbf{u}}_k = \int_{\Gamma_h} \mathbf{N}_k^{\text{SUPG}} \underbrace{\sum_m \mathbf{N}_m \vec{\mathbf{n}}(\mathbf{x})}_{=1} ds_h = \int_{\Gamma_h} \mathbf{N}_k^{\text{SUPG}} \vec{\mathbf{n}}(\mathbf{x}) ds_h \quad (4.3.1b)$$

$$\forall k \in \Gamma_h, \int_{\Gamma_h} \mathbf{N}_k^{\text{SUPG}} (\vec{\mathbf{v}} \cdot \vec{\mathbf{n}}(\mathbf{x})) ds \approx \vec{\mathbf{v}}_k \cdot \int_{\Gamma_h} \mathbf{N}_k^{\text{SUPG}} \vec{\mathbf{n}}(\mathbf{x}) ds_h \quad (4.3.1c)$$

Hence, after the nodal condensation, the integral form of the free surface residual can be replaced with the nodal form represented in equation (4.3.1c). The objective of this approximation is to replace the integral form in equation (3.3.3a) with the nodal form. This leads to a free surface residual in nodal form which becomes consistent with the nodal form used as the velocity contact constraint (4.1.4). Hence, the updated nodal free-surface residual $\tilde{\mathbf{S}}_k$ becomes:

$$\forall k \in \Gamma_h, \quad \tilde{\mathbf{S}}_k(\vec{\mathbf{v}}, t) = \vec{\mathbf{v}}_k \cdot \vec{\mathbf{u}}_k(\vec{x}) + \rho_c^{(2)} s_k [\bar{h}(t_k)]^+ \vec{\mathbf{d}}_k \cdot \vec{\mathbf{n}}_k^{\text{tool}} = 0 \quad (4.3.2)$$

4.3.2 2-DoF form of nodal free-surface equation (SUPG-NC-dif)

For the geometries with singularities like edges, we describe the 2-dof form of the nodal free-surface equation as following:

$$\forall k \in \Gamma_{\text{edge}}, \quad j = 1, 2 \quad \tilde{\mathbf{S}}_k^{[j]}(\vec{\mathbf{v}}, t^{[j]}) = \vec{\mathbf{v}}_k \cdot \vec{\mathbf{u}}_k^{[j]}(\vec{x}) + \rho_c^{(2)} s_k [\bar{h}(t_k)]^+ \vec{\mathbf{d}}^{[j]} \cdot \vec{\mathbf{n}}_k^{\text{tool}} = 0 \quad (4.3.3)$$

Here $\vec{\mathbf{u}}_k^{[j]}$ is the upwind normal defined at node k on the edge in the direction $\vec{\mathbf{d}}^{[j]}$. For an edge, only two directions $j = 1, 2$ are envisaged. This formulation is referred to as SUPG-NC-dif.

4.3.3 Optimization of SUPG stabilization term α

As described previously in the Section 2.3.5, the SUPG stabilization coefficient α must be chosen between $[0, 1]$. For 1D SUPG elements, the coefficient is generally chosen as $\alpha = 1/2$. However, in general practice, even with higher dimension elements, the coefficient α is chosen to be $1/2$. This choice is however arbitrary and can prove to be nonsensical while used with 2D elements. In this

section, we highlight the problems related to the choice of $\alpha = 1/2$ for 2D elements, which become significant with the nodal condensation. A technique to find the optimum value for α for a given dimension is hence presented. [1] remarks that the SUPG elements must conform to two main conditions:

$$\begin{cases} \sum_k N_k^{\text{SUPG}} = 1 \\ w_k^f = \int_f N_k^{\text{SUPG}} d\xi \geq 0 \end{cases} \quad (4.3.4)$$

The first condition represents that the total sum of shape functions in an element must be 1 and the second condition states that the weights computed for interpolation must always be non-negative. It can be easily verified that the 1D-SUPG elements with $\alpha = 1/2$ comply with the above-mentioned conditions in equation (4.3.4), as the SUPG stabilization coefficient must be $0 < \alpha \leq 1/2$ as found in equation (4.3.5).

$$\text{in 1D: } w_k^f = \underbrace{\int_f N_k d\xi}_{1/2} + \alpha \underbrace{\int_f C_k d\xi}_{[-1,1]} \geq 0; \quad \Rightarrow \alpha \leq \frac{1}{2} \quad (4.3.5)$$

Thus, for 1D elements (see Figure 4.3.1), $\alpha = 1/2$ results in weight $w_k^f = 0$ for the downwind elements, and $w_k^f = 1$ for the upwind elements. A value $\alpha > 1/2$ would result in negative weightage to the downwind elements, which would contradict the second condition. With $\alpha = 0$, we retrieve only the Galerkin part in the N_k^{SUPG} function (see equation (3.3.3b)), hence the numerical stabilization is lost.

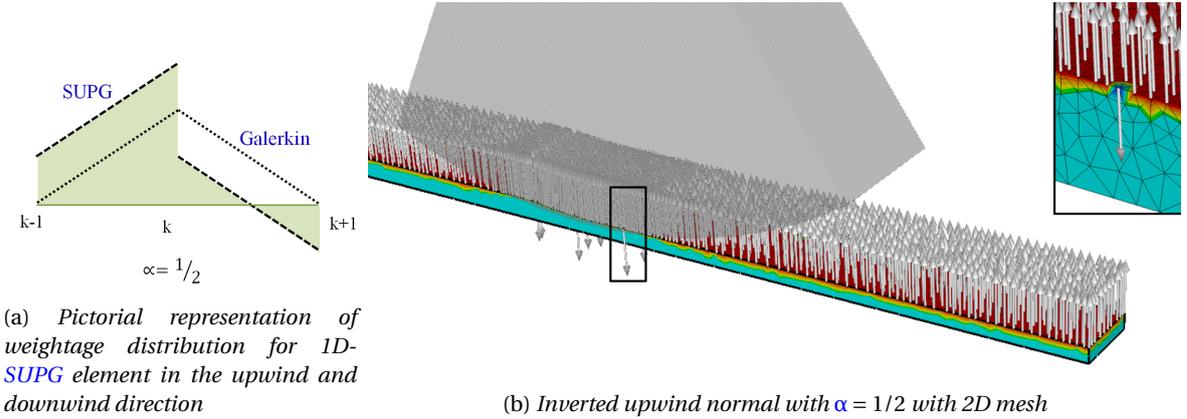


Figure 4.3.1: Finding an optimized value for the SUPG stabilization coefficient

On the application of the same conditions to the 2D-SUPG surface elements, as used with the free surface computation, the conditions in equation (4.3.5) suggest that the SUPG stabilization coefficient must be $0 < \alpha \leq \frac{1}{3}$.

$$\text{in 2D: } w_k^f = \underbrace{\int_f N_k d\xi d\eta}_{1/6} + \alpha \underbrace{\int_f C_k d\xi d\eta}_{[-1,1]} \geq 0; \quad \Rightarrow \alpha \leq \frac{1}{3} \quad (4.3.6)$$

With the value $\alpha = \frac{1}{2}$ used in ForgeNxt[®]2016, the second condition is violated and if the same value is used for computing the *upwind normal*, may result in the inversion of the \vec{u}_k as shown in Figure 4.3.1b. It is thus recommended to use $\alpha = \frac{1}{3}$ for 2D-SUPG elements, especially with *nodal condensation*.

4.3.4 Free-surface boundary condition for the symmetry plane nodes

The correction t must always be computed in the direction normal to the surface at the symmetry plane, and hence the component in the direction normal to the symmetry plane \vec{n}^{sym} is nullified.

Thus the upwind normal at the symmetry plane is found as following:

$$\forall k \in \Gamma_{sym}, \quad \vec{u}_k' = \vec{u}_k - \left(\vec{u}_k \cdot \vec{n}_k^{sym} \right) \vec{n}^{sym} \quad (4.3.7)$$

With the projection, the nodal form of the free-surface correction residual (see equation (4.3.8)) remains the same due to the condition (3.2.11).

$$\forall k \in \Gamma_{sym}, \quad \vec{v}_k \cdot \vec{u}_k' = \vec{v}_k \cdot \vec{u}_k - \underbrace{\left(\vec{u}_k \cdot \vec{n}_k^{sym} \right) \vec{v}_k \cdot \vec{n}_k^{sym}}_{=0} = \vec{v}_k \cdot \vec{u}_k \quad (4.3.8)$$

4.4 Test of new free-surface formulation with analytical test-cases

In this section, the proposed nodal form of the free surface equation (4.3.3) is compared with the integral form of the free-surface equation in (3.3.3a). For doing so, various analytical functions are imposed on a sheet, a cylinder and a square tube to deform their shape and predicted using the free surface condition imposed by the weak form. These cases are adopted from the work of [2]. The shape prediction from the nodal form of the free-surface equation is compared using both scalar (for sheet and cylinder) and vector forms (for square tube). It is to be noted that with the analytical tests, only free surface conditions are applied and there is no contact, hence, the penalty term is zero. The impact of SUPG stabilization coefficient α on the solution accuracy is also tested.

4.4.1 Principle of the analytical test-case

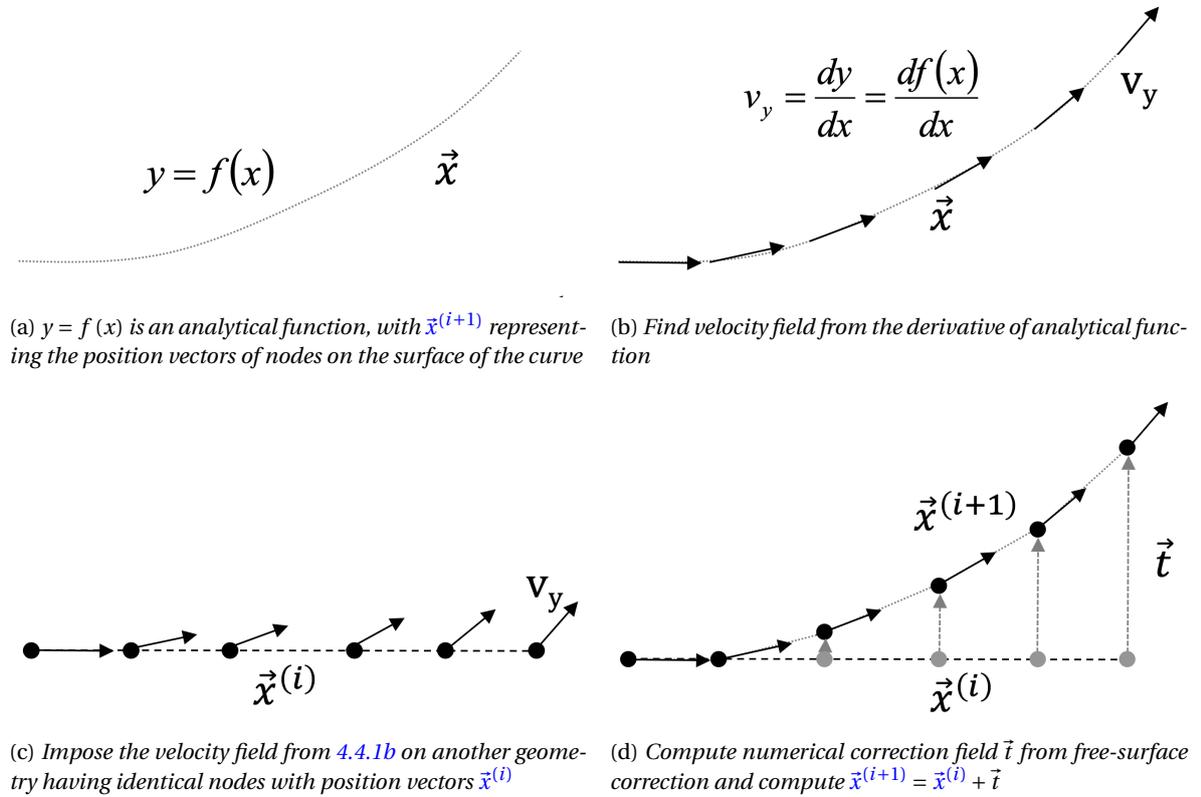


Figure 4.4.1: Principle of analytical test for free-surface correction

In order to compare the proposed formulations, we test them to predict analytical surface flows. The principle of the analytical test is to deform a 2D sheet or a 3D tube by a predefined analytical function $y = f(x, z)$ as shown in Figure 4.4.1a. The tangent at each point on the deformed sheet defines the velocity field (see Figure 4.4.1b) as the surface verifies the streamline conditions. If

this velocity field is imposed on an undeformed geometry (see Figure 4.4.1c) and then resolved using the weak form of the free-surface equation, the undeformed shape must comply to the free-surface conditions which are equivalent to the streamline conditions. Hence it finds the original shape from the velocities from the numerical resolution. The numerical solution of the function y_k is computed from the correction field \vec{d} as shown in Figure 4.4.1d.

4.4.2 Description of the analytical tests

Three different geometries have been tested, as shown in the Table 4.4.1. On these geometries, analytical Gaussian, Sinusoidal and Quadratic functions are applied as deformations.

S.No.	Geometry	Analytical function
1.	2D Flat sheet	Gauss
		Sinusoidal
2.	3D Cylinder tube	Quadratic
		Sinusoidal
3.	3D Square tube	Quadratic
		Sinusoidal

Table 4.4.1: Different analytical test functions tested with free-surface formulation

The nodal form of the free-surface equation uses SUPG function as developed in (4.3.3) and its solution is represented with the acronym SUPG-NC. The integral form uses LS-SUPG and SUPG test functions. For the SUPG shape functions, the coefficient $\alpha = 1/2, 1/3$ are tested.

The numerical solution computed with each of these methods are compared with the analytical solution for (i) the relative L_∞ and L_2 error norms from equation (4.4.1) (ii) the relative L_2 error convergence with mesh refinement. In the said equation, t_k^h and t_k^a represent the respective numerical and analytical correction computed with a chosen mesh size.

$$L_\infty = \frac{\max_k |t_k^h - t_k^a|}{\max_k |t_k^a|} \quad L_2 = \frac{\sqrt{\sum_k (t_k^h - t_k^a)^2}}{\sqrt{\sum_k (t_k^a)^2}} \quad (4.4.1)$$

4.4.3 Test problems with flat sheet

The first test is done with two analytical functions (i) Gauss (ii) Sinusoidal. Both these functions are applied as a deformation on a 2D rectangular sheet of dimension 100mm x 10mm. For this test, only a scalar correction is required in the direction normal to the surface.

4.4.3.1 Gauss deformation

A Gaussian deformation (equation (4.4.2)) is applied on the rectangular sheet as shown in Figure 4.4.2. The numerical solution to the corrections are computed with four different mesh sizes for mesh convergence study. The L_∞ and L_2 relative errors with each of the mesh sizes are tabulated in Table 4.4.2. The error convergences are also plotted in Figure 4.4.3.

$$z = 5 \exp\left(-\left(\frac{x-40}{10}\right)^2\right) \quad (4.4.2)$$

From the convergence plots in the Figure 4.4.3, we observe that all the tested formulations, including the new nodal formulation (SUPG-NC), are convergent. In general the L_2 error is lower for SUPG and LS-SUPG formulations than the L_∞ error except for the SUPG-NC formulation. This is because, with the integral form, the error is only concentrated in few elements, but with the nodal

form, the error is large at large slopes, which is a sizable part of the sheet. The solution accuracy is the best with the SUPG function with the integral formulation and the worst with SUPG-NC. The SUPG stabilization coefficient $\alpha = 1/3$ results in better accuracy in comparison to $\alpha = 1/2$, and this difference is much more significant for SUPG-NC method. It is also noticed that error convergence rate is lower with the SUPG-NC formulation in comparison to the SUPG integral formulation.

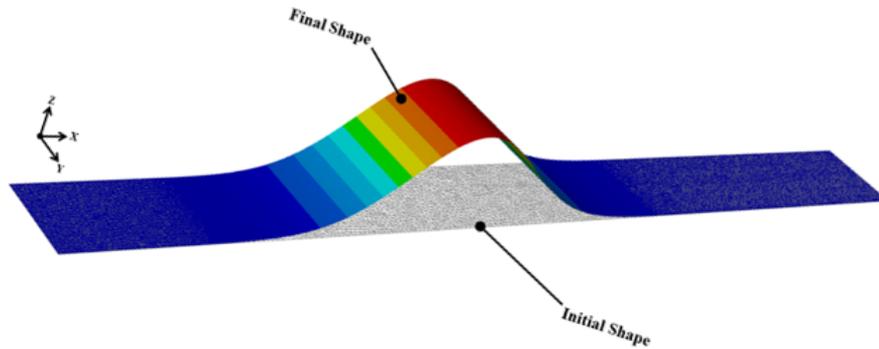


Figure 4.4.2: Gauss analytical functions tested on 2D sheet

Nodes →		1900		6000		16000		26000	
Method	α	L_∞	L_2	L_∞	L_2	L_∞	L_2	L_∞	L_2
SUPG	1/2	0.28	0.12	0.08	0.03	0.05	0.01	0.02	0.01
	1/3	0.25	0.11	0.07	0.03	0.04	0.01	0.02	0.01
SUPG-NC	1/2	3.34	3.46	1.80	1.96	1.08	1.18	0.70	0.77
	1/3	2.21	2.34	1.29	1.32	0.73	0.79	0.47	0.52
LS-SUPG	1.0	0.36	0.23	0.25	0.09	0.1	0.03	0.04	0.01

Table 4.4.2: Comparison of L_2 and L_∞ % errors for the Gauss function deformation on a 2D flat sheet shown in Figure 4.4.2

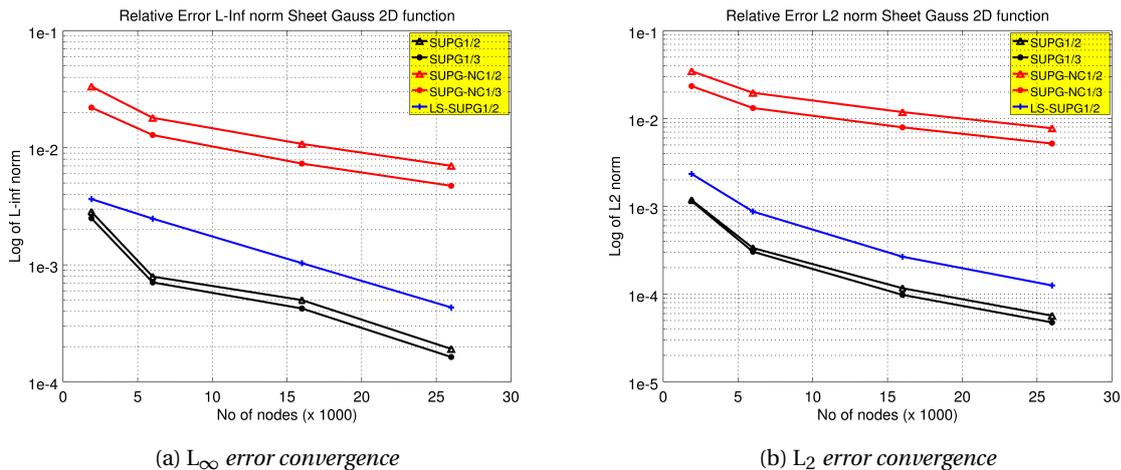


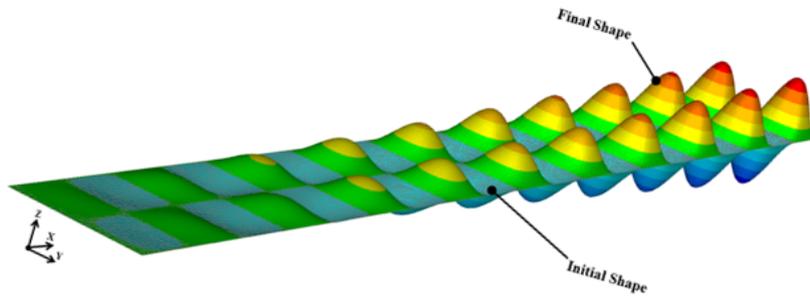
Figure 4.4.3: h convergence for 2D Gauss function with different formulations for free-surface resolution

4.4.3.2 Sinusoidal deformation

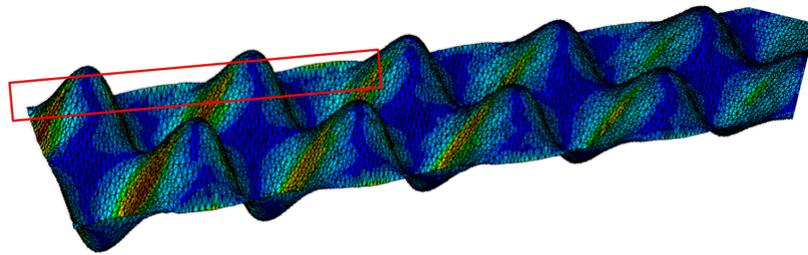
Next, we apply a sinusoidal function (equation (4.4.3)) of varying amplitude in the flow x and transverse z directions on the same 2D rectangular sheet. The analytical function is represented in Figure 4.4.4a and is tested as well with the four different mesh sizes for mesh convergence study. The L_∞ and L_2 relative errors are tabulated in Table 4.4.3 and the error convergences are plotted

in Figure 4.4.5.

$$z = 0.05 x \sin\left(\frac{\pi x}{5}\right) \sin\left(\frac{\pi z}{5}\right) \quad (4.4.3)$$



(a) Sinusoidal analytical functions tested on 2D sheet

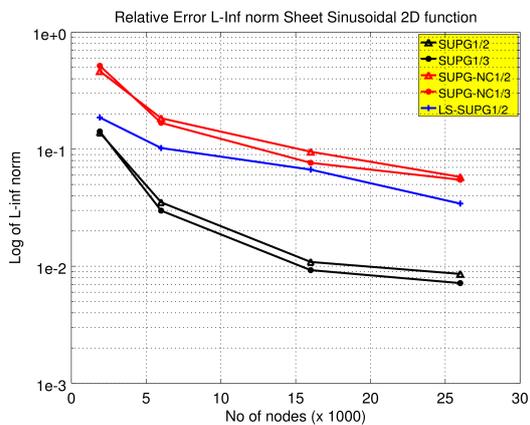


(b) Problem with non-conformance of boundary with the SUPG-NC method

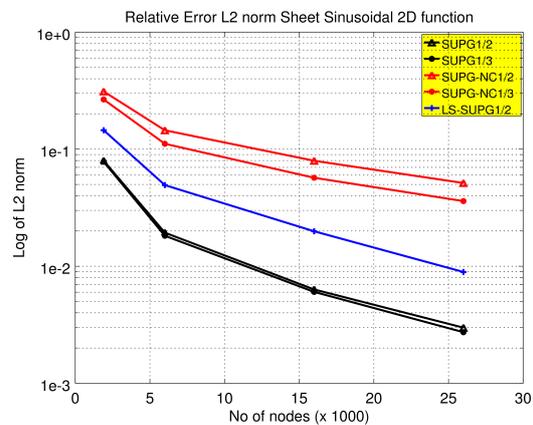
Figure 4.4.4: Sinusoidal analytical function and problems with SUPG-NC method

Nodes →		1900		6000		16000		26000	
Method	α	L_∞	L_2	L_∞	L_2	L_∞	L_2	L_∞	L_2
SUPG	1/2	13.85	7.98	3.52	1.94	1.09	0.63	0.86	0.30
	1/3	14.18	7.81	2.99	1.82	0.93	0.60	0.72	0.27
SUPG-NC	1/2	46.45	31.16	18.39	14.51	9.52	7.97	5.8	5.14
	1/3	51.59	26.6	16.8	11.13	7.65	5.7	5.48	3.6
LS-SUPG	1.0	18.65	14.53	10.24	4.94	6.69	1.99	3.43	0.89

Table 4.4.3: Comparison of L_2 and L_∞ % errors for the sinusoidal function deformation on a 2D flat sheet shown in Figure 4.4.4



(a) L_∞ error convergence



(b) L_2 error convergence

Figure 4.4.5: h convergence for 2D Sinusoidal function with different formulations for free-surface resolution

From the results, we observe that both errors are much larger in magnitude in comparison to the Gauss function, especially with the nodal condensation SUPG-NC formulation. However, the errors are seen to converge with mesh refinement. The SUPG method reassuringly proves to be the best in terms of accuracy and $\alpha = 1/3$ results in the improvement of accuracy, similar to what was observed with the case of Gauss analytical function. It is observed that the SUPG-NC method results in a larger error (i) at the locations with large slopes (ii) at edge boundaries where the boundary conditions are not very well imposed for the free-surface computation condition (see Figure 4.4.4b). The latter is not a big problem in the ForgeNxt[®]2016 algorithm as the boundary conditions are well imposed in the mechanical problem, which do not permit this phenomenon.

4.4.4 Test problems with 3D tubes

In this test, we deform two tubes of cylindrical and square shapes with two 3D analytical quadratic and sinusoidal functions on each. In comparison to the previous test-case, the cylindrical tube problem also requires only scalar free-surface correction (in 1D) but applied on a 3D geometry. Thus it does not have the problem related to the non-conformance of the boundary condition at the edge boundaries as seen with the 2D sheet. The 3D square tube is the test for the performance of 2-DoF or vector form of the free surface equation. The cylindrical tube has 1mm diameter and the square tube has an edge length of 1mm. Both measure 3mm in length (in the direction perpendicular to the cross-section of the tube). Three different meshes are tested for studying the error convergence.

4.4.4.1 Quadratic function deformation

First we apply a quadratic deformation separately on the 3D cylindrical and square tubes (see Figure 4.4.6) with the function represented in equation (4.4.4). The quadratic deformation results in a trumpet formation.

$$r = r_0 + 0.1x^2 \Rightarrow \begin{cases} y = y_0 + 0.1x^2 \frac{y_0}{\sqrt{y_0^2 + z_0^2}} \\ z = z_0 + 0.1x^2 \frac{z_0}{\sqrt{y_0^2 + z_0^2}} \end{cases} \quad (4.4.4)$$

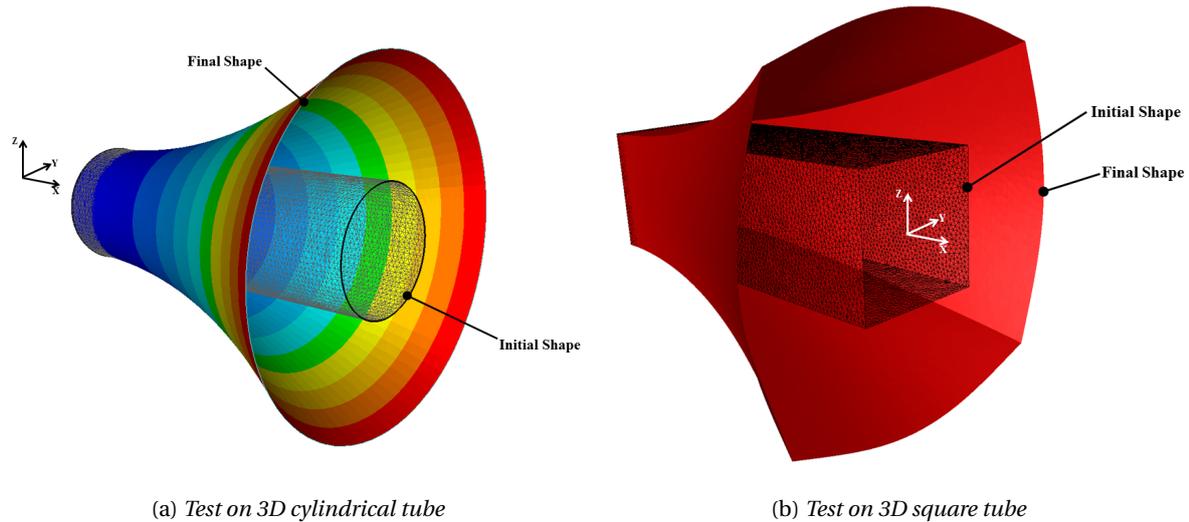


Figure 4.4.6: Quadratic analytical function tested on 3D cylindrical and square tubes

Cylindrical Tube (1-DoF correction)

The results for the cylindrical tube requires only scalar correction like the 2D rectangular sheet cases. The relative errors are tabulated in Table 4.4.4 and the error convergences with the different formulations are plotted in Figure 4.4.7. It is observed that the relative errors with all the

free-surface formulations are within 3%, even with the coarse mesh. The L_∞ error converges consistently with SUPG-NC as the mesh is refined, whereas the same is not true with the integral formulation as a larger error is seen with 4000 nodes in comparison to the error with 2000 nodes. However, the magnitude of the error with the fine mesh (4000 nodes) is still 10 times lesser than the SUPG-NC form. $\alpha = 1/3$ results in the improvement of accuracy as observed with the previous cases.

Nodes →		2000		4000		10000	
Method	α	L_∞	L_2	L_∞	L_2	L_∞	L_2
SUPG	1/2	0.38	0.25	0.4	0.15	0.17	0.08
	1/3	0.37	0.24	0.37	0.14	0.17	0.08
SUPG-NC	1/2	2.42	2.68	1.86	2.01	1.11	1.18
	1/3	1.74	1.77	1.29	1.34	0.82	0.78
LS-SUPG	1.0	0.41	0.25	0.62	0.26	0.23	0.1

Table 4.4.4: Comparison of L_2 and L_∞ % errors for the quadratic function deformation on a 3D cylindrical tube shown in Figure 4.4.6a

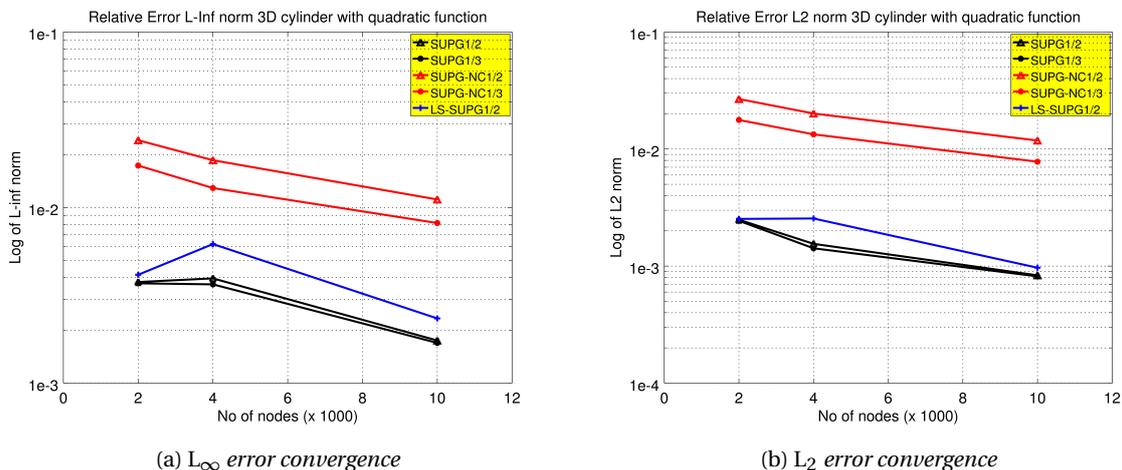


Figure 4.4.7: h convergence for quadratic deformation on 3D Cylindrical tube tested with different free-surface resolution formulations

Square Tube (2-DoF correction)

The square sectioned 3D tube consists of an edge (geometric singularity) and thus requires the vectorial formulation of the free-surface equation (both in the integral and the nodal form) to be resolved to compute the free surface boundary at the edge. The SUPG-dif formulation, as concluded by [2], results in an assured and faster convergence of the free surface solution. The relative errors and their convergence with the different mesh sizes are presented in Table 4.4.5 and Figure 4.4.8.

Nodes →		458		1814		10866	
Method	α	L_∞	L_2	L_∞	L_2	L_∞	L_2
SUPG	1/2	3.56	1.45	3.3	1.37	3.11	1.35
	1/3	3.51	1.44	3.27	1.37	3.11	1.35
SUPG-NC	1/2	8.77	7.77	5.52	4.31	3.98	2.43
	1/3	7.33	5.85	4.77	3.33	3.71	2.06
LS-SUPG	1.0	8.86	2.82	4.27	1.83	3.42	1.5

Table 4.4.5: Comparison of L_2 and L_∞ % errors for the quadratic function deformation on a 3D square tube shown in Figure 4.4.6b

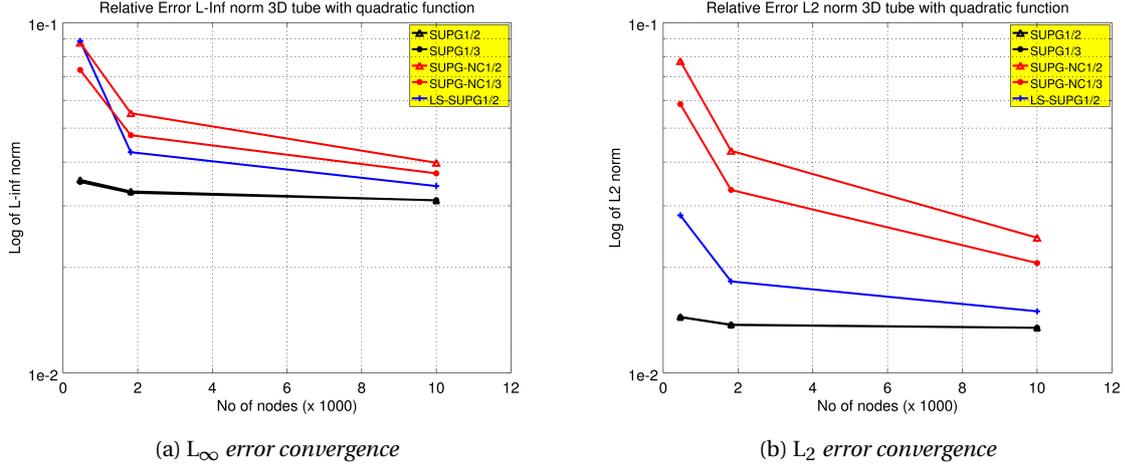


Figure 4.4.8: h convergence for quadratic deformation on 3D square tube tested with different free-surface resolution formulations

In general, the errors are much larger in magnitude with the singular geometry in comparison to the previous cases with scalar corrections, but the difference of error between the best SUPG and the worst SUPG-NC formulations reduces significantly, as the latter results in only 1.33 times the error in comparison to the former with the finer mesh. The SUPG stabilization coefficient $\alpha = 1/3$ improves the accuracy, especially with SUPG-NC formulation as seen for the previous cases and contrary to the previous cases, the L_2 error is lesser in comparison to the L_∞ . We observe a steep convergence of error with nodal formulation (SUPG-NC) in comparison to the integral formulations (LS-SUPG and SUPG).

4.4.4.2 Sinusoidal function

Next a sinusoidal deformation described with the function in equation (4.4.5) is applied on the cylindrical and square 3D tubes as shown in Figure 4.4.9a and Figure 4.4.9b respectively.

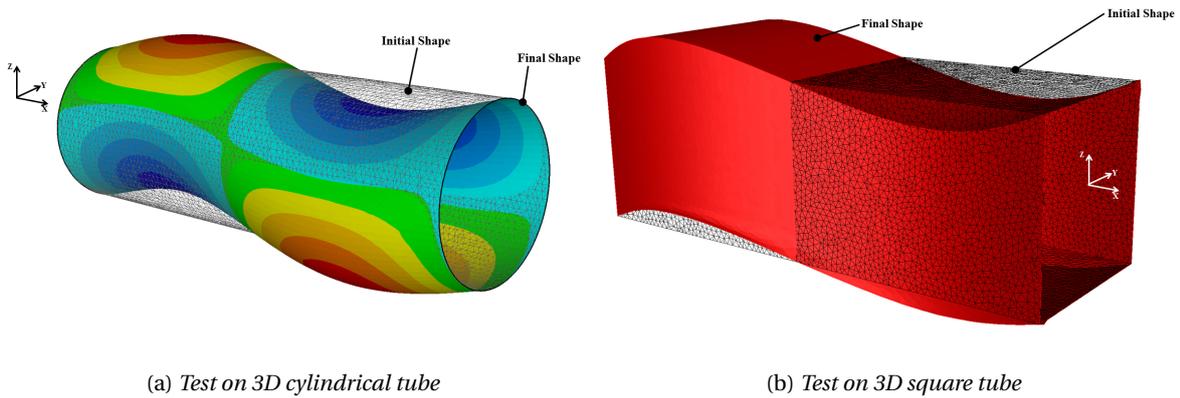


Figure 4.4.9: Sinusoidal analytical function tested on 3D cylindrical and square tubes

$$r = r_0 + 0.1x^2 \Rightarrow \begin{cases} y = y_0 + \sin\left(\frac{2\pi}{L}x\right) \\ z = z_0 + \sin\left(\frac{2\pi}{L}x\right) \end{cases} \quad (4.4.5)$$

Cylindrical tube (1-DoF correction)

Firstly, the results from the cylindrical tube are presented in Figure 4.4.10 and Table 4.4.6.

Nodes →		2000		4000		10000	
Method	α	L_∞	L_2	L_∞	L_2	L_∞	L_2
SUPG	1/2	3.27	1.29	3.77	1.47	3.34	1.33
	1/3	3.21	1.31	3.84	1.46	3.39	1.34
SUPG-NC	1/2	14.64	11.37	11.4	8.68	6.62	5.34
	1/3	10.07	7.7	7.7	5.88	4.66	3.72
LS-SUPG	1.0	4.13	1.54	3.86	1.53	3.78	1.37

Table 4.4.6: Comparison of L_2 and L_∞ % errors for the sinusoidal function deformation on a 3D cylindrical tube shown in Figure 4.4.9a

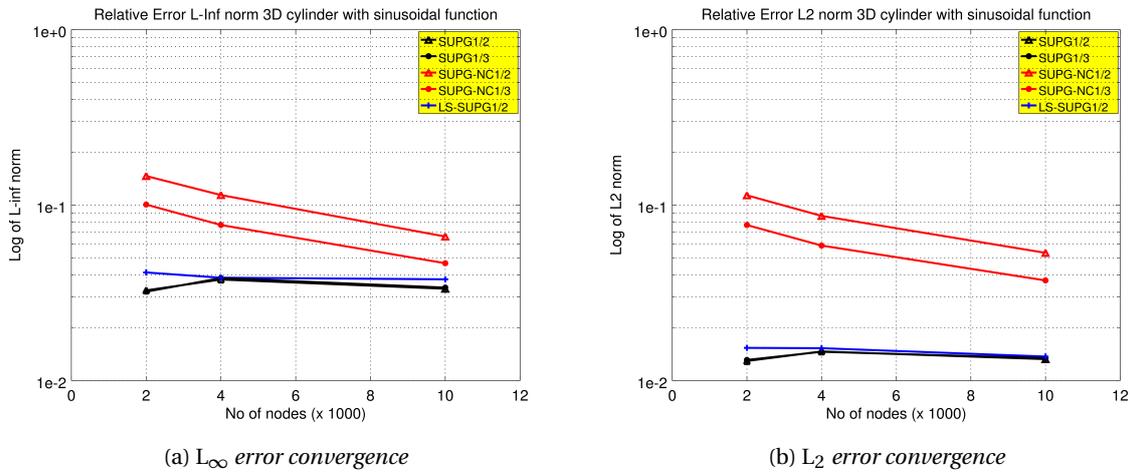


Figure 4.4.10: h convergence for sinusoidal deformation on 3D cylindrical tube tested with different free-surface resolution formulations

The error convergence rate with the integral formulation is almost zero. The SUPG-NC formulation, on the other hand, shows steeper convergence rate, but the magnitude of error with this method is still higher in comparison to the integral formulations. The magnitude of error reduces with $\alpha = 1/3$.

Square tube (2-DoF correction)

Lastly, the sinusoidal function is applied on the square tube as shown in Figure 4.4.9b. The error results are presented in Table 4.4.7 and the error convergences are plotted in Figure 4.4.11.

Nodes →		458		1814		10866	
Method	α	L_∞	L_2	L_∞	L_2	L_∞	L_2
SUPG	1/2	15.37	4.66	14.69	2.01	9.28	0.59
	1/3	14.3	4.2	14	1.89	8.99	0.55
SUPG-NC	1/2	37.64	29.61	17.72	14.48	9.32	5.67
	1/3	24.63	20.19	14.12	9.86	8.98	3.82
LS-SUPG	1.0	26.74	7.26	17.23	3.82	8.3	1.11

Table 4.4.7: Comparison of L_2 and L_∞ % errors for the sinusoidal function deformation on a 3D square tube shown in Figure 4.4.10b

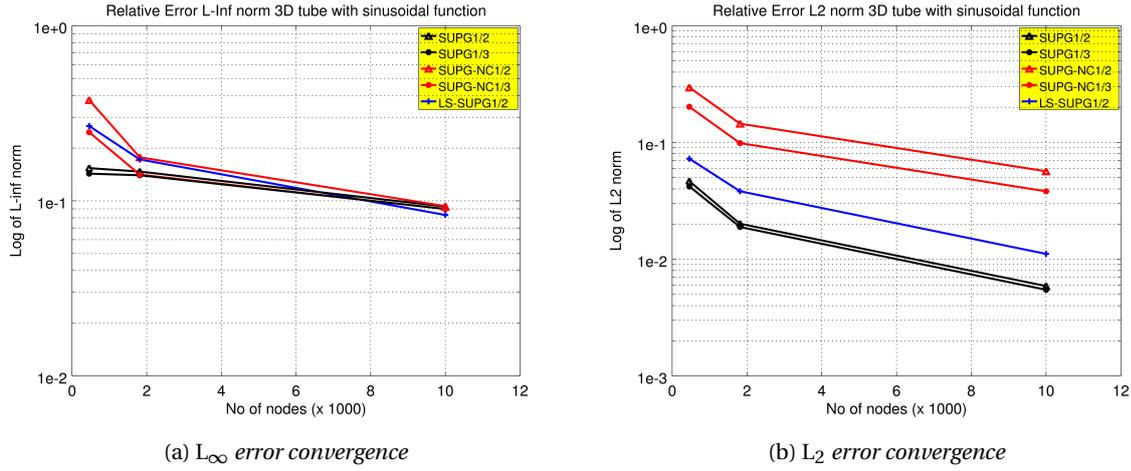


Figure 4.4.11: h convergence for sinusoidal deformation on 3D square tube tested with different free-surface resolution formulations

Most of the observations in the previous tests are re-established except for the fact that the L_2 error with the SUPG-NC is lesser in comparison to the L_∞ error. The rate of convergence of L_2 error is similar with all the formulations.

4.4.5 Conclusions from analytical test-cases

The conclusions from the results from the analytical test problems discussed in Section 4.4.3 and Section 4.4.4 are highlighted below.

1. **Error convergence:** For all the presented free-surface formulations in the integral form (with LS-SUPG, SUPG test functions) and nodal form (SUPG-NC) the relative errors are convergent for most of the cases. The error convergence rate for the method is found to be problem dependent, for example SUPG and LS-SUPG methods converge better than SUPG-NC formulations for the 1-DoF correction case, whereas for the 2-DoF case the convergence-rate for SUPG-NC is similar or better than the SUPG and/or LS-SUPG functions.
2. **Accuracy:** For the 2D or 3D test-cases, the integral formulation with SUPG test function is undoubtedly the most accurate method for computing the free-surface flows. The nodal form of the free-surface equation (SUPG-NC) is the worst in terms of the accuracy of the solution.
3. **SUPG Stabilization coefficient (α):** The SUPG stabilization factor $\alpha = 1/3$ results in better accuracy in comparison to the $\alpha = 1/2$ especially with SUPG-NC formulation. For the integral formulation with SUPG test function, there is no or minor improvement of accuracy with this change. Nonetheless, it has no effect on the rate of the convergence.
4. **From the results, we conclude that the integral formulation with SUPG function is both convergent and more accurate in comparison to the other formulations tested. It is thus very important to conserve this method for free surfaced computation. On the other hand, SUPG-NC form of the free surface equation, even though less accurate results in error reduction with mesh refinement. This form of the equation is absolutely necessary for finding a consistent contact condition for the mechanical problem, which is discussed in the next Section 4.5. It is also to be pointed out that the errors are reduced significantly with dense meshes, which is generally the case for the contact region. The choice of $\alpha = 1/3$ further improves the accuracy of the solution.**

4.5 Updated contact coupling

In continuation to the presentation of *nodal condensation* technique in the Section 4.2, we look to find a consistent nodal contact constraint for the mechanical problem, similar to the existing one in the equation (4.1.4). The new nodal constraint is derived from the nodal form of the free-surface equation (4.3.2). To do so, we use the same analysis for the contact surface nodes k , as explained in Section 4.1. For the contact node, the penalty term is non-negative, hence results in the condition (4.5.1a).

$$\forall k \in \Gamma_{ch} \quad \vec{v}_k \cdot \hat{u}_k(\vec{x}) \leq 0 \quad \text{with } \hat{u}_k = \vec{u}_k / |\vec{u}_k| \quad (4.5.1a)$$

$$\Phi_p'(\vec{v}) = \Phi'(\vec{v}) + \rho_c^{(1)} s_k [\vec{v}_k \cdot \hat{u}_k(\vec{x})]^+ \hat{u}_k(\vec{x}) \quad (4.5.1b)$$

The equation (4.5.1a) represents the constraint on the normal velocity which must be verified on the contact nodes in the mechanical resolution. Hence, for a node which remains on the contact surface must have zero normal velocity, and the one leaving the contact would find a negative normal velocity. For the latter, the free surface must be corrected to verify the free surface condition (zero normal velocity) in the Step-2 of the algorithm. In conclusion, it is not wrong to say that ideally we apply free surface conditions on the complete boundary whether or not in contact. The derived contact condition is a unilateral constraint on the normal velocity at the contact surface $\Gamma_c^{(1)}$ computed in the Step-2 of the previous fixed-point iteration (non-initialization case). It is to be noted here that we apply the constraint on the normal velocity computed using the *unit upwind normal* \hat{u}_k instead of the tool normal \vec{n}^{tool} in the existing formulation (4.1.4). The unilateral condition is applied with the penalty method, and the associated penalty functional is used to enrich the mechanical functional Φ . The resolution of the velocity aims to minimize the functional Φ_p in the equation (4.5.1b). The velocity field computed in the mechanical resolution is used to update the Lagrangian multiplier $\tilde{\lambda}_k$ from the normal velocity as represented in the equation (4.5.2a).

$$\tilde{\lambda}_k = -\rho_c^{(1)} s_k [\vec{v}_k \cdot \hat{u}_k(\vec{x})]^+ \quad (4.5.2a)$$

$$\Gamma_c^{(2)} = \{k \in \Gamma, \quad \tilde{\lambda}_k < 0\} \quad (4.5.2b)$$

The updated Lagrangian multiplier is used to update the contact surface $\Gamma_c^{(2)}$ with equation (4.5.2b). A node k is added to the contact surface $\Gamma_c^{(2)}$ if the $\tilde{\lambda}_k < 0$ and allowed to leave the contact if $\tilde{\lambda}_k = 0$. The new contact surface, like in the existing algorithm, is used in free surface computation where bilateral contact conditions are applied during the surface correction.

4.5.1 Explicit form of contact constraint

As mentioned, the new contact constraint in equation (4.5.1a) is described on the normal velocity, and the surface normal used in the formulation is computed on the $\vec{x}^{(i)}$ configuration computed after the free-surface computation step of the previous fixed-point iteration i . Hence, the contact constraint is explicit:

$$\forall k \in \Gamma_c^{(1)} \quad h(\vec{v}_k) = \vec{v}_k \cdot \hat{u}_k(\vec{x}^{(i)}) \leq 0 \quad (4.5.3)$$

For the implementation of this contact condition, we must only replace the tool normal \vec{n}^{tool} used in the existing formulation with the *unit upwind normal* \hat{u}_k . The *upwind normal* is computed as following.

$$\forall k \in \Gamma_{c,h} \quad \vec{u}_k(\vec{x}^{(i)}) = \int_{\Gamma_{c,h}} N_k^{SUPG} \vec{n}(\vec{x}^{(i)}) ds = \sum_{f \in \mathcal{I}_k} \int_f N_k^{SUPG} \vec{n}^f(\vec{x}^{(i)}) ds^f \quad (4.5.4a)$$

$$\vec{n}^f(\vec{x}^{(i)}) = \frac{\vec{u}^f(\vec{x}^{(i)})}{|\vec{u}^f(\vec{x}^{(i)})|} \quad (4.5.4b)$$

In the above formulation, \vec{n}^f is the *unit surface normal* of the facet element f found from the normalization of the *surface normal* \vec{u}^f (see equation (4.5.4b)). The *surface normal* is computed

from the reference coordinates (ξ, η) of the element using equation (4.5.5), with ds^f being the surface area of the element.

$$\vec{u}^f(\vec{x}^{(i)}) = \frac{\partial \vec{x}^{(i)}}{\partial \xi} \times \frac{\partial \vec{x}^{(i)}}{\partial \eta}; \quad \vec{x}^{(i)} = \sum_m N_m \vec{x}_m^{(i)} \quad (4.5.5a)$$

$$ds^f = |\vec{u}^f(\vec{x})| d\xi d\eta \quad (4.5.5b)$$

Explicit contact constraint with geometric singularities

In the case of geometries with singularities (like edges), the new contact formulation must as well be derived conforming to the 2-DoF free-surface formulation. In order to do so, we use the SUPG-NC-dif nodal form in equation (4.3.3). For the treatment of contact, we assume that the system $(\vec{d}_k^{[1]}, \vec{d}_k^{[2]})$ is orthogonal, saying that:

$$\vec{d}_k^{[1]} \cdot \vec{d}_k^{[2]} = 0 \quad (4.5.6)$$

We now assume a change of reference to (\hat{n}, \hat{m}) where \hat{n} is a vector in the direction of the tool normal \vec{n}^{tool} and \hat{m} is the complementary vector of the basis in the plane perpendicular to the principal flow direction. The position vectors of a node $k \in \Gamma_{edge}$ can be written in the new basis as:

$$\begin{pmatrix} \vec{x}_k^{[\hat{n}]} \\ \vec{x}_k^{[\hat{m}]} \end{pmatrix} = \begin{pmatrix} \cos \theta_k & \sin \theta_k \\ -\sin \theta_k & \cos \theta_k \end{pmatrix} \begin{pmatrix} \vec{x}_k^{[1]} \\ \vec{x}_k^{[2]} \end{pmatrix} \quad (4.5.7)$$

In the equation above θ_k is the angle between \vec{n}_k^{tool} and $\vec{d}_k^{[1]}$. Hence, from the relation (4.5.6), we can state that:

$$\begin{cases} \vec{d}_k^{[1]} \cdot \vec{n}_k^{tool} = \vec{d}_k^{[2]} \cdot \hat{m}_k = \cos \theta \\ \vec{d}_k^{[2]} \cdot \vec{n}_k^{tool} = \vec{d}_k^{[1]} \cdot \hat{m}_k = \sin \theta \end{cases} \quad (4.5.8)$$

Subsequently, we describe the residuals $\tilde{S}_k^{[1]}, \tilde{S}_k^{[2]}$ from (4.3.5) in the new reference frame (\hat{n}, \hat{m}) with the following operation:

$$\begin{pmatrix} \tilde{S}_k^{[\hat{n}]} \\ \tilde{S}_k^{[\hat{m}]} \end{pmatrix} = \begin{pmatrix} \vec{d}_k^{[1]} \cdot \vec{n}_k^{tool} & \vec{d}_k^{[2]} \cdot \vec{n}_k^{tool} \\ -\vec{d}_k^{[2]} \cdot \vec{n}_k^{tool} & \vec{d}_k^{[1]} \cdot \vec{n}_k^{tool} \end{pmatrix} \begin{pmatrix} \tilde{S}_k^{[1]} \\ \tilde{S}_k^{[2]} \end{pmatrix} \quad (4.5.9)$$

From equations (4.3.3) and (4.5.9), we find the residuals in the new frame of reference given as $\tilde{S}_k^{[\hat{n}]}$ and $\tilde{S}_k^{[\hat{m}]}$ and represented with the relation (4.5.10a). When a node $k \in \Gamma_{edge}$ is in contact, the penalty constraint in the equation (4.5.10a) is active, meaning $\rho_c^{(2)} s_k [\bar{h}(t_k)]^+ \geq 0$ and hence we find the inequality (4.5.10b). Thus, for such a node $k \in \Gamma_{edge} \cap \Gamma_c$, we impose this inequality as the contact constraint for the velocity computation.

$$\forall k \in \Gamma_{edge}$$

$$\begin{aligned} \tilde{S}_k^{[\hat{n}]} &= \vec{v}_k \cdot \vec{u}_k^{[1]}(\vec{x}^{(i)}) \left(\vec{d}_k^{[1]} \cdot \vec{n}_k^{tool} \right) + \vec{v}_k \cdot \vec{u}_k^{[2]}(\vec{x}^{(i)}) \left(\vec{d}_k^{[2]} \cdot \vec{n}_k^{tool} \right) \dots \\ &\dots + \rho_c^{(2)} s_k [\bar{h}(t_k)]^+ \underbrace{\left(\left(\vec{d}_k^{[1]} \cdot \vec{n}_k^{tool} \right)^2 + \left(\vec{d}_k^{[2]} \cdot \vec{n}_k^{tool} \right)^2 \right)}_{=1} = 0 \end{aligned} \quad (4.5.10a)$$

$$\forall k \in \Gamma_{edge} \cap \Gamma_c \quad \vec{v}_k \cdot \vec{u}_k^{[1]}(\vec{x}^{(i)}) \left(\vec{d}_k^{[1]} \cdot \vec{n}_k^{tool} \right) + \vec{v}_k \cdot \vec{u}_k^{[2]}(\vec{x}^{(i)}) \left(\vec{d}_k^{[2]} \cdot \vec{n}_k^{tool} \right) \leq 0 \quad (4.5.10b)$$

The residual in direction \hat{m} is given with the relation (4.5.11a), which can be simplified with the equation (4.5.11b).

$$\begin{aligned} \tilde{S}_k^{[\hat{m}]} &= \vec{v}_k \cdot \vec{u}_k^{[1]}(\vec{x}^{(i)}) \left(-\vec{d}_k^{[2]} \cdot \vec{n}_k^{tool} \right) + \vec{v}_k \cdot \vec{u}_k^{[2]}(\vec{x}^{(i)}) \left(\vec{d}_k^{[1]} \cdot \vec{n}_k^{tool} \right) \dots \\ &\dots + \rho_c^{(2)} s_k [\bar{h}(t_k)]^+ \underbrace{\left(-\left(\vec{d}_k^{[2]} \cdot \vec{n}_k^{tool} \right) \left(\vec{d}_k^{[1]} \cdot \vec{n}_k^{tool} \right) + \left(\vec{d}_k^{[1]} \cdot \vec{n}_k^{tool} \right) \left(\vec{d}_k^{[2]} \cdot \vec{n}_k^{tool} \right) \right)}_{=0} = 0 \end{aligned} \quad (4.5.11a)$$

$$\vec{v}_k \cdot \vec{u}_k^{[1]}(\vec{x}^{(i)}) \left(\vec{d}_k^{[2]} \cdot \vec{n}_k^{tool} \right) = \vec{v}_k \cdot \vec{u}_k^{[2]}(\vec{x}^{(i)}) \left(\vec{d}_k^{[1]} \cdot \vec{n}_k^{tool} \right) \quad (4.5.11b)$$

In the case if $\vec{d}_k^{[1]} \cdot \vec{d}_k^{[2]} \neq 0$, the residual in \hat{n} is written as (4.5.12). When in contact, as the sum of squares is a positive number, the term on the right is non-negative. Thus, we again retrieve the inequality (4.5.10b).

$$\begin{aligned} \tilde{S}^{[\hat{n}]} &= \vec{v}_k \cdot \vec{u}_k^{[1]}(\vec{x}^{(i)}) \left(\vec{d}_k^{[1]} \cdot \vec{n}_k^{tool} \right) + \vec{v}_k \cdot \vec{u}_k^{[2]}(\vec{x}^{(i)}) \left(\vec{d}_k^{[2]} \cdot \vec{n}_k^{tool} \right) \dots \\ &\quad \dots + \underbrace{\rho_c^{(2)} s_k [\bar{h}(t_k)]^+ \left(\underbrace{\left(\vec{d}_k^{[1]} \cdot \vec{n}_k^{tool} \right)^2 + \left(\vec{d}_k^{[2]} \cdot \vec{n}_k^{tool} \right)^2}_{\in [0,1]} \right)}_{\geq 0} = 0 \end{aligned} \quad (4.5.12)$$

It can also be noted that the constraint (4.5.10b) in the explicit form can be written as the scalar product of \vec{v}_k and $\vec{u}_k(\vec{x}^{(i)})$, with the latter being the weighted average of the respective *upwind normal* components in the directions $\vec{d}_k^{[1]}$ and $\vec{d}_k^{[2]}$. Hence, we can represent the explicit contact constrain in a generalized manner as in equation (4.5.2a).

$$h(\vec{v}_k) = \vec{v}_k \cdot \hat{u}_k(\vec{x}^{(i)}) \leq 0 \quad (4.5.13a)$$

$$\hat{u}_k(\vec{x}^{(i)}) = \frac{\vec{u}_k^{[1]}(\vec{x}^{(i)})}{|\vec{u}_k^{[1]}(\vec{x}^{(i)})|} \left(\vec{d}_k^{[1]} \cdot \vec{n}_k^{tool} \right) + \frac{\vec{u}_k^{[2]}(\vec{x}^{(i)})}{|\vec{u}_k^{[2]}(\vec{x}^{(i)})|} \left(\vec{d}_k^{[2]} \cdot \vec{n}_k^{tool} \right) \quad (4.5.13b)$$

Updated contact hessian in mechanical description

The updated contact hessian matrix can be derived with the derivative of the functional in (4.5.1b) w.r.t. velocity.

$$\forall k \in \Gamma_c^{(2)}, \quad \hat{i}, \hat{j} \in [1, 3] \quad H_{k, \hat{i}, k, \hat{j}}(\vec{v}) = \rho_c^{(1)} s_k \begin{cases} \vec{u}_{k, \hat{i}} \otimes \vec{u}_{k, \hat{j}} & \text{if } h(\vec{v}_k) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4.5.14)$$

The parallel order implementation of the explicit constraint is easy. The upwind normal and the facet surface area are firstly computed at each node of the triangular facet of the partitioned mesh by individual processor and summed. Each processor then sends its individual contributions of the summed upwind normal computed in each mesh partition to the root processor for summation of the contributions at the interface nodes. The updated sum is then communicated back to the non root processors. After communicating, the upwind normals are normalized.

4.5.2 Implicit form of contact constraint

An implicit form of the consistent contact constraint is possible with the prediction of the current configuration $\vec{x}^{(i+1)}$ before it is actually computed in the free surface computation. In the incremental formulation, an implicit contact constraint is imposed (2.3.17) which ensures the velocity field complies with the impenetrability condition on the current configuration. Similarly, the aim of the present section is to derive an implicit contact condition for the steady-state formulation. Hence, the computed velocity field is found on the current configuration. In the existing formulation, the impenetrability condition in the free surface computation (3.3.2b) represents an implicit contact constraint defined on the current configuration. To obtain an implicit constraint in velocity, this condition must be used as the starting point, and we must be able to describe the velocity in terms of correction. However, these two are strongly coupled in the free surface equation. So we must first decouple the velocity \vec{v} and correction \vec{t} fields. For doing so, we invoke the nodal

form of free-surface equation (4.3.2) without the penalty constraint. The free-surface condition is an implicit equation as the upwind normal \vec{u}_k is described on the current configuration $\vec{x}^{(i+1)}$.

$$\forall k \in \Gamma_h, \quad \vec{v}_k \cdot \vec{u}_k(\vec{x}^{(i+1)}) = 0 \quad (4.5.15)$$

Using Taylor's expansion, we can write the *upwind normal* $\vec{u}_k(\vec{x}^{(i+1)})$ as following:

$$\forall k \in \Gamma_h, \quad \vec{u}_k(\vec{x}^{(i+1)}) = \vec{u}_k(\vec{x}^{(i)}) + \underbrace{\frac{d\vec{u}_k}{dt} t + \frac{d^2\vec{u}_k}{d^2t} \frac{t^2}{2}}_{\delta\vec{u}_k} \dots \quad (4.5.16)$$

Here we simplify by assuming only a linear dependence of the *upwind normal* on the correction field t and neglecting the higher order terms. Hence, we find the change in upwind normal $\delta\vec{u}_k$ as following:

$$\delta\vec{u}_k = \sum_{\hat{l} \in P_k} D\vec{u}_{k\hat{l}} t_{\hat{l}}^{(i+1)} = 0 \quad \text{where } D\vec{u}_{k\hat{l}} = \frac{\partial \vec{u}_k}{\partial t_{\hat{l}}} \quad (4.5.17)$$

In the above description \hat{l} is the index of nodes in the patch \mathfrak{P}_k of surface elements around node k as shown in Figure 4.5.1. Next we replace the implicitly defined upwind normal $\vec{u}_k(\vec{x}^{(i+1)})$ in the equation (4.5.15) with the linearized form obtained from (4.5.16) and (4.5.17) and find the following relation:

$$\forall k \in \Gamma_h, \quad \vec{v}_k \cdot \vec{u}_k(\vec{x}^{(i)}) + \vec{v}_k \cdot \sum_{\hat{l} \in P_k} D\vec{u}_{k\hat{l}} t_{\hat{l}}^{(i+1)} = 0 \quad (4.5.18)$$

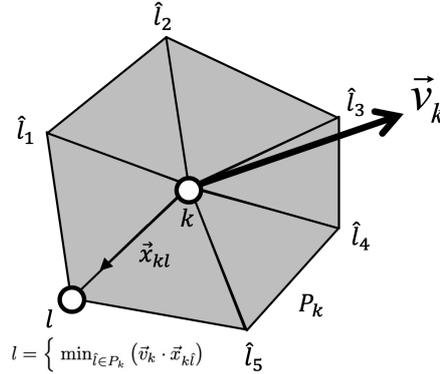


Figure 4.5.1: Location of node l for implicit formulation on 3D (surface) mesh

The equation (4.5.18) is still strongly coupled in shape (correction field) and velocity even after *nodal condensation*. It is quintessential to decouple velocity and shape to find an explicit relation between the two for finding the implicit contact constraint. We can decouple the fields using either of the following ideas:

1. Apply a second order *nodal condensation* of the matrix $D\vec{u}_{k\hat{l}}$ to find the vector $D\vec{u}_{kk}$ defined for each node of the mesh
2. Assume a single node dependency for the normal gradient matrix $D\vec{u}_{k\hat{l}}$ on the node l in order to reduce the matrix to a unitary rank $D\vec{u}_{kl}$.

It is to be impressed to the reader that the *nodal condensation* of the matrix $D\vec{u}_{k\hat{l}}$ leads to the loss of implicitness (see further in this section), so, we choose the latter. The single dependency node with index l is chosen such that it is found upwind to, and preferably on the same streamline as, node k (see Figure 4.5.1). On the element patch \mathfrak{P}_k , we can algebraically find l as:

$$\forall k \in \Gamma_c, \quad l = \left\{ \hat{l} \in \mathfrak{P}_k \text{ such that } \min_{\hat{l}} \left(\frac{\vec{v}_k \cdot \vec{x}_{k\hat{l}}}{|\vec{v}_k| |\vec{x}_{k\hat{l}}|} \right) \right\} \quad (4.5.19)$$

This choice for l results in the nodal free-surface equation (4.5.18) which is similar to backward integration with the streamline approach. This is also consistent with the choice of SUPG function used for the free-surface resolution, which gives higher weightage to the upwind element. With this choice, we can describe the nodal form of free-surface correction equation with single dependency as following.

$$\forall k \in \Gamma \text{ and } l \text{ from (4.5.19), } \quad \vec{v}_k \cdot \vec{u}_k(\vec{x}^{(i)}) + \vec{v}_k \cdot D\vec{u}_{kl} t_l^{(i+1)} = 0 \quad (4.5.20a)$$

$$\Rightarrow \forall k \in \Gamma \text{ and } l \text{ from (4.5.19), } \quad t_l^{(i+1)} = \frac{\vec{v}_k \cdot \vec{u}_k(\vec{x}^{(i)})}{-\vec{v}_k \cdot D\vec{u}_{kl}} \quad (4.5.20b)$$

The new assumption (single dependency) enables us to separate the velocity and shape correction fields and hence describe the correction field t_l as an explicit relation in velocity as represented in the equation (4.5.20b). It can be verified here that the denominator term $\vec{v}_k \cdot D\vec{u}_{kl} < 0$.

Implicit contact condition

From the material impenetrability condition in equation (3.3.2b), the correction t_l respect the impenetrability condition (4.5.21a). Next we replace the correction t_l defined explicitly in velocity in the equation (4.5.20b) into the impenetrability condition (4.5.21a). This results in the implicit constraint on velocity that we are looking for.

$$l \text{ from (4.5.19), } \quad \delta_l \geq 0 \quad \Rightarrow \bar{h}(t_l) = t_l \vec{d}_l \cdot \vec{n}_l^{tool} - \delta_l(\vec{x}^{(i)}) \geq 0 \quad (4.5.21a)$$

$$\forall k \in \Gamma_c \text{ and } l \text{ from (4.5.19), } \quad \vec{v}_k \cdot \vec{u}_k(\vec{x}^{(i)}) - \frac{\delta_l(\vec{x}^{(i)})}{\vec{d}_l \cdot \vec{n}_l^{tool}} (-\vec{v}_k \cdot D\vec{u}_{kl}) \leq 0 \quad (4.5.21b)$$

It can be clearly seen that in the implicit contact constraint in (4.5.21b), the term on the right gives the implicit character to the constraint. Since l is ideally on the same streamline as k due to the upwind definition, and that we are in the context of steady-state, l represents the state of node k before coming in contact with the tool. *This is why the impenetrability condition (4.5.21a) at the node l leads to an implicit contact constraint. Had we chosen lumping/nodal condensation of the matrix $D\vec{u}_{kl} \rightarrow D\vec{u}_{kk}$, instead of dependency on l while arriving at the transformation in (4.5.20a) from (4.5.18), there would be no implicitness in the contact constraint, as the term on the right in equation (4.5.21b) would become zero ($\delta_k = 0$ for $k \in \Gamma_c$).* From (4.5.21b) we can also infer that for each contact node k , whose dependency node l is also in contact ($\delta_l = 0$), the implicit character of the constraint is lost (see Figure 4.5.2a). This is true for the nodes of the workpiece inside the tool surface including the contact exit edge. The implicit constraint is hence active only on the nodes whose dependency nodes l are not in contact with the tool (see Figure 4.5.2b) and targets principally the first contact line on the workpiece free-surface, which is generally the region that contributes to the contact instability and the observed problem in the existing formulation.

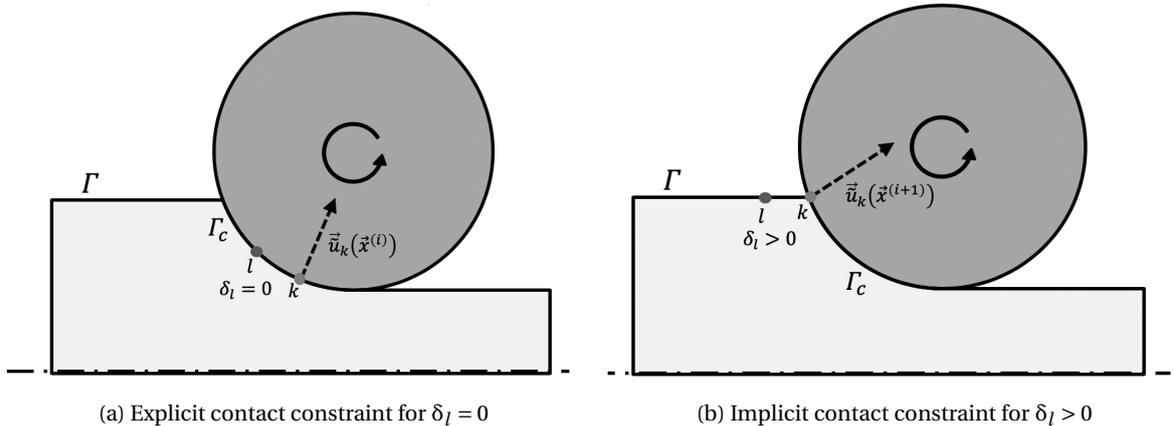


Figure 4.5.2: Implementation of Implicit contact constraint

One can also identify the similarity between the steady-state implicit contact constraint (4.5.21b) and the contact constraint for incremental formulation in (2.3.17). We rewrite the contact condition in incremental form for ease of referring.

$$\forall k \in \Gamma_c, \quad \left(\vec{v}_k^t - \vec{v}_k^{tool} \right) \cdot \vec{n}_k^{tool} - \frac{\delta_k^t}{\Delta t} \leq 0 \quad (4.5.22)$$

In the steady-state formulation, since there is no time, we have the representation of time with the pseudo time-increment $\Delta \vec{t} \approx (-\vec{v}_k \cdot D\vec{u}_{kl})^{-1}$. Thus, the inequality (4.5.21b) holds true as long as $\vec{v}_k \cdot D\vec{u}_{kl} < 0$.

Implicit contact constraint with geometric singularities

The implicit contact constraint must as well be found for singular geometries. We can do so in a manner similar to the explicit form of the contact constraint (4.5.13a), and we use the same process as done for finding the scalar form of the implicit contact constraint. The starting point of finding the constraint is the 2-DoF nodal form of the free-surface equation (4.3.3), but we do not take into account the penalty term:

$$\forall k \in \Gamma_{edge} \begin{cases} \vec{v}_k \cdot \vec{u}_k^{[1]}(\vec{x}) = 0 \\ \vec{v}_k \cdot \vec{u}_k^{[2]}(\vec{x}) = 0 \end{cases} \quad (4.5.23)$$

The respective upwind normals $\vec{u}_k^{[1]}$ and $\vec{u}_k^{[2]}$ can be linearized, similar to the scalar equation (4.5.18) as following:

$$\forall k \in \Gamma_{edge} \begin{cases} \vec{u}_k^{[1]}(\vec{x}^{(i+1)}) = \vec{u}_k^{[1]}(\vec{x}^{(i)}) + \sum_{\hat{l} \in P_k} D\vec{u}_{k\hat{l}}^{[1]} t_{\hat{l}}^{[1](i+1)} & \text{where } D\vec{u}_{k\hat{l}}^{[1]} = \partial \vec{u}_k^{[1]} / \partial t_{\hat{l}}^{[1]} \\ \vec{u}_k^{[2]}(\vec{x}^{(i+1)}) = \vec{u}_k^{[2]}(\vec{x}^{(i)}) + \sum_{\hat{l} \in P_k} D\vec{u}_{k\hat{l}}^{[2]} t_{\hat{l}}^{[2](i+1)} & \text{where } D\vec{u}_{k\hat{l}}^{[2]} = \partial \vec{u}_k^{[2]} / \partial t_{\hat{l}}^{[2]} \end{cases} \quad (4.5.24)$$

As done for the scalar form, we invoke the single dependency to the upwind node l , which is likely an edge node with two correction components $(t_l^{[1]}, t_l^{[2]})$:

$$\forall k \in \Gamma_{edge} \begin{cases} \vec{u}_k^{[1]}(\vec{x}^{(i+1)}) \approx \vec{u}_k^{[1]}(\vec{x}^{(i)}) + D\vec{u}_{kl}^{[1]} t_l^{[1](i+1)} \\ \vec{u}_k^{[2]}(\vec{x}^{(i+1)}) \approx \vec{u}_k^{[2]}(\vec{x}^{(i)}) + D\vec{u}_{kl}^{[2]} t_l^{[2](i+1)} \end{cases} \quad (4.5.25)$$

It is also impressed here that only a unique upwind node l is possible for the edge node k in both the directions. So, we can compute $(t_l^{[1]}, t_l^{[2]})$ from the free-surface equation (4.5.25) and linearized definition of upwind normals in (4.5.27):

$$\forall k \in \Gamma_{edge} \quad t_l^{[1](i+1)} = \frac{\vec{v}_k \cdot \vec{u}_k^{[1]}(\vec{x}^{(i)})}{-\vec{v}_k \cdot D\vec{u}_{kl}^{[1]}} \quad (4.5.26a)$$

$$\forall k \in \Gamma_{edge} \quad t_l^{[2](i+1)} = \frac{\vec{v}_k \cdot \vec{u}_k^{[2]}(\vec{x}^{(i)})}{-\vec{v}_k \cdot D\vec{u}_{kl}^{[2]}} \quad (4.5.26b)$$

Thus the impenetrability constraint at the node $l \in \Gamma_{edge}$ is written using equations (3.3.1) and (3.3.2b) as following:

$$\left(t_l^{[1]} \vec{d}_l^{[1]} + t_l^{[2]} \vec{d}_l^{[2]} \right) \cdot \vec{n}_l^{tool} - \delta_l \leq 0 \quad (4.5.27)$$

Using the definition of $(t_l^{[1]}, t_l^{[2]})$ from (4.5.26) and the impenetrability constraint at node l in equation (4.5.27), we find the implicit contact constraint in the 2-DoF form as following:

$$\vec{v}_k \cdot \left(\vec{u}_k^{[1]}(\vec{x}^{(i)}) \frac{\vec{d}_l^{[1]} \cdot \vec{n}_l^{tool}}{-\vec{v}_k \cdot D\vec{u}_{kl}^{[1]}} + \vec{u}_k^{[2]}(\vec{x}^{(i)}) \frac{\vec{d}_l^{[2]} \cdot \vec{n}_l^{tool}}{-\vec{v}_k \cdot D\vec{u}_{kl}^{[2]}} \right) \leq \delta_l \quad (4.5.28)$$

Implementation of implicit contact condition

For the implementation of the implicit contact constraint, we must ensure that its form is similar to the one described in equation (4.5.1a). Thus the normalized upwind normal must be used such that the constraint is purely in velocity dimension. The condition (4.5.21) consists of two principal terms: the explicit term $\vec{u}_k \rightarrow \hat{u}_k$ and the implicit term $D\hat{u}_{kl}$. We normalize the explicit upwind normal as described in the equation (4.5.1a) and for the implicit part, we normalize it with the equation (4.5.29a). The resultant normalized implicit constraint is given with the equation (4.5.21b).

$$D\hat{u}_{kl} = \frac{\partial \hat{u}_k(\vec{x})}{\partial t_l} = D\vec{u}_{kl} - \frac{\vec{u}_k(\vec{x})}{|\vec{u}_k(\vec{x})|} \left(\frac{\vec{u}_k(\vec{x})}{|\vec{u}_k(\vec{x})|} \cdot D\vec{u}_{kl} \right) \quad (4.5.29a)$$

$$\forall k \in \Gamma_c \text{ and } l \text{ from (4.5.21), } h(\vec{v}_k) = \vec{v}_k \cdot \hat{u}_k(\vec{x}^{(i)}) \left(\vec{d}_l \cdot \vec{n}_l^{tool} \right) - \delta_l(\vec{x}^{(i)}) \left(-\vec{v}_k^{(i)} \cdot D\hat{u}_{kl} \right) \leq 0 \quad (4.5.29b)$$

The implementation of the gradient term $D\vec{u}_{kl}$ is done as follows:

$$D\vec{u}_{kl} = \frac{\partial \vec{u}_k(\vec{x})}{\partial t_l} = \frac{\partial}{\partial t_l} \int_{\Gamma} N_k^{SUPG} \vec{n}(\vec{x}) ds = \int_{\Gamma} N_k^{SUPG} \frac{\partial \vec{u}(\vec{x})}{\partial t_l} d\xi d\eta = \sum_{f \in \mathbb{P}_k \cap \mathbb{P}_l} \int_f N_k^{SUPG} \frac{\partial \vec{u}^f(\vec{x})}{\partial t_l} d\xi d\eta \quad (4.5.30)$$

In the above formulation, the local facet contribution of the gradient term $\partial \vec{u}^f(\vec{x}) / \partial t_l$ is described as following:

$$\frac{\partial \vec{u}^f(\vec{x})}{\partial t_l} = \vec{d}_l \times (\vec{x}_{l1} - \vec{x}_{l2}) \quad (4.5.31)$$

In the equation above, $l1$ and $l2$ represent the cyclic indices of the two other nodes of the element with respect to the node l . The contact constraint (4.5.29b) is applied with the penalty method, and used to update the mechanical functional as explained in the equation (4.5.1b). The 2-DoF implicit contact constraint presented in equation (4.5.28) is implemented in the following manner:

$$h(\vec{v}_k) = \vec{v}_k \cdot \left(\hat{u}_k^{[1]}(\vec{x}^{(i)}) \left(\vec{d}_l^{[1]} \cdot \vec{n}_l^{tool} \right) + \hat{u}_k^{[2]}(\vec{x}^{(i)}) \left(\vec{d}_l^{[2]} \cdot \vec{n}_l^{tool} \right) \frac{-\vec{v}_k^{(i)} \cdot D\vec{u}_{kl}^{[1]}}{-\vec{v}_k^{(i)} \cdot D\vec{u}_{kl}^{[2]}} \right) - \delta_l \left(-\vec{v}_k^{(i)} \cdot D\hat{u}_{kl}^{[1]} \right) \leq 0 \quad (4.5.32)$$

The advantage of this form is that it automatically becomes the 1-DoF form for the surface nodes. It is to be remarked here that the implicit (RHS) term is dependent on the velocity which is considered to be the value converged from the mechanical solution of the last fixed-point iteration. *For the parallel order implementation of the implicit contact constraint, the individual explicit and implicit parts of the upwind normal are computed at the individual processor and each processor sends its contribution to the root. The explicit part is computed exactly in the same manner as explained in the Section 4.5.1. The computation of implicit part is tricky as the node l , which is an upwind node to node k may be located in a different mesh partition. This is the scenario when for the node k on the interface of two mesh partitions. A global loop on all element facets is made in each mesh partition, with a buffer that stores the directions \vec{d} , the tool normal \vec{n}^{tool} and the gap function δ at node l to each node k . In addition, a table also stores the cosine of the angular position of node l wrt to node k . During the communication call, the minimization of cosine (at the root) locates the rightful node l for each node k and the corresponding directions, tool normal and gap function are communicated back to the originating processor. Once these values are known at each node, it becomes fairly simple to compute the implicit upwind normal at each node k with the equation (4.5.32).*

Updated hessian in mechanical description

The hessian matrix for the implicit contact constraint is described in the same manner as with the explicit constraint, but with the changed description of the constraint $h(\vec{v}_k)$.

$$\forall k \in \Gamma_c^{(2)} \quad H_{k,i,k,j}(\vec{v}) = \rho_c^{(1)} s_k \begin{cases} \tilde{u}_{k,i} \otimes \tilde{u}_{k,j} & \text{if } h(\vec{v}_k) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4.5.33)$$

4.5.3 Updated steady-state algorithm

The main objective of the current section is to improve the robustness of the existing ForgeNxt[®]2016 algorithm. A new contact formulation, in explicit and implicit form, has been discussed in the previous sections. This formulation is consistent and would hopefully help eradicate the problem of solution instability in the existing algorithm. The implicit formulation aims to improve the behavior of the nodes of the workpiece on the first line of contact with the tool. A fully consistent nodal formulation with the same normal and same weak form for the free-surface computation and the contact constraint is ideal for improving the consistency. But as concluded in Section 4.4.5, the nodal form of the free-surface equation is not the best for free surface computation as it is less accurate. Hence the chosen algorithm adopts the new contact formulation while retaining the original integral formulation of the free-surface computation, hence not affecting the loss of accuracy in shape prediction. Hence, the updated algorithm continues to use the integral form of the free-surface equation (4.1.1) but the new contact constraint in the equation (4.5.1a) for velocity computation. From here the two algorithms with explicit and implicit contact formulations are identified as *Steady-State Explicit* and *Steady-State Implicit* respectively.

4.6 Performance with hot-rolling test-cases

The updated algorithms *Steady-State Explicit* and *Steady-State Implicit* are tested with a number of hot rolling test-cases, including the ones presented in Chapter 3. It is of the primary concern to improve the problem related to instability observed with the ForgeNxt[®]2016 formulation, especially with the test-case *VP-Test-Case-02* presented in Section 3.5. The exact same models are used with the same mesh and fictitious material properties presented in Table 3.5.1. **We use the adiabatic conditions for the test-cases VP-Test-Case-01 and VP-Test-Case-02, as done in the previous chapter**, and hence try to focus only on the improvement in the contact problem in the Mechanical step without any thermal computation and coupling. It is to be highlighted here that the temperature itself is a convective field and has associated problems of instability because of the unstable velocity field and contact. It is thus important to isolate the contact problems from this coupling with the adiabatic conditions. However, as an assessment of the new stable formulation, we add another test-case to closely simulate a process with industrial hot rolling conditions. Hence, more realistic material properties with temperature and strain dependency (viscoplastic hardening) are considered with the coupling of the mechanical problem with the thermal one. The convergence of geometry, temperature, equivalent strain, and force for the steady-state algorithm, presented in equation (3.4.6), are studied to compare the results from different steady-state formulations tested. The flux loss in the equation (3.5.1) is also compared with each formulation. The results are validated with the ones from the ForgeNxt[®]inc.

4.6.1 Flat rolling problem: VP-Test-Case-01

The *VP-Test-Case-01* presented in the Section 3.5.1 is tested again with the consistent contact formulation. Since the geometry includes an edge, the 2-DoF contact explicit and implicit formulations presented are used. Even though the ForgeNxt[®]2016 was able to find a stable solution for this simple test-case, the same is tested with the new formulation, to establish that the stability and the quality of the solution are retained with the updated formulation. The global results

for the [VP-Test-Case-01](#) are presented in Table 4.6.1. We see that both the [Steady-State Explicit](#) and [Steady-State Implicit](#) formulations result in converged solution which is very good agreement with the [ForgeNxt[®]inc](#) and the existing [ForgeNxt[®]2016](#) algorithm results. In general, we see a minor under-prediction of force and torque with the steady-state formulations in comparison to the [ForgeNxt[®]inc](#). This difference might come due to the difference in the equivalent strain calculation with the incremental formulations. There is no change observed in the number of fixed-point iterations for convergence with the new contact formulations. However, a minor increase in computation time is observed with [Steady-State Explicit](#), yet it is 50 times faster than the incremental solution. We see a small gain in computation time with the [Steady-State Implicit](#). It is to be brought to the notice of the reader that the computation of the upwind normal \hat{u}_k for the new contact formulation is computationally inexpensive (in terms of time) in comparison to the two resolution steps.

Contact formulation	Shape Correction method	Convergence incs/iters	Comp. Time (min)	Tool Force (Tonnes)	Tool Torque (kN-m)
ForgeNxt[®]2016	SUPG-dif ($\alpha = 1/2$)	5	5.1	17.2	4.13
Steady-State Explicit	SUPG-dif ($\alpha = 1/3$)	5	6.4	17.3	4.12
Steady-State Implicit	SUPG-dif ($\alpha = 1/3$)	5	4	17.2	4.13
ForgeNxt[®]inc (ref)		541	300	17.6	4.26

Table 4.6.1: Global performance of the [VP-Test-Case-01](#) with the updated steady-state algorithm in comparison to the [ForgeNxt[®]inc](#) solution (on 12 cores)

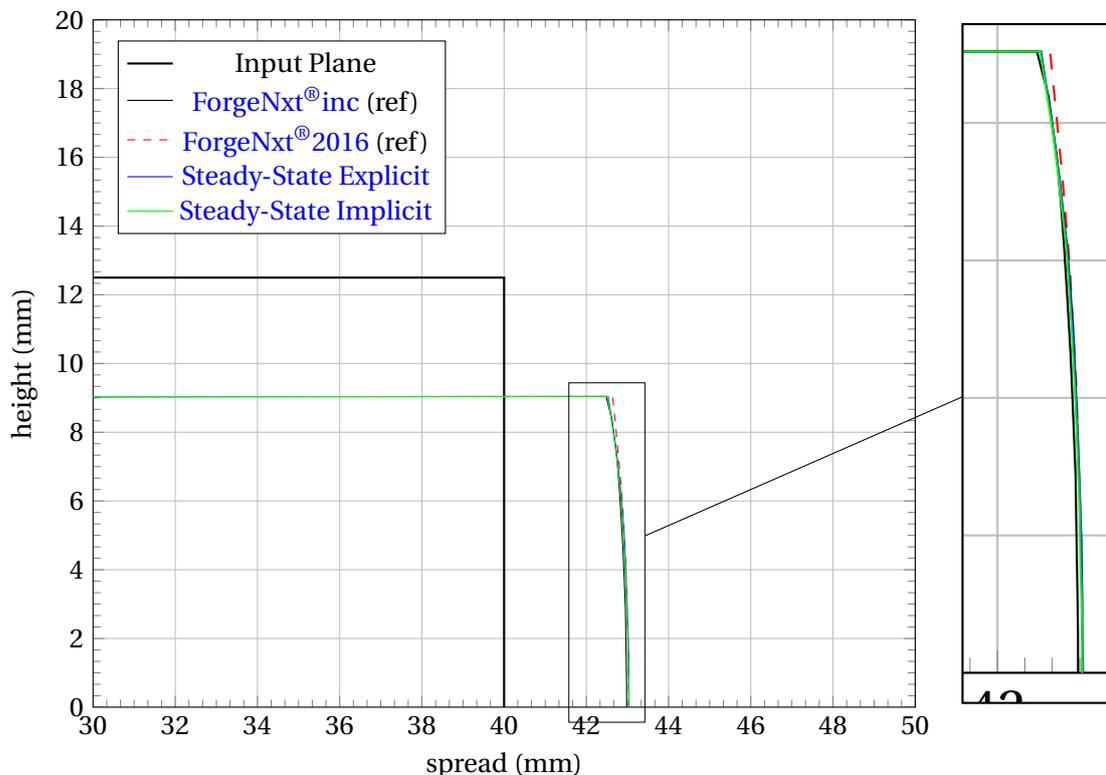


Figure 4.6.1: Comparison of deformed section (steady-state) shape of the [VP-Test-Case-01](#) with the steady-state and incremental formulations

The steady-state shapes computed with the different formulations are compared in Figure 4.6.1. The location of the measurement of shape/section is the same as explained in Section 3.5.1. In the previous chapter, we found out that the [ForgeNxt[®]2016](#) over-predicts the lateral spread with lower prediction of tool force which was assumed to be due to the computed lateral velocity field.

Here, we can observe that the steady-state algorithm with the consistent contact condition (both explicit and implicit formulations), is able to improve the prediction for the lateral spread at the top face in comparison to the ForgeNxt[®]2016 solution. This means that the new contact formulation is really able to improve the velocity computation at the contact interface in comparison to the existing formulation.

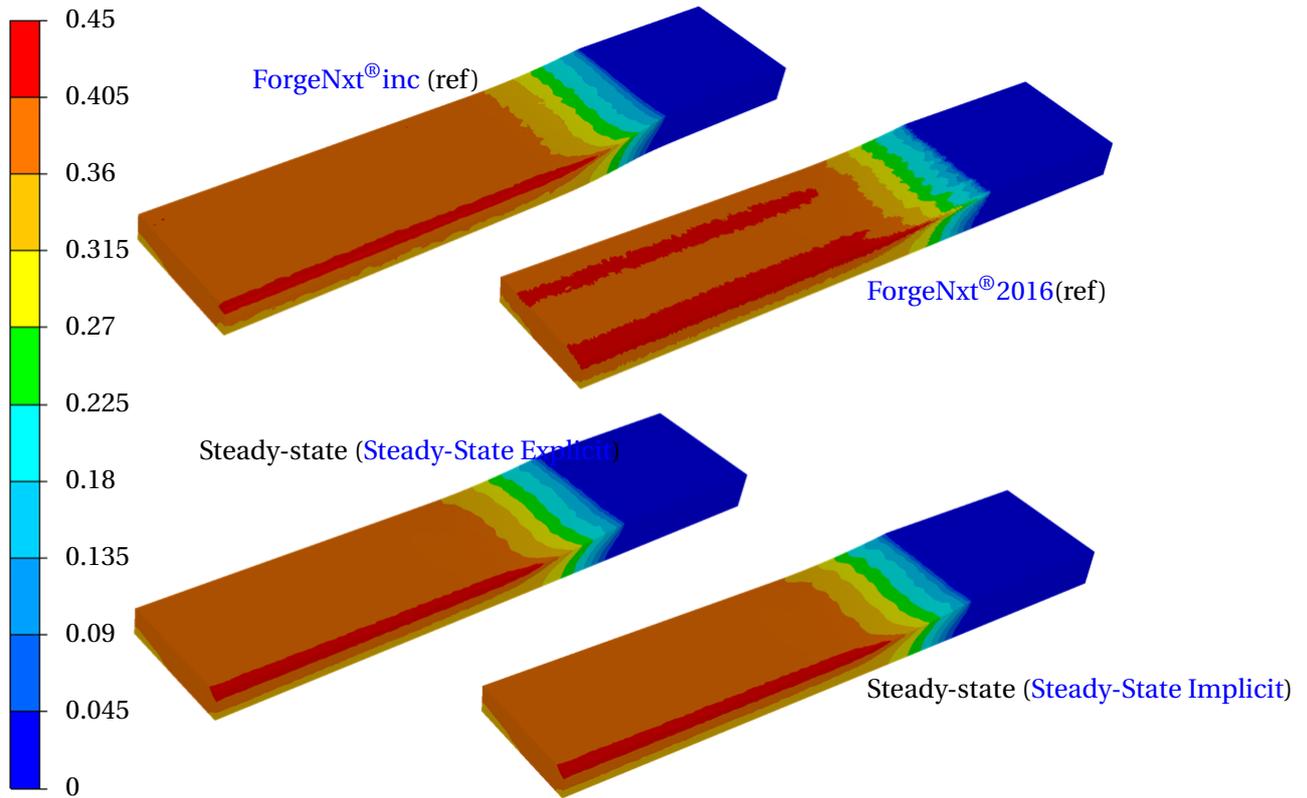


Figure 4.6.2: Comparison of equivalent strain for VP-Test-Case-01

Next, we present the solution contour for the equivalent strain in Figure 4.6.3. We see that the equivalent strain from both the consistent contact formulations is in good agreement with the incremental results, and improves the prediction in comparison to the ForgeNxt[®]2016 results, as the wider spread-out of the peak strains and oscillations below the tool are no more visible with the new formulations. The $\bar{\epsilon}$ field looks much more stable in comparison to the existing formulation. This could well be due to a more stable contact surface in comparison to the previous algorithm. The von Mises stress solution from the different algorithms are compared in Figure 4.6.4. The first observation is that unlike equivalent strain, the von Mises stress computed with the steady-state formulation is more stable as it is not a convective field. At the input plane Γ_{in} we see non-zero stress in all the steady-state solutions. This can be avoided by choosing a larger length of the workpiece before the contact. However, this is neither critical to the convergence of the algorithm nor the solution is sensitive to this with a hot rolling case (with VP material). This must be surely be concerning with EVP material where the stress is also a convective field and residual stresses need to be transported from the input plane.

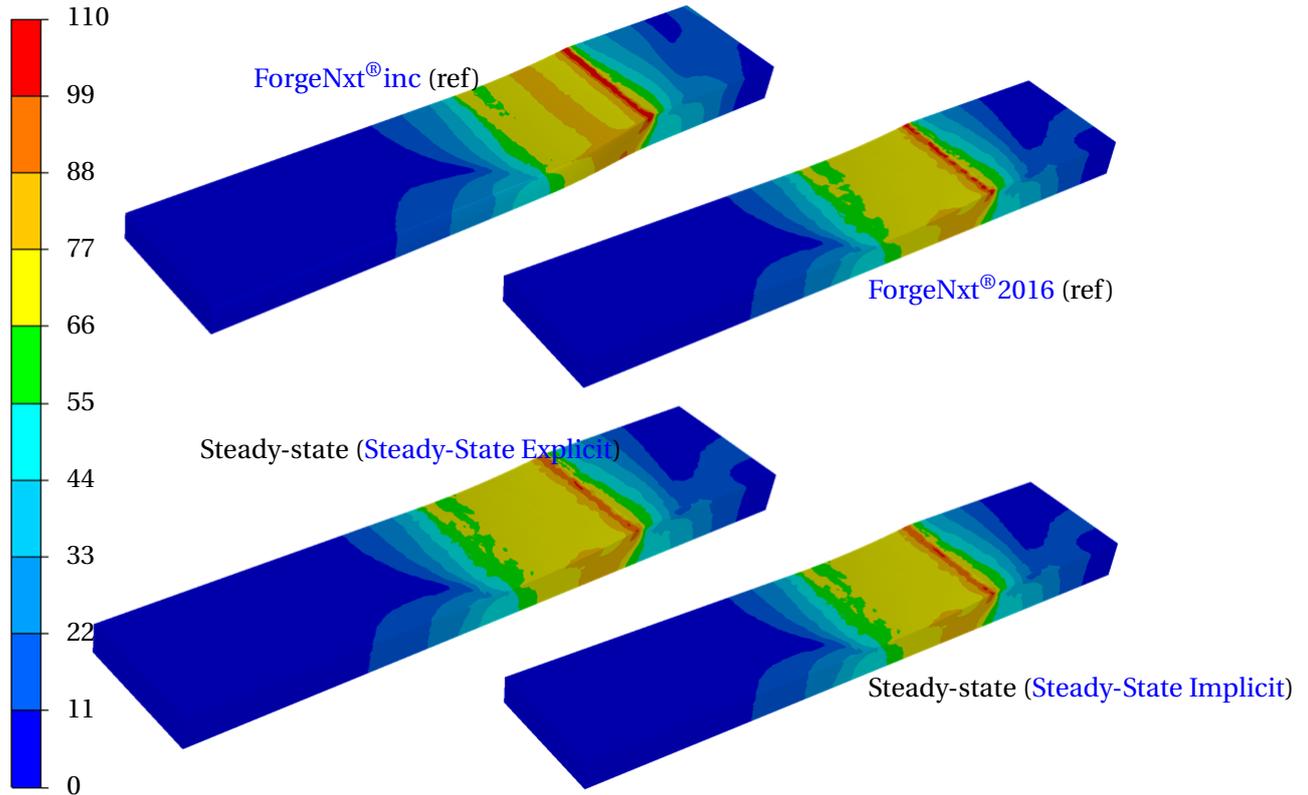


Figure 4.6.3: Comparison of von Mises stress for VP-Test-Case-01

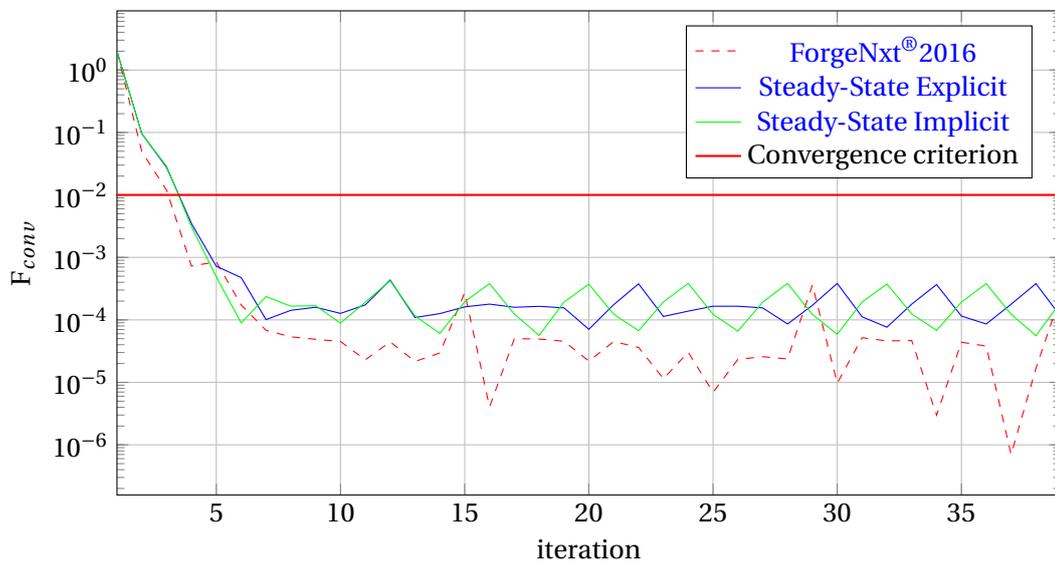


Figure 4.6.4: Comparison of Tool force convergence for VP-Test-Case-01 with different steady-state formulations

The solution stability is studied with the four different criteria presented in the equation (3.4.6) with a reduced geometric convergence criterion $\epsilon_{geom} = 10^{-5}$ to compare the different steady-state formulations. The Force convergence and geometry convergence results are presented in Figure 4.6.4 and Figure 4.6.5 respectively. It can be observed that the force and geometry solutions converge with all the steady-state formulations and remain stable below their respective convergence criterion. The Force residual, signifying the difference of tool force between two consecutive iterations, is almost the same order with the three formulations, however, ForgeNxt® 2016 formula-

tion results in a slightly better solution because of the low magnitude of oscillation. The geometry convergence signifies the maximum difference of correction between two consecutive iterations, measured at the output plane Γ_{out} . It is found to be below the (regular and not the reduced one chosen for stability test) defined criterion.

The convergence of equivalent strain plotted in Figure 4.6.6 signifies the maximum difference of the equivalent strain in an element between two iterations. It can be observed that all the three methods result in convergence of the equivalent strain criterion, with ForgeNxt® 2016 resulting in the least residual for this case. The flux loss with the three formulations is compared in Figure 4.6.7. It represents the difference of material flow entering and leaving the control volume. We observed that with all the formulations we have near-zero flux loss, but ForgeNxt® 2016 results in the minimum loss. Nevertheless, the loss with the updated formulations results in a minor loss ($< 0.25\%$) of the material.

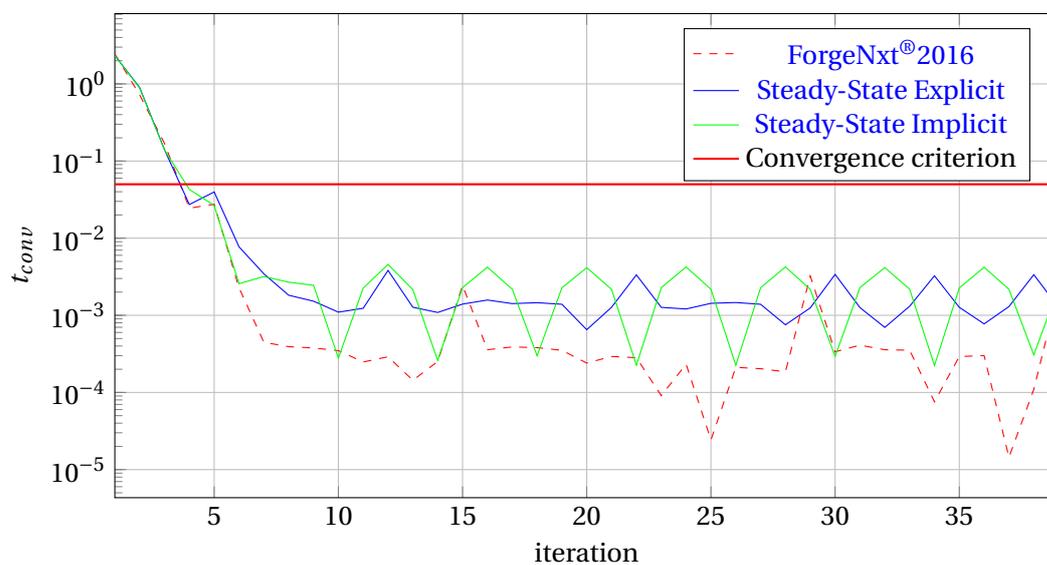


Figure 4.6.5: Comparison of geometry convergence for VP-Test-Case-01 with different steady-state formulations

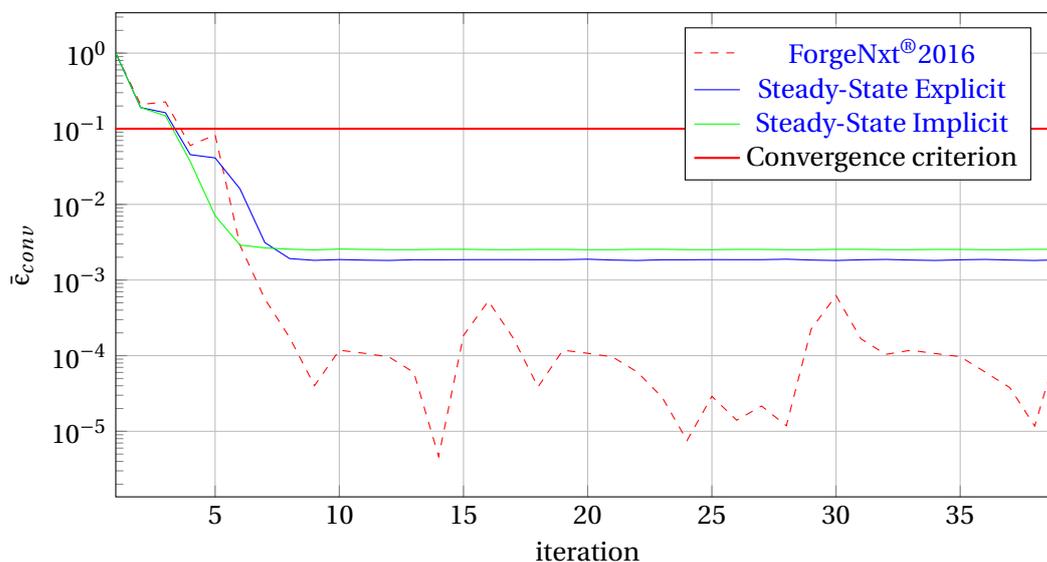


Figure 4.6.6: Comparison of equivalent strain convergence for VP-Test-Case-01 with different steady-state formulations

Conclusions from solution

From the observed results, we can conclude that the new proposed contact formulations do not degrade the solution obtained with the ForgeNxt[®]2016. The tool force and torque predicted have a good agreement and the shape prediction is improved with the updated steady-state algorithm. The stress and strain obtained with the new formulation have an excellent agreement with the ForgeNxt[®]inc and the instabilities in the solution observed with the ForgeNxt[®]2016 solution are suppressed to a good extent. It was also observed that the convergence parameters for all the presented steady-state formulations are stable and well under the convergence criteria chosen for the algorithm. However, with the observed results of the VP-Test-Case-01, it cannot be concluded on which of the two new contact formulations is better.

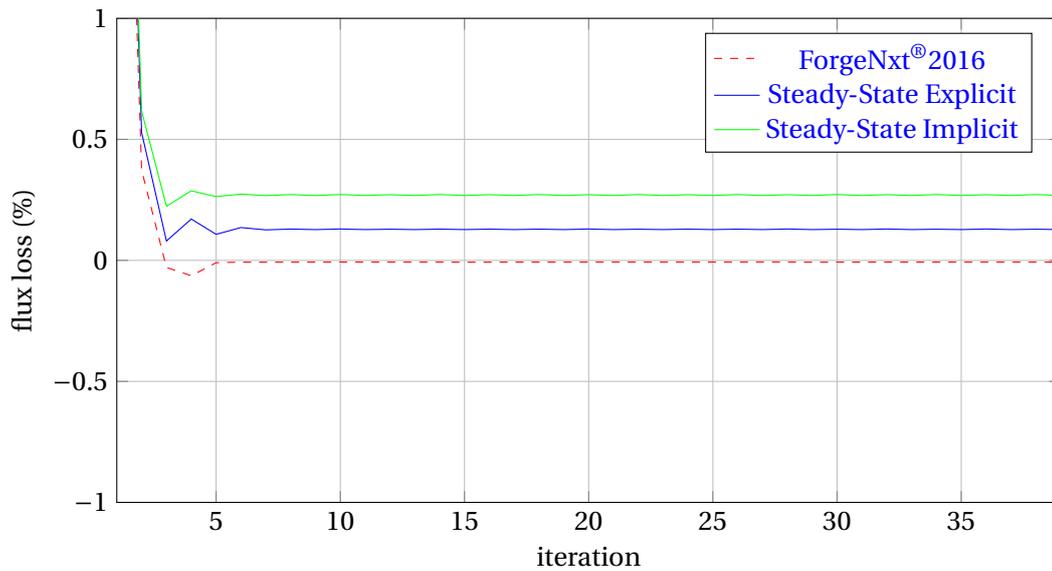


Figure 4.6.7: Comparison of flux loss for VP-Test-Case-01 with different steady-state formulations

4.6.2 Shape rolling problem: VP-Test-Case-02

Next the shape rolling of VP-Test-Case-02 is discussed. As seen in the previous Chapter 3.5.2, this test-case did not converge to a unique solution but meandered around a solution which was quite close to the one predicted with the incremental formulation. For this case, the (equivalent strain) convergence criteria of the ForgeNxt[®]2016 algorithm was found not to converge. The problem is tested with the updated steady-state formulation with both Explicit and Implicit contact conditions and is validated with the ForgeNxt[®]inc solution. Same fictitious material properties and mesh model are chosen as presented in Table 3.5.1 and Figure 3.5.7 respectively.

Contact formulation	Shape Correction method	Conv. Incs/Iters	Comp. Time (min)	Tool Force	Tool Torque
				(Tonnes)	(kN-m)
ForgeNxt [®] 2016	SUPG-dif ($\alpha = 1/2$)	×	×	15.94 ± 0.02	9.42 ± 0.04
Steady-State Explicit	SUPG-dif ($\alpha = 1/3$)	12	8.25	16.06	9.36
Steady-State Implicit	SUPG-dif ($\alpha = 1/3$)	9	8.32	15.98	9.37
ForgeNxt [®] inc (ref)			132	15.49	9.20

Table 4.6.2: Global performance of the VP-Test-Case-02 with the updated steady-state algorithm in comparison to the incremental solution (on 12 cores)

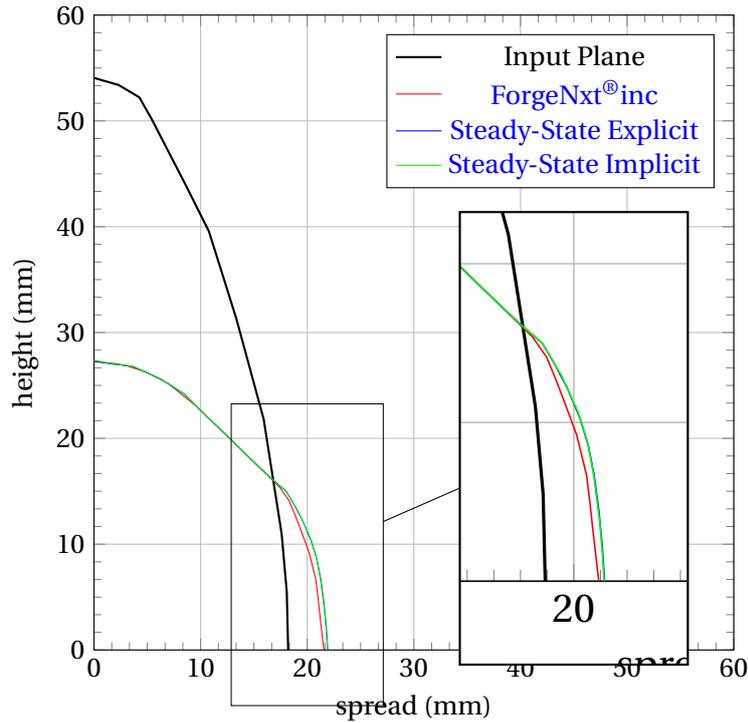
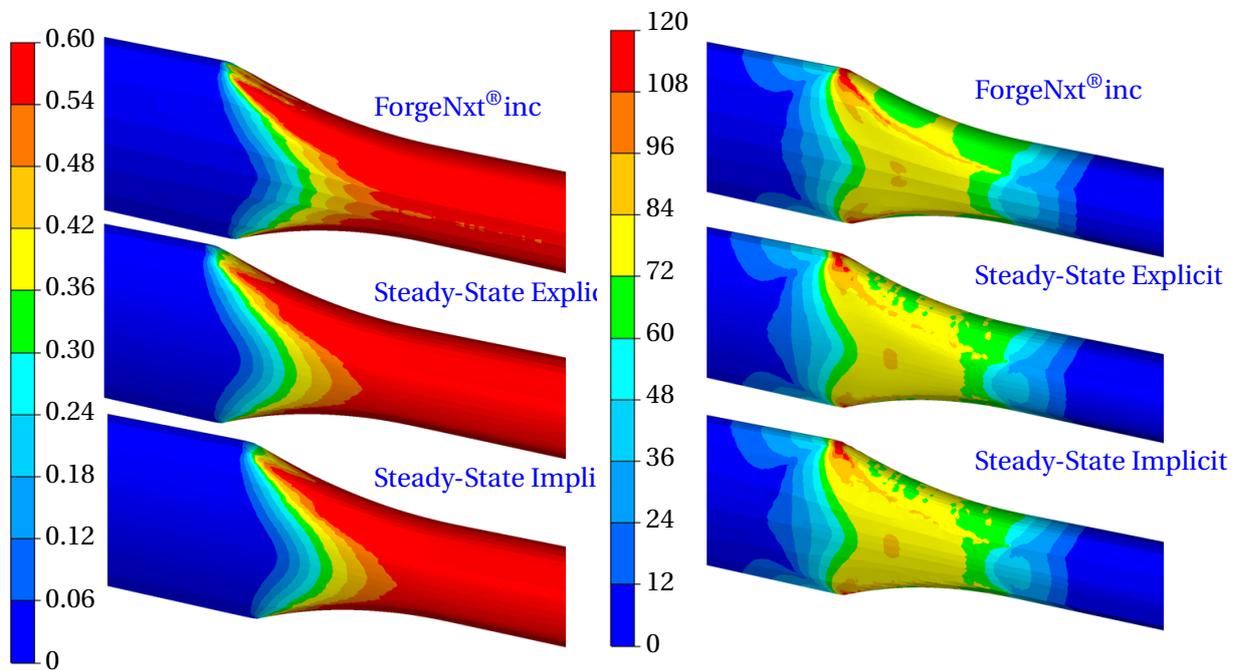


Figure 4.6.8: Comparison of deformed section (steady-state) shape of the **VP-Test-Case-02** with the steady-state and incremental formulations



(a) Comparison of equivalent strain for **VP-Test-Case-02**

(b) Comparison of von Mises stress (in MPa) for **VP-Test-Case-02**

Figure 4.6.9: Comparison of **VP-Test-Case-02** steady-state solution from different formulations with the incremental solution

The **VP-Test-Case-02** global results and shape comparison are presented in Table 4.6.2 and Figure 4.6.8 respectively. The shape computed with **ForgeNxt®2016** is not represented as the algorithm computes an oscillatory solution. It is seen that the algorithms **Steady-State Explicit** and

Steady-State Implicit result in converged solution in 12 and 9 iterations respectively. The time taken for both the steady-state methods is similar and a speedup of ≈ 16 times achieved in comparison with the **ForgeNxt[®]inc** resolution. The tool force is over-predicted by 3% and the torque by 2%, in comparison to their under-prediction in **VP-Test-Case-01**. It is seen that the steady-state shape computed from the **Steady-State Explicit** and **Steady-State Implicit** formulations have a good agreement with the incremental solution but the lateral spread is over-predicted. This corresponds to 3.9 mm with steady-state resolution in comparison to 3.56 mm with **ForgeNxt[®]inc**, which is nearly 10% more. This observation is in-line with the over-prediction of the tool force with the steady-state algorithm which leads to larger material side flow, and hence more lateral spread. It is to be pointed out that the steady-state resolution involves surface mesh regularization, hence, we may not have the exact correspondence of nodes on the lateral surface between the steady-state and incremental meshes. In addition, the incremental solution is the reference here, but not is not an absolute solution as it involves e.g. time-stepping which may cumulate numerical errors.

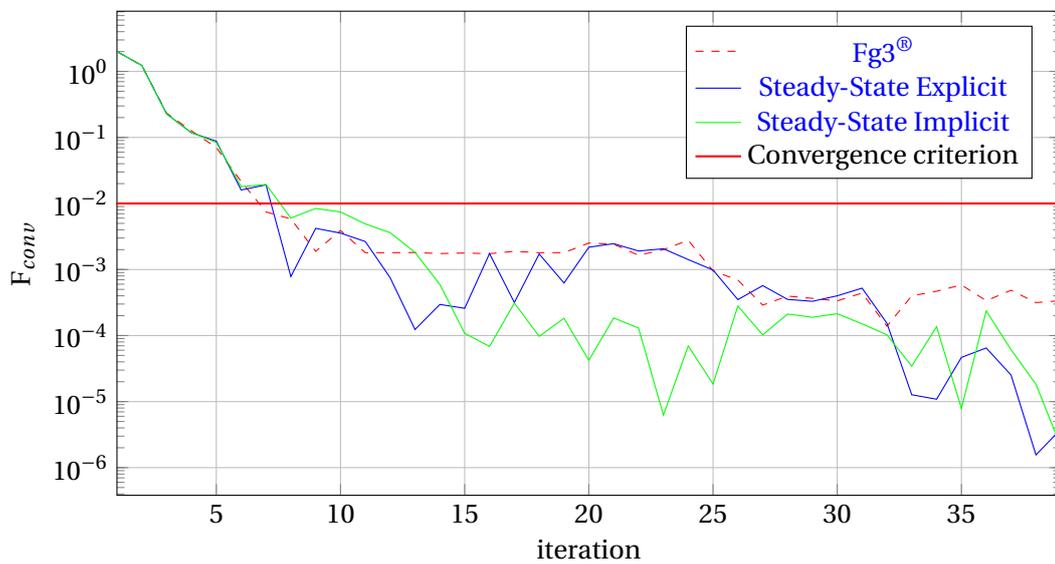


Figure 4.6.10: Comparison of Tool force convergence for **VP-Test-Case-02** with different steady-state formulations

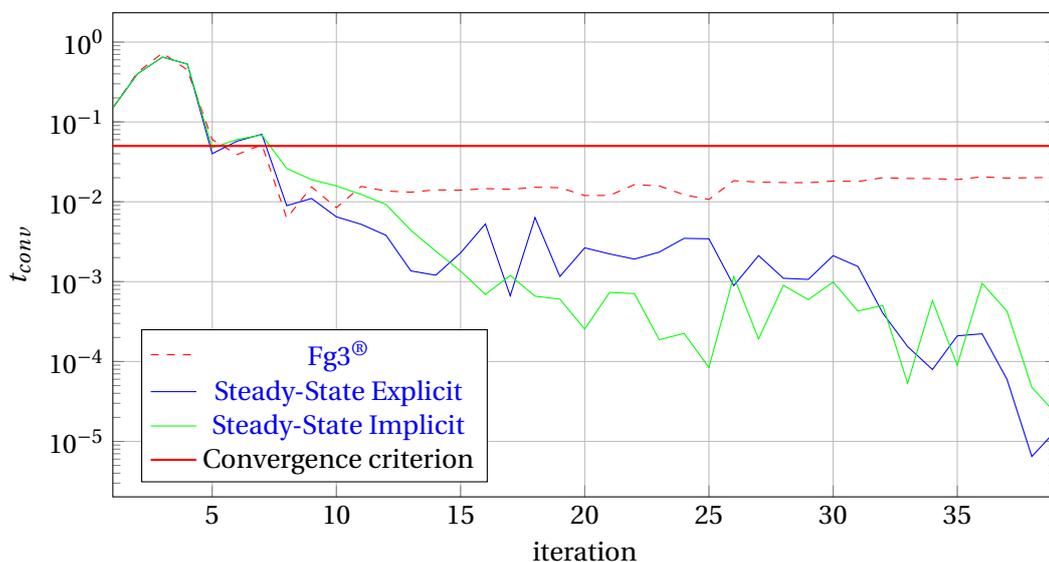


Figure 4.6.11: Comparison of geometry convergence for **VP-Test-Case-02** with different steady-state formulations

The equivalent strain and von Mises stresses with the incremental and the steady-state formulations are compared in Figure 4.6.9a and Figure 4.6.9b respectively. We observe a stable equivalent strain with the steady-state formulations in comparison to the peaks observed with the incremental solution. In general, we see a very good agreement of both these fields computed with the consistent steady-state formulation in comparison to the incremental solution. Between the explicit and implicit contact formulations, there are very minor differences in the results. The von Mises stress from the steady-state formulation also has a very good agreement with the incremental solution. There are some oscillations seen in the middle of the contact with the tool, but this appears to come from the discretization.

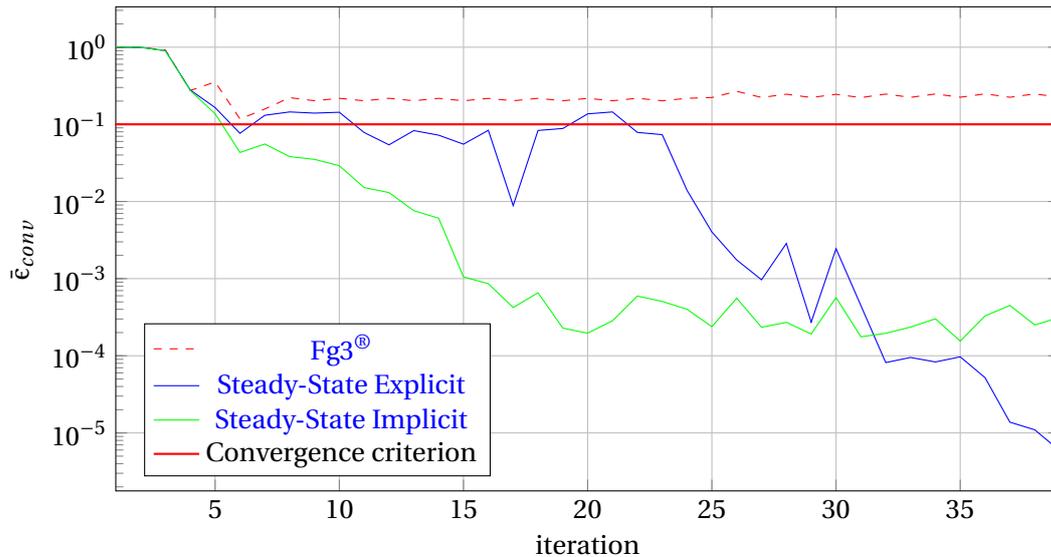


Figure 4.6.12: Comparison of equivalent strain convergence for [VP-Test-Case-02](#) with different steady-state formulations

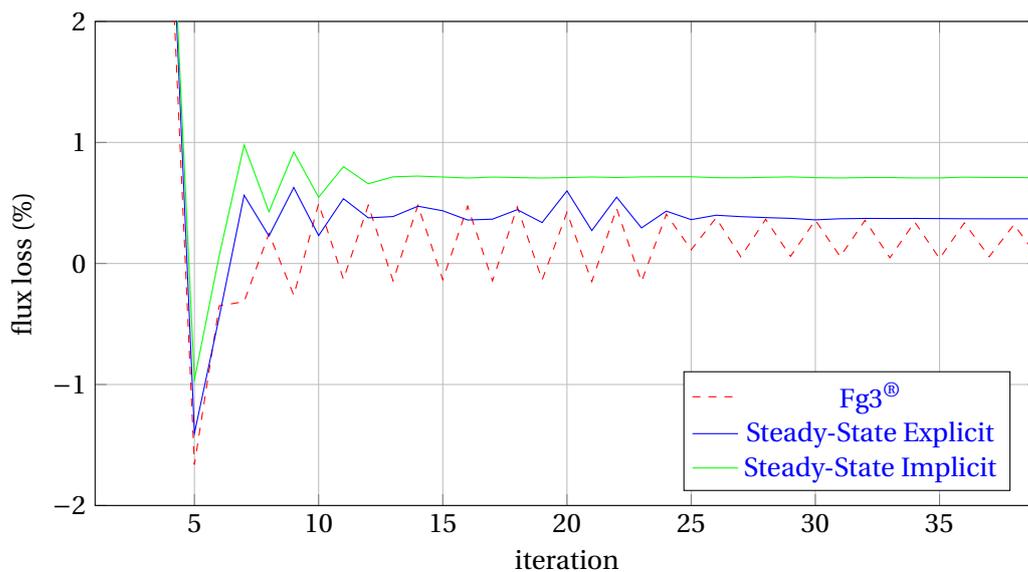


Figure 4.6.13: Comparison of flux loss for [VP-Test-Case-02](#) with different steady-state formulations

It would be interesting to see the stability of the converged results obtained from [Steady-State Explicit](#) and [Steady-State Implicit](#) by reducing the geometric convergence criterion. This is important to ensure the claim that the modified formulation is better in terms of the stability of the solution. The Force convergence with the three steady-state formulations presented is plotted in

Figure 4.6.10. We see that the Force residual converges with all the three formulations after 7 iterations. The minimum tool force residual is seen with the implicit formulation. The geometry convergence criterion (in Figure 4.6.11) is seen to converge in 4 iterations with all the formulations. The residual with the new consistent contact formulations is much lower in comparison to the ForgeNxt®2016 formulation. The evolution of the critical equivalent strain convergence is presented in Figure 4.6.12. We see that in comparison to the ForgeNxt®2016, the equivalent strain converges with both the consistent formulations. The behavior with explicit contact formulation is however not seen to be as stable as the implicit formulation. The flux loss function with the different steady-state formulations is presented in Figure 4.6.13. We observe that the flux loss with the ForgeNxt®2016 formulation is seen to undulate, which suggests larger oscillations in the velocity and shape fields. The consistent formulations have constant positive flux loss, but the magnitude is larger in comparison to ForgeNxt®2016, but from the industrial viewpoint, this value is negligible.

Conclusion from solution

It is found that as the complexity of the test-case is increased, the ForgeNxt®2016 formulation fails to find a stabilized solution for the rolling problem. In such a scenario, the modified steady-state algorithm with consistent contact formulation proves to be a better choice as we find a unique and stable solution to the problem. It is also to be mentioned that the implicit contact formulation presented in this chapter results in a better solution in terms of the number of increments for convergence and the solution stability. However, the results obtained for VP-Test-Case-02 behave in a different manner than the previous test-case. This may be attributed to the inherent differences in the two test-cases. Firstly, VP-Test-Case-01 had singular geometry and hence the 2-DoF formulation is active. Secondly, the tool contact in VP-Test-Case-01 was flat and the initial contact is almost a straight line whereas in VP-Test-Case-02, the initial contact load is concentrated on a few nodes due to the elliptical contact. So, the elements at the initial contact have large curvatures and most of the tool force is transferred through these nodes. So, in comparison to the VP-Test-Case-01, this test-case has a larger discrepancy.

4.6.3 Shape rolling problem: VP-Test-Case-03

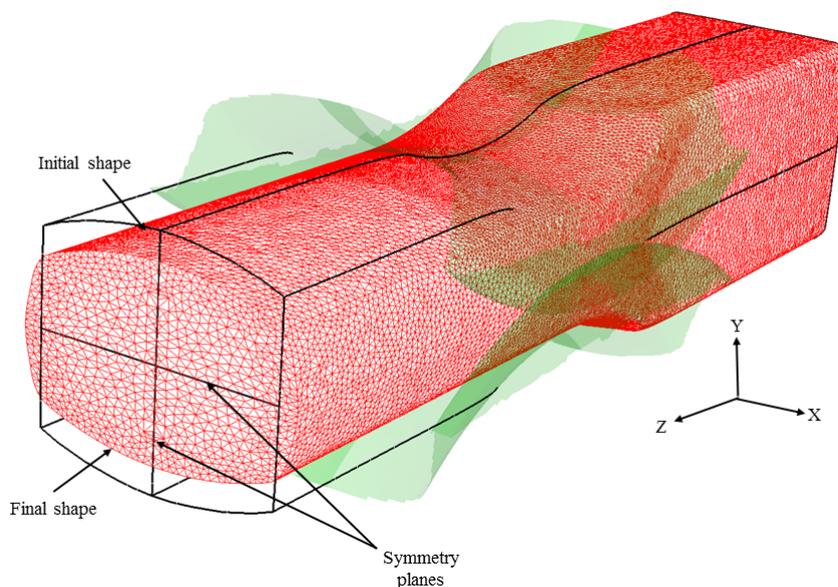


Figure 4.6.14: Test-Case 03: Square to Rectangle Shape rolling

In this section, we would like to present another test-case for further testing the capabilities of the updated steady-state algorithm in comparison to the existing [ForgeNxt[®]2016](#). In this test-case, a square section long bar of dimension 205mm x 205mm with filleted corners is rolled between two rigid rolls of diameter 290mm rotating at 15.3 rpm. Similar to [VP-Test-Case-02](#), the workpiece in this case has a non-uniform contact with the roll. The rolls impose a side constraint on the workpiece and this restricts the free flow of the material in the lateral direction. In addition, to further complicate the problem for testing the robustness of the algorithm, the following changes are made to the model in comparison to the previous test-cases: The end product is a rectangular section of dimension 147.56mm x 232.22mm with curved free-surfaces in the steady-state (see [Figure 4.6.14](#)). This test-case is proposed by the Transvalor S.A. team.

S.No.	Parameter	Value	
1	Norton Hoff (strain rate dependence) $\dot{\epsilon}$	0.01-20 (10 values) s^{-1}	
2	Norton Hoff (Temperature dependence)	900K-1400K (11 values)	
3	Density ρ	7360 kg/m^3	
4	Specific heat	677.2 JK^{-1}	
5	Coulomb friction \bar{m}	0.8	
6	Coulomb friction μ	0.4	
Heat transfer coefficients			
S.No.	Parameter	Adiabatic	Fully Coupled
7	Thermal Conductivity K_{cond}	0.0	29.26
8	Emissivity ϵ_{rad}	0.0	0.85
9	Die heat transfer coefficient	0.0	2000
10	Die Effusivity	0.0	11763.62
11	Ambient heat transfer coefficient	0.0	10

Table 4.6.3: Properties used for the [VP-Test-Case-03](#)

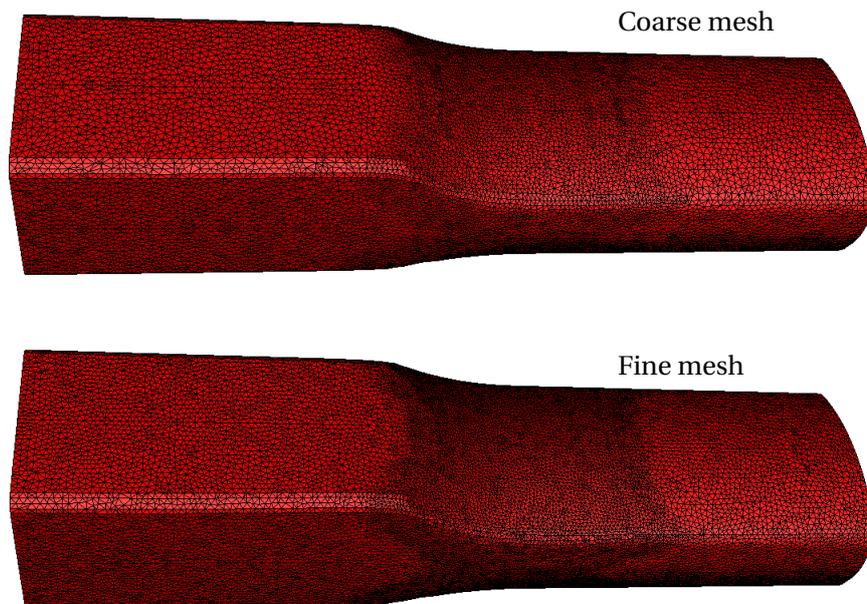


Figure 4.6.15: Coarse and fine mesh models tested for [VP-Test-Case-03](#)

There are some more differences in comparison to the previous two test-cases, listed as following:

1. **Initialization** with forging method, as shown in [Figure 3.2.2c](#)

2. **Friction** with Coulomb model, instead of viscoplastic model, for the contact between the tool and the workpiece
3. **Material behavior** rate of deformation/Temperature dependent material is used as shown in Figure 4.6.16
4. **Mesh convergence study** of steady-state solution with two different mesh sizes: (i) **coarse** contact surface element size : 6 mm, free-surface element size 8 mm and bulk element size = 10 mm- 87139 elements (19142 nodes) and (ii) **fine** contact surface element size : 4 mm, free-surface element size 6 mm and bulk element size = 8 mm- 125015 elements (27249 nodes) as shown in Figure 4.6.15.

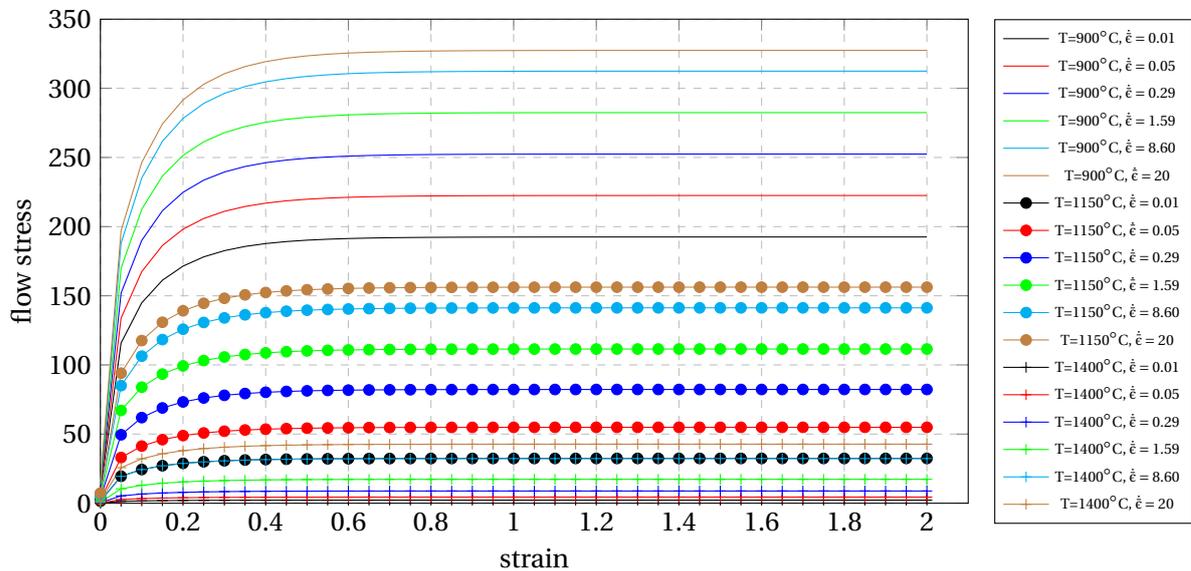


Figure 4.6.16: Rate-temperature dependent stress-strain curve used for resolution of VP-Test-Case-03

The material, friction, and thermal properties used for the simulation are tabulated in Table 4.6.3. The temperature and rate of deformation dependent properties between 900°C-1400°C and 0.01-20 s^{-1} respectively, are plotted in Figure 4.6.16. The material strength is seen to increase at higher rate of deformations and lower temperatures. The work-piece is initially heated to 1250°C and cooled in ambient air (50°C) to get the temperature profile shown in Figure 4.6.17. Two different sub-tests are carried out:

- Firstly, an **adiabatic condition** is simulated to test only the mechanical solver isolated from thermal coupling by nullifying the coefficients that permit the various forms of heat transfer
- Next, the **fully coupled** thermal-structural problem is resolved with appropriate heat transfer coefficients (as shown in Table 4.6.3).

Because of the filleted corners in the initial geometry, there is no presence of singularity and hence the 1-DoF formulation is used for both contact and free-surface computation.

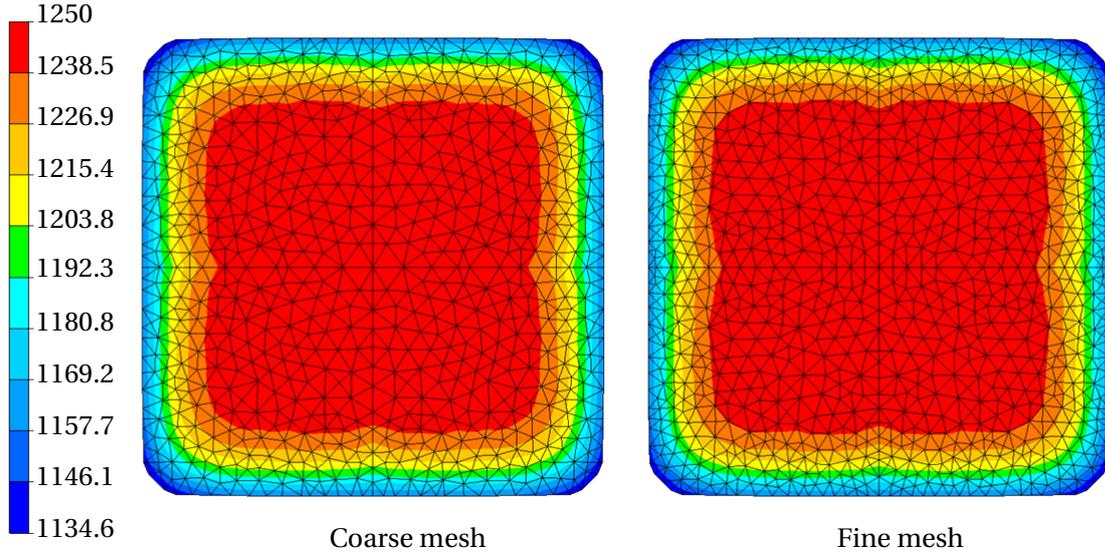


Figure 4.6.17: VP-Test-Case-03: Temperature field (in °C) profile at the input plane Γ_{in} of the VP-Test-Case-03 model

4.6.3.1 VP-Test-Case-03 with Adiabatic conditions

Contact formulation	Shape Corr. method	Mesh	Conv. Incs/Iters	Comp. Time (min)	Tool Force	Tool Torque
					(T)	(kN-m)
ForgeNxt®2016	SUPG-dif ($\alpha = 1/2$)	Coarse	×	×	116.1 ±0.3	92.53 ±1.1
		Fine	×	×	115.38 ±0.33	92.14 ±1.0
Steady-State Explicit	SUPG-dif ($\alpha = 1/3$)	Coarse	×	×	116.25 ±0.15	92.17 ±0.26
		Fine	14	7.55	116.15	92.42
Steady-State Implicit	SUPG-dif ($\alpha = 1/3$)	Coarse	14	6.13	116.54	92.22
		Fine	13	8.87	115.72	91.77
ForgeNxt®inc		Coarse	452	179.5	115.1	91.9

Table 4.6.4: Global performance of the VP-Test-Case-03-adiabatic with the updated steady-state algorithm in comparison to the incremental solution (on 12 cores)

The main objective of the adiabatic test is to compute the structural fields in a workpiece with a temperature gradient but without any changes in the thermal field due to the heat transfer from conduction, convection and radiation effects similar to VP-Test-Case-01 and VP-Test-Case-02. The steady-state results from this test-case are presented in Table 4.6.4. On comparing the steady-state algorithms, we observe that the solution from ForgeNxt®2016 formulation is unstable and does not converge, even with the mesh refinement. The same is observed with the Steady-State Explicit formulation with the coarse mesh but it converges with the fine mesh. However, the formulation with implicit contact condition converges with both coarse and fine meshes and the number of iterations for resolution are similar to both. The number of elements in the finer mesh is 1.5 times in comparison to the coarse mesh, and explains a similar increase in computation time with the fine mesh. In comparison to the incremental solution, which is done with coarse mesh, a speedup of 30 times with coarse mesh and 20 times with fine mesh is observed with the steady-state algorithms.

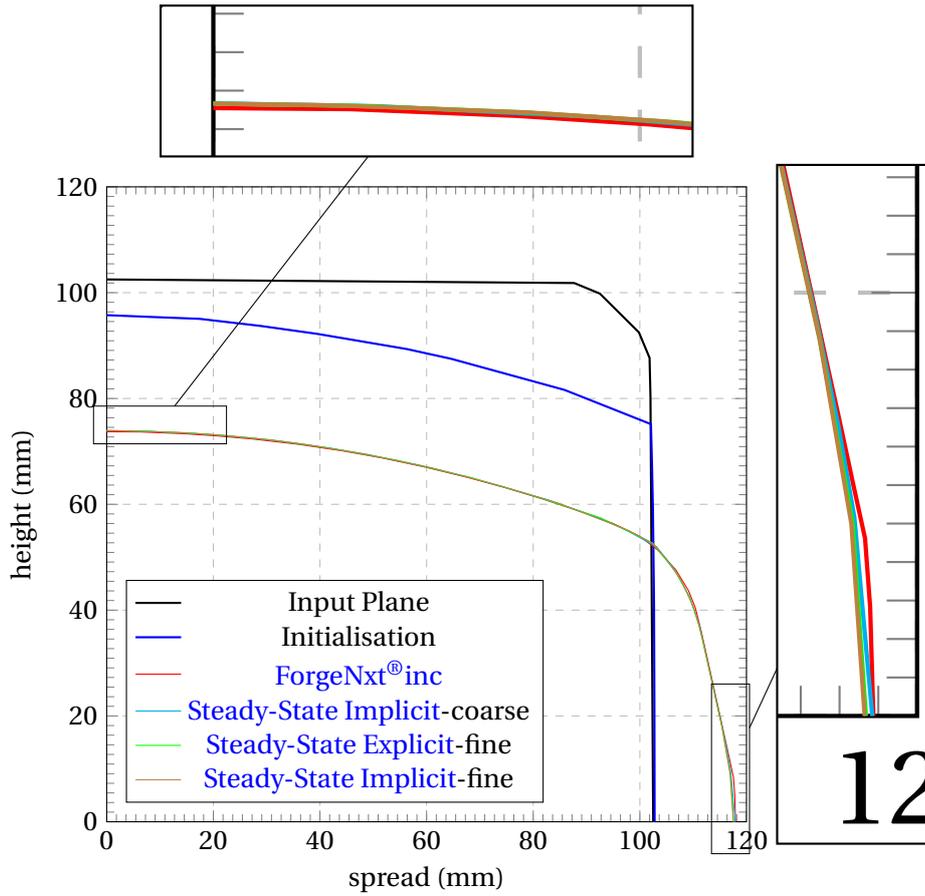


Figure 4.6.18: Comparison of deformed section (steady-state) shape of the VP-Test-Case-03-adiabatic with the steady-state and incremental formulations

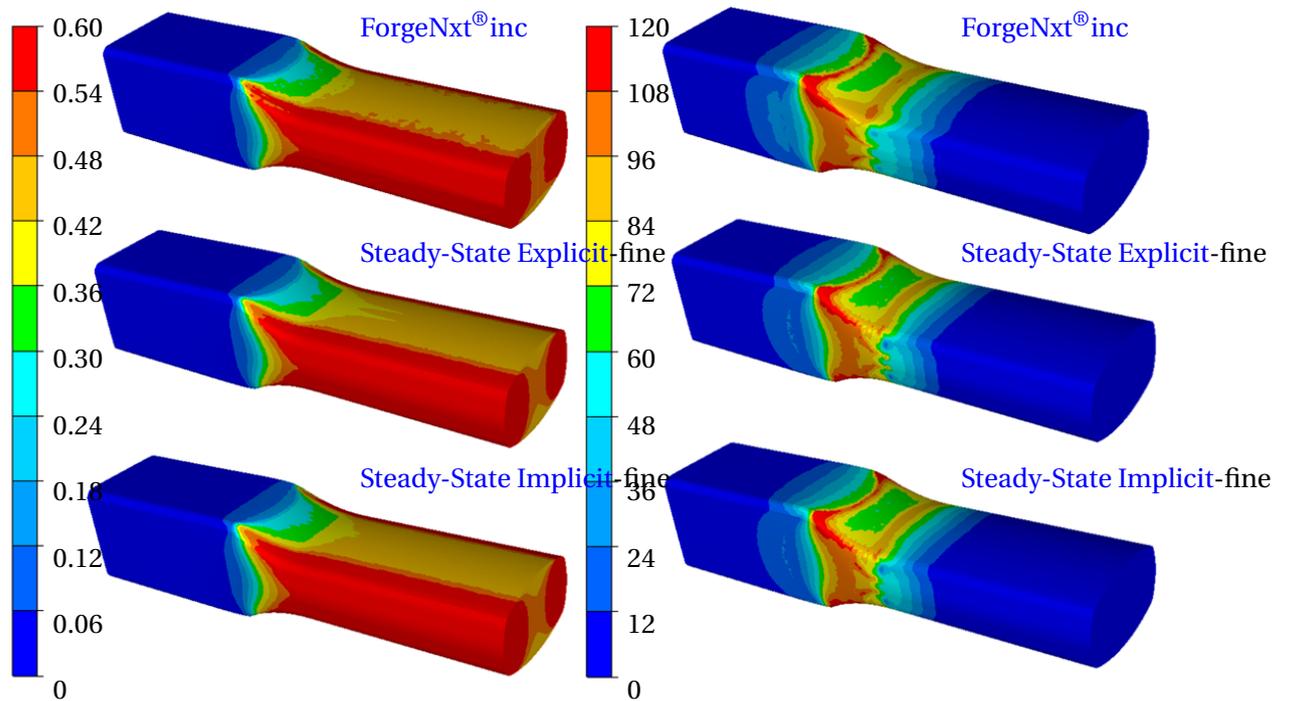


Figure 4.6.19: Comparison of equivalent strain (on left) and von Mises stress (in MPa, on right) for VP-Test-Case-03-adiabatic with different steady-state contact formulations in comparison to the ForgeNxt® inc solution

It can also be seen that the tool force and torque computed with the steady-state formulation are slightly over-predicted (maximum 1.2% and 0.5% respectively) in comparison to the incremental solution. The tool force and torque obtained from the two mesh sizes are quite similar with <1% difference, thus we can conclude that the obtained solution is converged. Given the large complexities introduced into the problem with the thermal-strain hardening, Coulomb's friction model, and nonuniform temperature profile, this difference is quite nominal. The steady-state shapes computed with the different formulations are compared in Figure 4.6.18. We can see an overall very good prediction of the steady-state shape with the steady-state algorithm in comparison to the ForgeNxt® inc solution. Looking carefully, the coarse mesh with implicit formulation has an exact prediction of the lateral spread wrt the ForgeNxt® inc solution. However, a little far away from the symmetry plane, there is a minor non-conformance of the shape, which could be due to the absence of a corresponding node in the incremental resolution mesh. With the fine mesh, both explicit and implicit formulations predict almost the same steady-state shape. The equivalent strain with the steady-state formulations are compared in Figure 4.6.19 (left) respectively. A very little difference was observed between the results from coarse and fine meshes with the steady-state algorithm, hence, we present the results obtained with only fine mesh. The prediction with both explicit and implicit formulations is in very good agreement with the incremental solution. The ForgeNxt® inc solution is seen to have higher strain peaks on the top free-surface, which disappears with the updated steady-state algorithm. With implicit contact formulation, the peaks obtained with explicit formulation (in yellow) are also suppressed. The von Mises stress has an excellent prediction with both the steady-state formulations and is seen to be more stable in comparison to the ForgeNxt® inc. The higher stress concentrations on the side face are well predicted with Steady-State Implicit which is not the case with Steady-State Explicit.

Conclusions from solution

From these results, we can affirm that the robustness of the steady-state algorithm is improved with the proposed modification of the contact formulation. The results obtained with the implicit formulation are seen to be much better in comparison, as it ensures a stabilized solution with different mesh sizes.

4.6.3.2 VP-Test-Case-03 with fully coupled thermo-structural resolution

This test is carried out to assess the performance of the steady-state mechanical solver when coupled with the thermal solver. Hence, the heat transfer coefficients tabulated in Table 4.6.3 for the fully coupled resolution are used. So, during the course of resolution of the fully coupled problem, the heat is transferred between the surroundings & the workpiece by convection and radiation and between the die & the workpiece by conduction. In addition, the internal heat generated due to the deformation is transmitted by conduction. The same mesh, initialized geometry, contact, and mechanical boundary conditions are used as with the adiabatic simulations. The validation is done with the incremental resolution of the same test-case.

The ForgeNxt® 2016 results in an unstable solution as seen with the adiabatic case. The simulation with the consistent contact formulations (both explicit and implicit) converges towards a unique solution with both coarse and fine meshes. The global solution is seen to have <1% change with the mesh refinement and hence considered converged. However, contrary to the over-prediction for the adiabatic resolution, we observe an under-prediction of $\approx 4\%$ of force and torque with explicit (fine mesh) and $\approx 5\%$ with implicit (fine mesh) formulation for the fully coupled case in comparison to the incremental results. This difference could come from the difference in the formulation for thermal problem in the two solvers. A speed-up of 13-15 times is achieved with the different steady-state formulations tested.

Contact formulation	Shape Corr. method	Mesh	Conv. Incs/Iters	Comp. Time (min)	Tool Force	Tool Torque
					(T)	(kN-m)
ForgeNxt®2016	SUPG-dif ($\alpha = 1/2$)	Coarse	×	×	108.18 ±0.06	86.81 ±0.8
		Fine	×	×	107.16 ±0.12	85.96 ±0.75
Steady-State Explicit	SUPG-dif ($\alpha = 1/3$)	Coarse	17	7.1	108.71	86.63
		Fine	17	11.87	107.97	86.53
Steady-State Implicit	SUPG-dif ($\alpha = 1/3$)	Coarse	17	7.33	108.1	86.37
		Fine	16	11.05	107.24	85.82
ForgeNxt®inc		Coarse	391	104	113.5	91.3
		Fine	448	146	112.9	91.1

Table 4.6.5: Global performance of the VP-Test-Case-03-fully coupled with the updated steady-state algorithm in comparison to the incremental solution (on 12 cores)

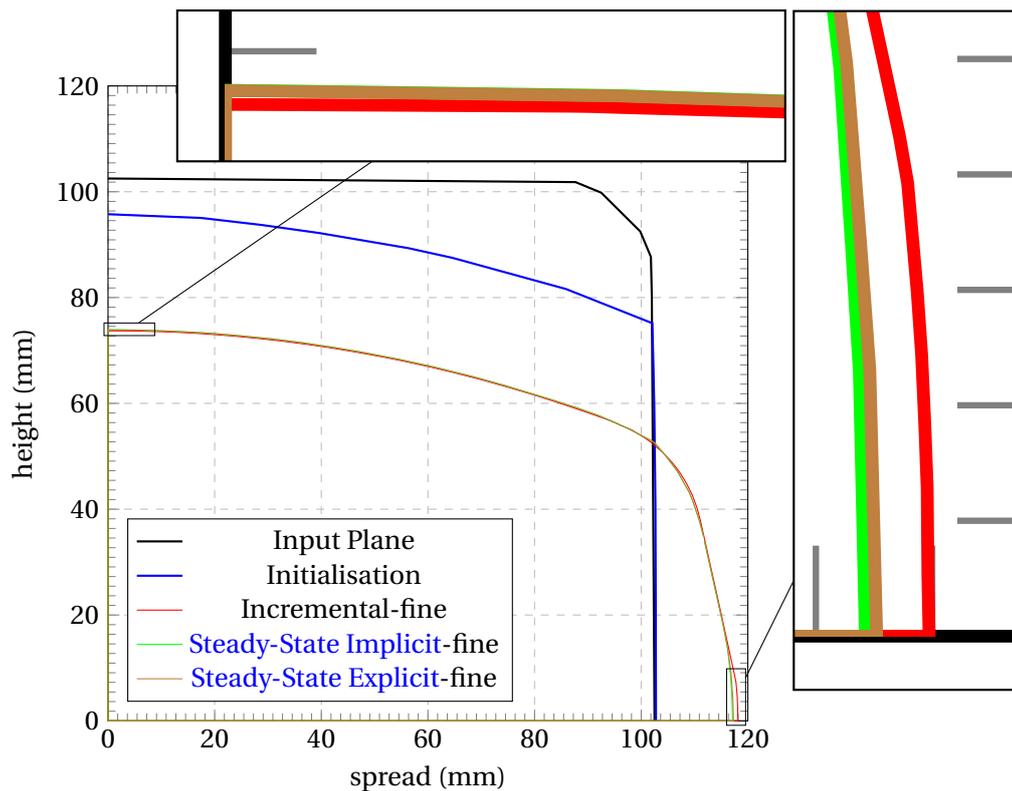


Figure 4.6.20: Comparison of deformed section (steady-state) shape of the VP-Test-Case-03-fully coupled with the steady-state and incremental formulations

The steady-state shape resulting from different steady-state formulations in comparison to the incremental formulation are shown in Figure 4.6.20. We can see that the steady-state algorithm with consistent contact formulation is able to predict very well the global shape after rolling in steady-state. However, the lateral spread is under-predicted with steady-state formulations in comparison to the incremental resolution. This is consistent with the fact that the tool force with the incremental resolution is higher in comparison to the steady-state resolution. A more detailed

comparison of solution fields is done for this test-case with the fully coupled thermal-structural resolution. Firstly, the axial velocity and the contact surface predicted with the incremental and steady-state solutions are presented in Figure 4.6.21. A gradually increasing velocity can be seen as the material is rolled between the tools. The binary contact can be interpreted as the nodes of the workpiece in contact with the tool surface. It can be clearly seen that both the top and the side face of the workpiece get in contact with the tool during the rolling. It can also be observed that the steady-state algorithm with consistent implicit contact formulation is able to predict accurately both the velocity and contact fields computed with the incremental formulation.

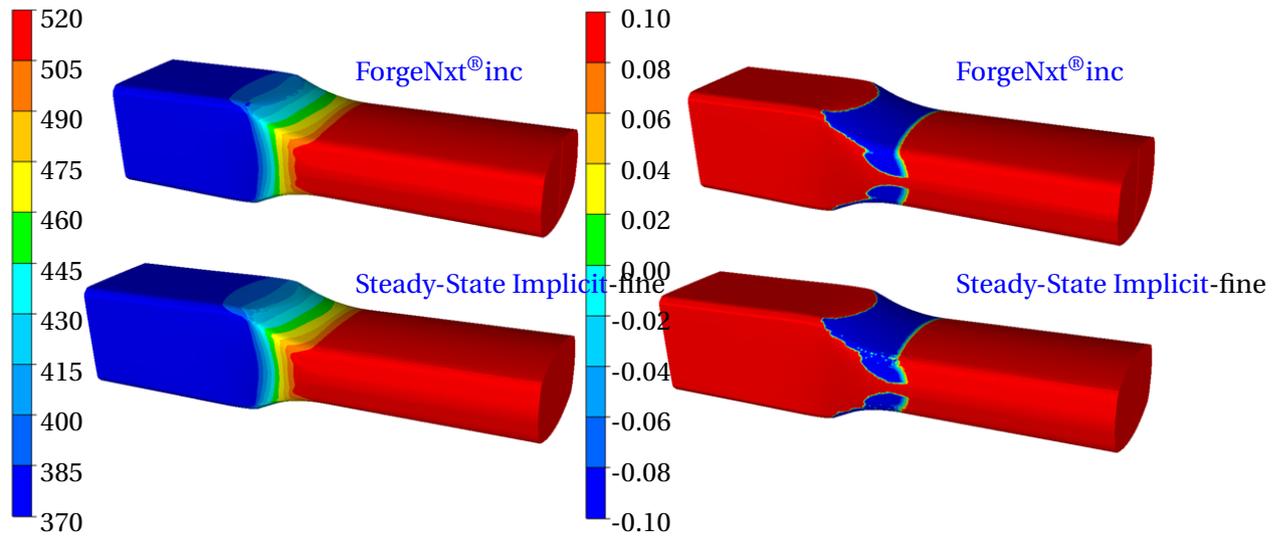


Figure 4.6.21: Comparison of axial velocity (on left) and binary contact (on right) for VP-Test-Case-03-fully coupled with Steady-State Implicit in comparison to the incremental solution

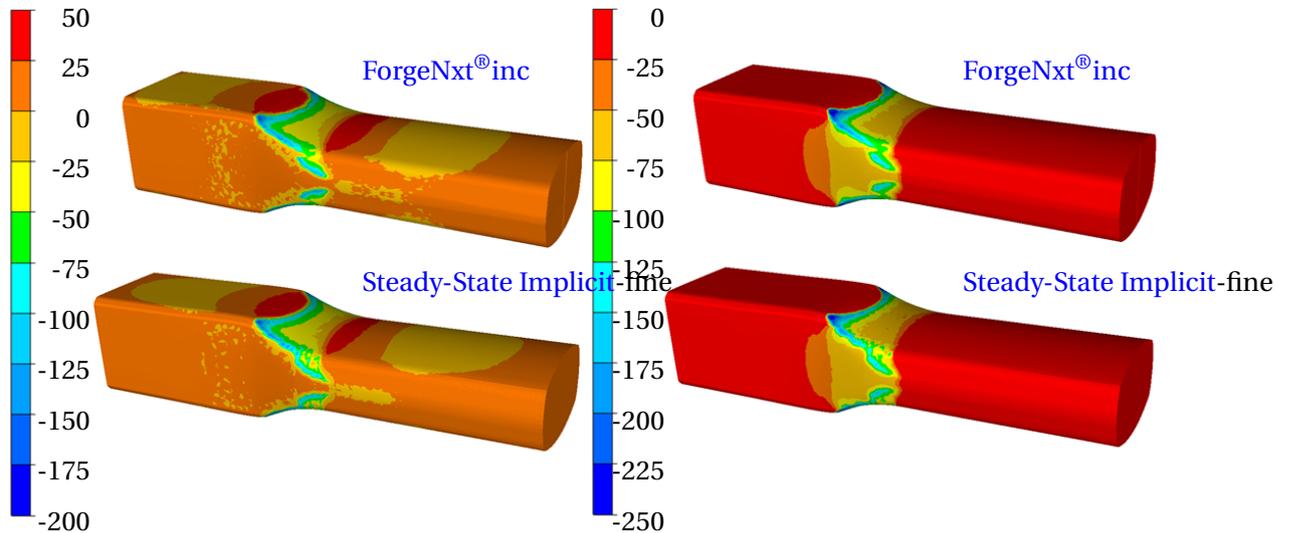


Figure 4.6.22: Comparison of stress in XX (transverse, in MPa, on left) and YY (vertical, in MPa, on right) directions (in MPa) respectively for VP-Test-Case-03-fully coupled with Steady-State Implicit in comparison to the incremental solution

In Figure 4.6.22, the stresses in XX and YY directions, which correspond to the transverse and vertical directions respectively, are compared. During rolling, the tool exerts a compressive force in the vertical direction and thus forces the material to flow in the other directions. However, this flow is constrained in the transverse direction as well. Hence, a compressive XX stress develops below the tool contact. The sign of the stress in the rolling direction ZZ (shown in Figure 4.6.23 on the left) is compressive below the tool contact and tensile right after. The von Mises stresses

for the test-case is presented on the right. The prediction of both these fields with the steady-state algorithm is in very good agreement in comparison to the incremental resolution.

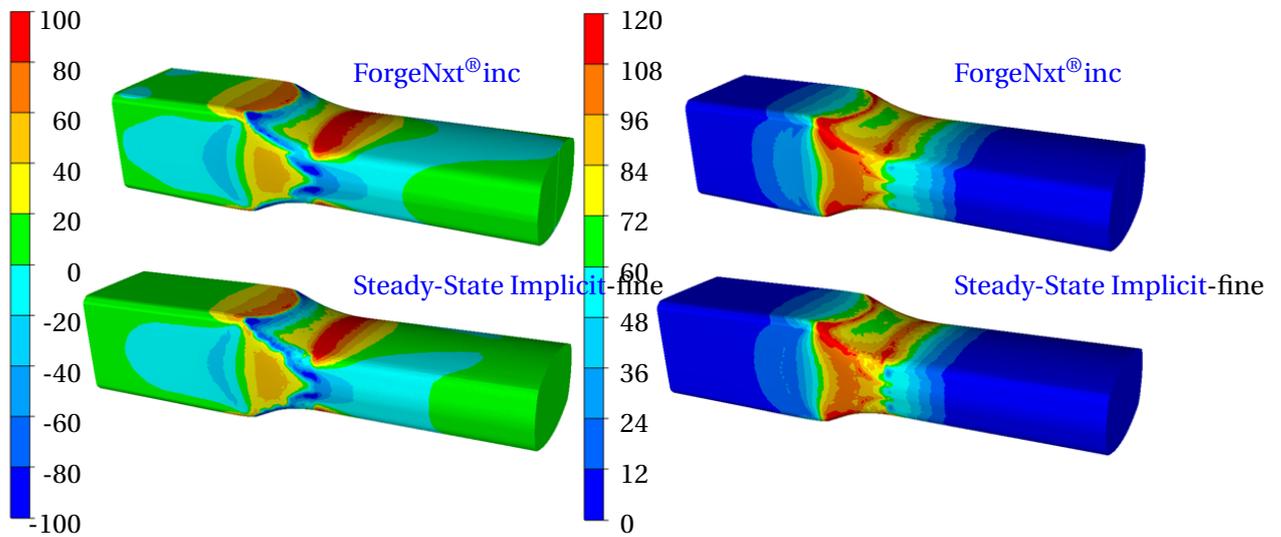


Figure 4.6.23: Comparison of ZZ (rolling direction, in MPa, on left) direction stress and von Mises stress (in MPa, on right) for VP-Test-Case-03 with Steady-State Implicit in comparison to the incremental solution

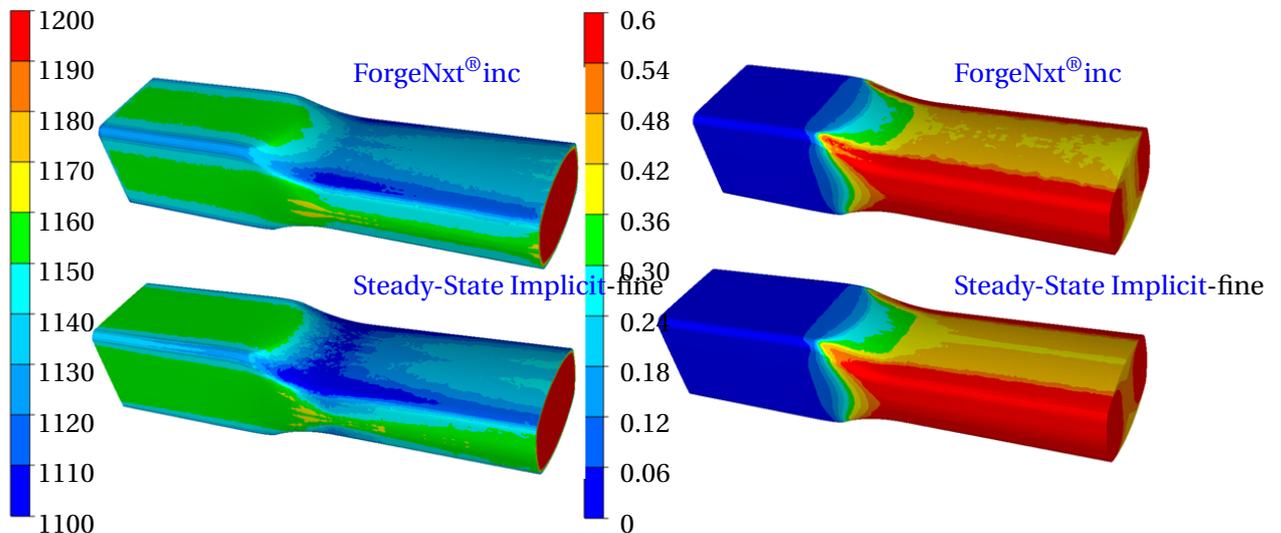


Figure 4.6.24: Comparison of Temperature (left) and Equivalent strain (right) for VP-Test-Case-03-fully coupled with Steady-State Implicit in comparison to the incremental solution

The comparison of the temperature fields at steady-state from the incremental and steady-state formulation are presented in Figure 4.6.24 on the left. We observe a good agreement of the temperature field away from the contact surface with the two formulations. However, the workpiece surface below the tool contact with the steady-state formulation is found to be at a lower temperature in comparison to the incremental formulation. This difference could come due to the difference in thermal formulation inside ForgeNxt®2016 that could result in the difference in the tool force and torque (between incremental and steady-state formulation results) observed in Table 4.6.5. The equivalent strain field with the two formulations is presented in Figure 4.6.24 on the right. We see a reasonable agreement of the fields away and below the tool contact. However, we can see an under-prediction of strain with the steady-state formulation at the workpiece edge in contact with the tool.

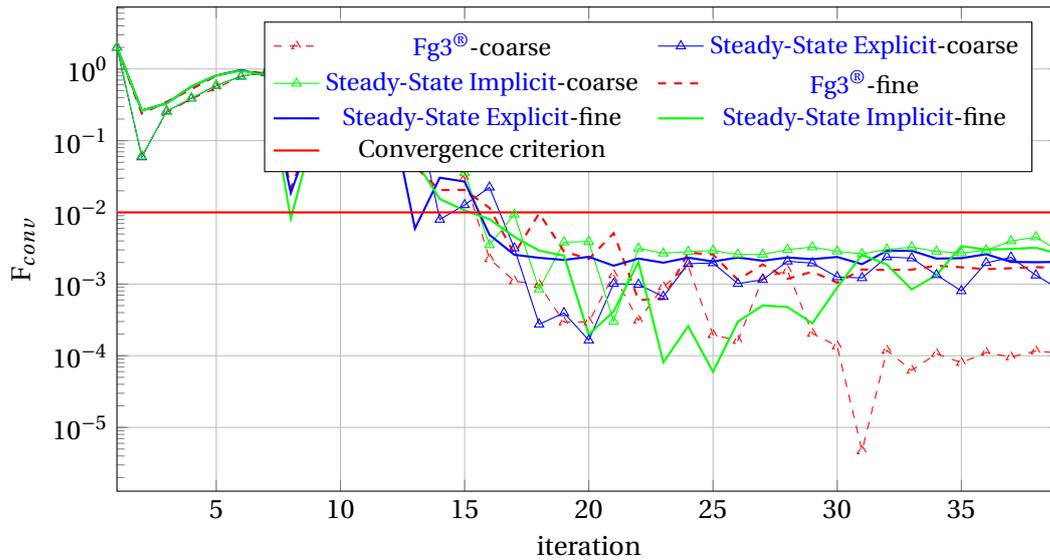


Figure 4.6.25: Comparison of Tool force convergence for *VP-Test-Case-03* fully coupled with different steady-state formulations

The solution obtained with the steady-state formulations is analyzed for the stability of the convergence parameters for the fully coupled *VP-Test-Case-03*. The comparison of tool force convergence ((3.4.6d)) with the different tests studied are presented in Figure 4.6.25. Globally, we observe that the force converges with all the formulations within 16-17 iterations, and the residual remains stable ($\approx 10^{-3}$ to 10^{-4}) and converged afterward. The minimum residual is achieved with the *ForgeNxt*[®] 2016 resolution of the coarse mesh model. For the fine mesh model, the implicit formulation results in the least residual.

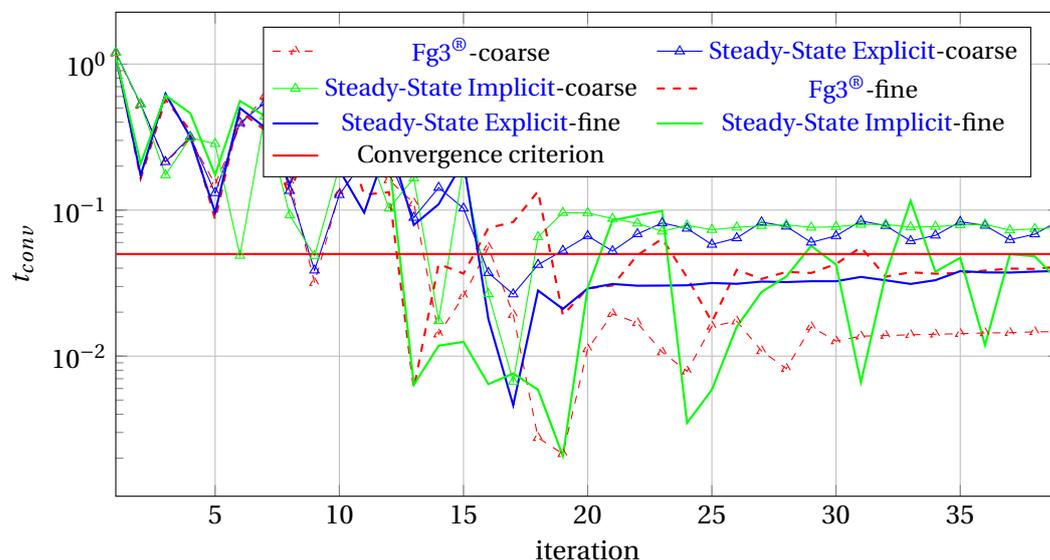


Figure 4.6.26: Comparison of Geometry convergence for *VP-Test-Case-03* fully coupled with different steady-state formulations

Next, the geometry convergence ((3.4.6a)) is presented in Figure 4.6.26. The stability of this parameter provides a good comparison of the different steady-state formulations tested. With the coarse mesh, the *ForgeNxt*[®] 2016 formulation is undoubtedly the best as it yields the lowest and stable residual. With the fine mesh, the best solution in terms of the minimum residual is achieved with implicit contact formulation (0.2%) but the solution becomes unstable as the resolution progresses further. In comparison, the explicit method results in a stable converged solution, even though with a larger variation (3%). The equivalent strain convergence (3.4.6c) with the different

steady-state formulations is compared in Figure 4.6.27. The ForgeNxt[®]2016 algorithm, like the VP-Test-Case-02 does not converge because of this criterion. On the other hand, the equivalent strain field converges and remains stable with both explicit and implicit contact formulations presented with all the different meshes tested. With the coarse mesh, the explicit contact formulation results in the minimum residual. With the fine mesh, the implicit contact formulation results in the minimum residual.

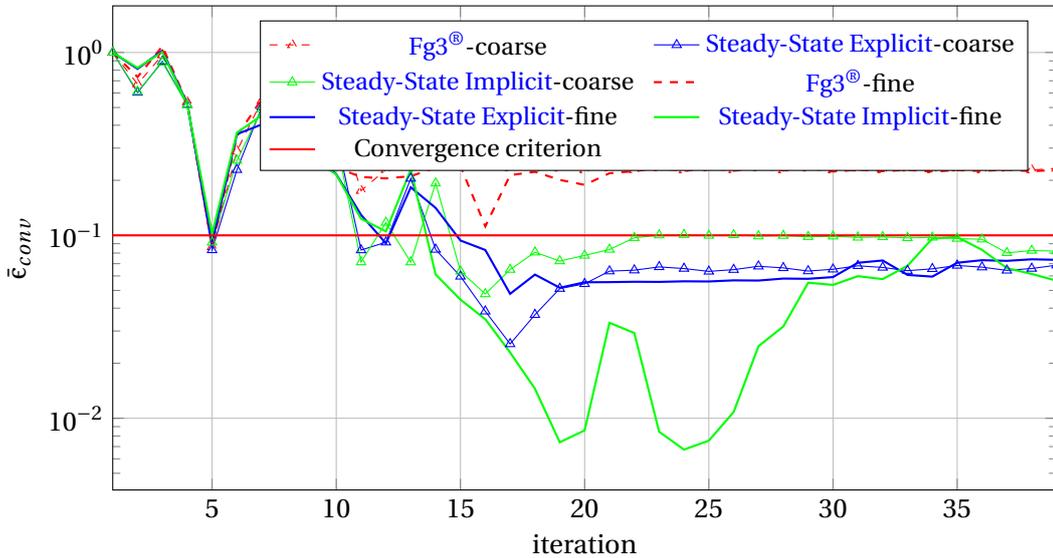


Figure 4.6.27: Comparison of Equivalent strain convergence for VP-Test-Case-03-fully coupled with different steady-state formulations

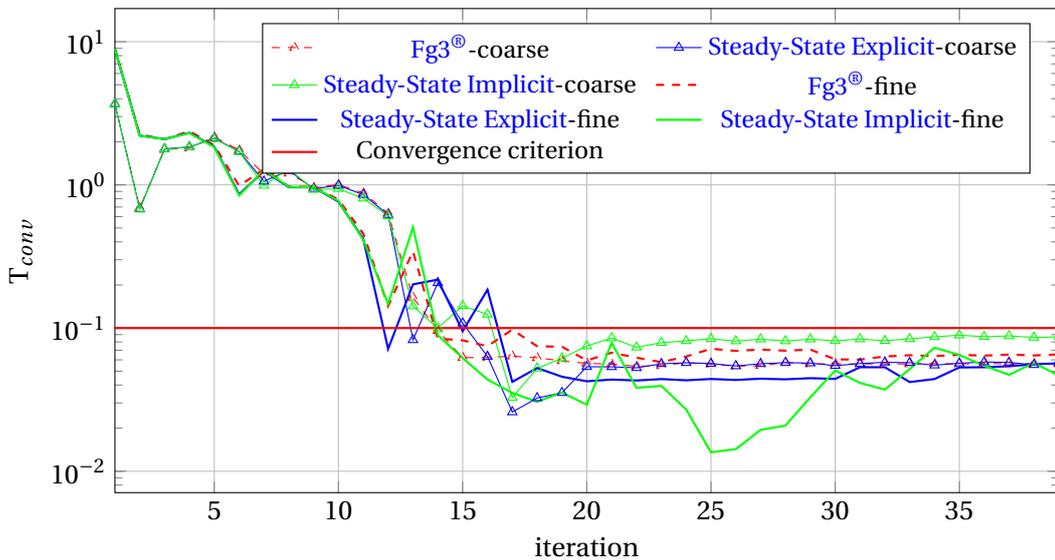


Figure 4.6.28: Comparison of Temperature convergence for VP-Test-Case-03-fully coupled with different steady-state formulations

The comparison of the Temperature convergence (3.4.6b) is presented in Figure 4.6.28. We can observe that the convergence criterion is met with all the presented formulations with both coarse and fine meshes. With the coarse mesh, ForgeNxt[®]2016 formulation and consistent explicit contact formulation results in the minimum (and similar) residual. However, with the fine mesh, the implicit formulation results in the minimum residual.

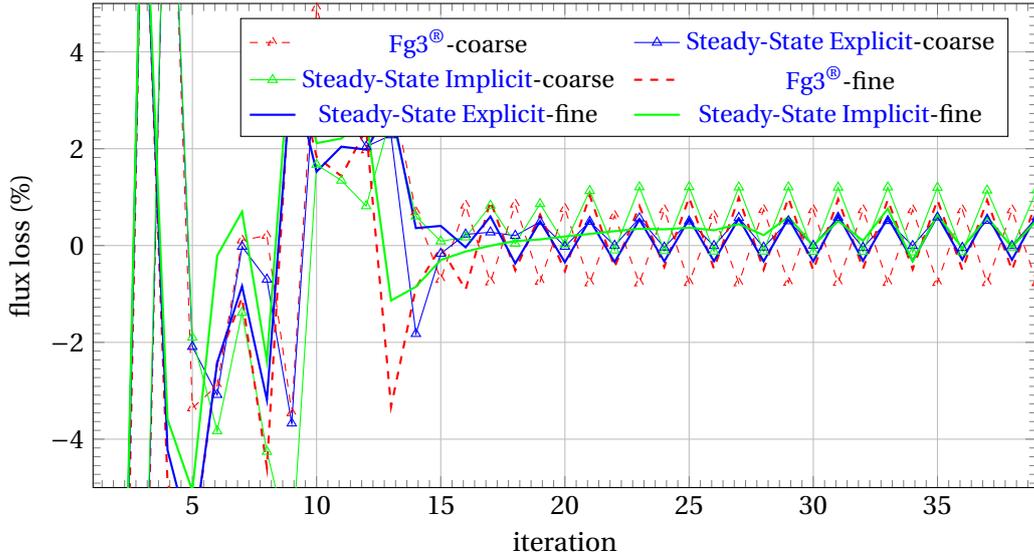


Figure 4.6.29: Comparison of flux loss for *VP-Test-Case-03* fully coupled with different steady-state formulations

It is to be reminded to the notice of the reader that the convergence of the parameter does not assure a good agreement with the incremental solution but only a non-oscillating (or oscillation with small magnitude) solution, which has low variation in the staggered fixed-point algorithm. The difference in the Temperature field with respect to the incremental solution is merely due to different formulations, where the *ForgeNxt[®]inc* uses several tricks in the thermal solver to avoid large temperature shocks (for example). The flux loss function in equation (3.5.1) with different formulations are compared in Figure 4.6.29. We see that for the coarse mesh the minimum flux loss is seen with the explicit formulations to be below 0.25%. With fine mesh, the flux loss in minimum $\approx 0.3\%$ with the implicit contact formulation.

Conclusions from solution

From the fully coupled thermal-mechanical solution of the *VP-Test-Case-03*, we conclude that the modified steady-state formulation is robust in comparison to the *ForgeNxt[®]2016* formulation. We see a good agreement between the mechanical results from steady-state formulation and the incremental formulation. However, while comparing with the results from the adiabatic resolution of the same problem, we can also conclude that the observed difference in results is originated from the thermal solver which has inherent differences in comparison to the *ForgeNxt[®]inc* thermal solver. In terms of the solution accuracy, both explicit and implicit contact formulation results in a similar solution, however, the implicit contact formulation attains better stability and the residual converges faster with the mesh refinement.

4.7 Conclusions from the chapter

The main focus of this chapter was to improve the robustness of the steady-state algorithm presented in Chapter 3. A hypothesis is made at first behind the non-convergence of the steady-state algorithm. The contact coupling is analyzed and the inconsistency in the contact equations in the two steps of the *ForgeNxt[®]2016* algorithm is highlighted. An idea to find consistent contact coupling is propounded by invoking a *nodal condensation* technique on the weighted residual form of the free-surface equation. Thus the consistent contact coupling aims to replace the integral form of the free-surface equation with the nodal form and then use the nodal form to find the consistent contact condition (in the explicit form) for the velocity computation in the mechanical solver. This entails the replacement of the existing scalar product of the velocity \vec{v} with the tool normal \vec{n}^{tool} by

the scalar product of velocity \vec{v} with an upwind normal \hat{u}_k . From the analytical shape test-cases, it is concluded that the nodal form of the free-surface equation SUPG-NC is convergent, yet not as accurate as the integral form of the free-surface equation for predicting the free-surface shape. Hence, it was proposed to use the *nodal condensation* technique only to find a consistent contact condition, and to retain the integral form of free-surface equation SUPG for the shape computation with the semi-consistent coupling. The derived explicit contact condition is described on the previous configuration $\vec{x}^{(i)}$ or on the shape computed from the free-surface computation in the Step-2 of the previous iteration. The contact condition is also attempted to be expressed in an implicit form with its expression in the current configuration $\vec{x}^{(i+1)}$. Respective consistent contact conditions are also derived for the singular geometries with edges, with 2-DoF formulations. The modified steady-state algorithm with the explicit and implicit contact condition is tested with three hot rolling test-cases with increasing complexity in terms of geometry, material property, and the coupling with the thermo-mechanical behavior. The modified formulation improved the results for an already-converging-with-existing-ForgeNxt[®]2016-formulation, especially in terms of the equivalent strain and shape prediction. The new consistent contact formulation also improves the robustness of the algorithm as mostly all the test-cases tested with the new formulation have all the steady-state convergence parameters converged. The modified formulation is tested with the deformation coupling, and thermal dependency, and the mechanical solver proves to be robust enough to handle the complexities of a rolling problem. It is also to be highlighted that the implicit formulation further improves the stability of the solution in comparison to the explicit formulation.

Résumé

L'objectif principal de ce chapitre était d'améliorer la robustesse de l'algorithme stationnaire présenté dans le chapitre 3. Tout d'abord, une hypothèse est faite sur la non-convergence de l'algorithme: l'instabilité observée est liée avec l'inconsistance des équations de contact dans les deux étapes de l'algorithme ForgeNxt® 2016. Une idée pour trouver les équations de contact cohérentes est proposée en invoquant une technique de condensation nodale sur la forme faible de l'équation de surface libre. La nouvelle equation du contact vise à remplacer la forme intégrale de l'équation de surface libre par la forme nodale, puis à utiliser la forme nodale pour trouver la condition de contact (sous la forme explicite) pour le calcul de la vitesse dans le solveur mécanique. Cela implique le remplacement du produit scalaire existant de la vitesse \vec{v} par la normale à la surface de l'outil \vec{n}^{tool} par le produit scalaire de la vitesse \vec{v} par une normale amont \hat{u}_k . D'après les cas tests de forme analytique, nous concluons que la forme nodale de l'équation de surface libre SUPG-NC est convergente, mais moins précise que la forme intégrale de l'équation de surface libre pour prédire la forme de cette surface. Par conséquent, il a été proposé d'utiliser la technique de condensation nodale uniquement pour trouver une condition de contact cohérente et de conserver la forme intégrale de l'équation de surface libre SUPG pour le calcul de la forme avec un couplage semi-cohérent. La condition de contact doit également être exprimée sous une forme implicite, comme dans la formulation incrémentale, avec son expression dans la configuration actuelle. Des conditions de contact cohérentes respectives sont également dérivées pour les géométries singulières comme des arêtes, avec des formulations 2-DoF. L'algorithme stationnaire modifié, avec les nouvelles conditions de contact en explicite et en implicite, est testé pour trois cas tests de laminage à chaud avec une complexité croissante en termes de géométrie, de propriétés des matériaux et de couplage de comportement thermo-mécanique. La formulation modifiée a amélioré les résultats pour une formulation ForgeNxt® 2016 déjà convergente, en particulier en termes de prédiction de déformation équivalente et de forme. La nouvelle formulation de contact cohérente améliore les résultats pour le cas de laminage à chaud simple et aussi s'avère plus robuste avec les cas de laminage de forme avec une meilleure convergence. La formulation modifiée est testée avec le couplage de la déformation et la dépendance thermique, et le solveur mécanique s'avère suffisamment robuste pour gérer les complexités d'un problème de laminage. Il convient également de souligner que la formulation implicite améliore encore la stabilité de la solution par rapport à la formulation explicite.

Bibliography

- [1] A. Mizukami and T.J.R. Hughes. A Petrov-Galerkin finite element method for convection-dominated flows: An accurate upwinding technique for satisfying the maximum principle. *Computer Methods in Applied Mechanics and Engineering*, 50(2):181 – 193, 1985. ISSN 0045-7825.
- [2] U. Ripert. *Méthode itérative de recherche de l'état stationnaire des procédés de mise en forme : application au laminage*. PhD thesis, ENSMP, 2014. URL <http://www.theses.fr/2014ENMP0006>.
- [3] M. Vacance, E. Massoni, and J.L. Chenot. Multi stand pipe mill finite element model. *Journal of Materials Processing Technology*, 24:421 – 430, 1990. ISSN 0924-0136. doi: [https://doi.org/10.1016/0924-0136\(90\)90203-7](https://doi.org/10.1016/0924-0136(90)90203-7).

Chapter 5

Steady-state resolution with history dependent EVP behavior

Contents

5.1	Introduction to elastoplasticity	123
5.1.1	Prandtl Reuss rate description	123
5.1.2	Incremental Prandtl-Reuss equations	125
5.1.3	Resolution of elastoplasticity in steady-state formulations	127
5.2	New ForgeNxt®2016 EVP resolution algorithm	129
5.2.1	Building streamline in ForgeNxt®2016 steady-state solver	129
5.2.2	Constant pseudo-time-step for unstructured mesh	131
5.2.3	Basic ForgeNxt®2016 EVP algorithm	132
5.2.4	Detailed Sub-iter loop for the elasto-viscoplastic resolution in ForgeNxt®2016 EVP	132
5.2.5	Newton-Raphson resolution of Stokes Problem	136
5.2.6	Detailed ForgeNxt®2016 EVP Algorithm-1	138
5.2.7	Convergence criterion/a for the Sub-iter loop	139
5.3	Validation of ForgeNxt®2016 EVP Algorithm-1 with a simple cold rolling problem	139
5.3.1	EVP-Test-Case-01 model details	139
5.3.2	Initialization with viscoplastic solution	140
5.3.3	Different EVP-Test-Case-01 model variants tested	141
5.3.4	Comparison of EVP-Test-Case-01 results	141
5.3.5	Conclusions from the EVP-Test-Case-01 results	145
5.4	Improvement of speedup with ForgeNxt®2016 EVP Algorithm-2	145
5.4.1	Detailed ForgeNxt®2016 EVP Algorithm-2	146
5.4.2	Updated Stream-loop and Sub-iter loop convergence criteria	146
5.4.3	EVP-Test-Case-01 resolution with ForgeNxt®2016 EVP Algorithm-2	147
5.5	Validation of ForgeNxt®2016 EVP algorithm with complex test cases	147
5.5.1	EVP-Test-Case-02 resolution with ForgeNxt®2016 EVP algorithm	148
5.5.2	EVP-Test-Case-03 resolution with ForgeNxt®2016 EVP algorithm	150
5.5.3	Conclusions from the EVP-Test-Case-02 and EVP-Test-Case-03 results	152
5.6	Conclusions from the Chapter	153

One of the biggest challenges in the steady-state formulation of metal forming processes is the modeling of history-dependent material behavior. In the previous chapters, a certain kind of history-dependence was encountered in the viscoplastic framework with work hardening and thermal coupling. The present chapter is focused on a specific type of history-dependency with elastic recovery, in which the dependence comes from the stress at a previously known state (or time-step). As mentioned in the Chapters 1 and 2, such scenarios are envisaged when the rolling is done in cold conditions. In such conditions, the elastic strain cannot be neglected, therefore, elastoplastic or elasto-viscoplastic material models need to be considered.

It is of foremost importance to understand the mathematical framework for modeling an elastoplastic problem. In the Section 2.3.3, we saw different manners to model this material behavior. Our interest is in metal forming problems which are of hypoelastic, plasticity dominant nature and solved using Prandtl-Reuss equation (2.3.6a). The aim of the elastoplasticity resolution is to compute the elastoplastic stresses from an initial assumption of purely elastic/plastic material. The rate form of the equation cannot be solved with numerical methods; hence, an incremental form of the same is used. The latter can easily be coupled with an iterative solver to compute the unknown state-variables as a correction to an initial assumption. This method is generally used for solving elastoplastic problems with *incremental methods*, in which the state-variables are computed at an integration point at a time instant $t + \Delta t$ from their knowledge at t , and hence the history of state-variable is transported. The main objective of this resolution is to compute the incremental state-variables for Δt . This approach can also be adopted to compute the elastoplastic stresses in steady-state formulations as proposed by [3]. However, in the time-independent steady-state formulation, it is unavoidable to keep Prandtl-Reuss equations independent of time. Hence, the time-step in the incremental formulation is rather modeled with a pseudo-time-step $\Delta \tilde{t}$ in the steady-state formulation. In the former it is the time taken for a material point to move from one point to the other (pathline). In the steady-state formulations, the streamlines are coincident with the pathlines, hence, for any two points in the material, the point in the upwind describes the previous state of the one in the downwind. Hence, the pseudo-time-step makes possible to compute the updated state-variables from the incremental Prandtl-Reuss equations with backward integration.

The **Heterogeneous pseudo-time-step (HPTS)** approach, as introduced in Section 2.3.3, models the pseudo-time-step as the time taken to traverse the distance between two consecutive integration points ($int-1, int$) on a streamline. This method must ensure that the consecutive integration points lie on the same streamline, which is only possible with the chosen structured mesh. The streamline integration along the Gaussian points also ensures that the information of state-variables is naturally available. The streamline integration requires the continuous update of the stress field and the velocity until a converged stable field is obtained. For unstructured meshes, doing this is rather difficult, as the integration points are not necessarily aligned with the streamlines. A streamline-framework independent of the mesh is possible but tracing them becomes a tedious process of regular follow-up with mapping of variables, which is diffusive in nature. Hence, non-diffusive strategies become imperative to counter this problem. Succinctly, this chapter is divided into three main sections. The first section introduces the elastoplasticity Prandtl-Reuss equations in the respective rate and incremental forms followed by the strategy for adopting the latter to the steady-state formulations. The second section is focused on proposing a streamline-framework in the unstructured meshes and the adaptation of pseudo-time-step approach in ForgeNxt®2016. In the pursuit, some non-diffusive strategies are proposed and tested on simple cases. Subsequently, an algorithm for the resolution of the elastoplastic steady-state forming problem is proposed. The final section is focused on the testing of the algorithm with different cold rolling test-cases in ForgeNxt®2016 and the validation of results with the time-dependent incremental solution of the same problem in ForgeNxt®inc. At last, some speed-up strategies are also proposed and tested with the same problems.

5.1 Introduction to elastoplasticity

For the hypoelastic, plasticity dominant application of metal forming, the additive decomposition is enough to model the material behavior with the Prandtl-Reuss equations.

5.1.1 Prandtl Reuss rate description

The basis of elastoplastic modeling is the additive decomposition of the strain-rate tensor to the respective elastic $\dot{\boldsymbol{\epsilon}}^e$ and plastic $\dot{\boldsymbol{\epsilon}}^p$ parts, and the description of their relationship with the stress tensor.

$$\dot{\boldsymbol{\epsilon}} = \dot{\boldsymbol{\epsilon}}^e + \dot{\boldsymbol{\epsilon}}^p \quad (5.1.1)$$

Elasticity is characterized by the recovery behavior of the material to go back to unloaded state after the load is removed. Elasticity could be linear or nonlinear, but for the metal forming, we assume linear elasticity, as given below:

$$\overset{\vee}{\boldsymbol{\sigma}} = \mathbb{C} : \dot{\boldsymbol{\epsilon}}^e = 2\mu^e \dot{\boldsymbol{\epsilon}}^e + \lambda^e \text{tr}(\dot{\boldsymbol{\epsilon}}^e) \quad (5.1.2a)$$

$$\mu^e = \frac{E}{2(1+\nu)} \quad \text{and} \quad \lambda^e = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad (5.1.2b)$$

$$\overset{\vee}{\boldsymbol{\sigma}} = \dot{\boldsymbol{\sigma}} + \boldsymbol{\sigma} \boldsymbol{w} - \boldsymbol{w} \boldsymbol{\sigma} \quad (5.1.2c)$$

In the equation above, λ^e and μ^e are the Lamé's coefficients, which are generally constant for homogeneous materials. E is the Young modulus, ν is the Poisson coefficient, $\dot{\boldsymbol{\epsilon}}^e$ is the elastic strain-rate, and $\dot{\boldsymbol{\sigma}}$ is the material time derivative of stress and \boldsymbol{w} is the antisymmetric part of the velocity gradient. In practice, we use Truesdell's equation for elasticity introduced in Section 2.3.3, for keeping this relation free of the reference frame. For example, in Lam3[®], the Jaumann objective stress (5.1.2c) is used (ie the Hill-Sidoroff formulation of Prandtl-Reuss decomposition), but to avoid making the elastoplastic resolution more complex, the computation of the objective stress is delayed (from the rotation term \boldsymbol{w}) to the end of the time-step. Elastoplasticity models are used when the elastic recovery of the metal is sufficiently large for not to be neglected. Elastoplastic models are characterized by change of material behavior from elastic state to a plastic state after a yield limit σ_y is reached. The scalar function \bar{f} defines the yielding criteria:

$$\begin{cases} \bar{f}(\boldsymbol{\sigma}, \sigma_y) < 0 & \text{purely elastic} \\ \bar{f}(\boldsymbol{\sigma}, \sigma_y) = 0 & \text{plasticity} \end{cases} \quad (5.1.3)$$

The yield function is generally a mathematical representation of a failure theory (by plastic deformation), which could either be based on maximum principal stress (Tresca criterion) or on maximum distortion energy (von Mises criterion) principle. We chose here maximum distortion energy principle to define the material yield criterion:

$$\bar{f} = \sqrt{\boldsymbol{s} : \boldsymbol{s}} - \sqrt{\frac{2}{3}} \sigma_y(\bar{\boldsymbol{\epsilon}}, \dot{\boldsymbol{\epsilon}}, T) \quad (5.1.4)$$

The plastic flow occurs when $\bar{f} = 0$, as described in equation (5.1.3). The kinematic hardening (see Figure 5.1.1a) can be modeled with the backstress \boldsymbol{X} . In this type of hardening, the yield surface retains its size and shape but is repositioned in the stress space. Hence, for the yield criteria, the origin of the stress space is shifted, and the deviatoric stress in the equation (5.1.4) is replaced with $\boldsymbol{s} - \boldsymbol{X}$. Alternatively, in isotropic hardening, the yield surface dilates as the yield limit becomes larger once a plastic state is reached as shown in Figure 5.1.1b.

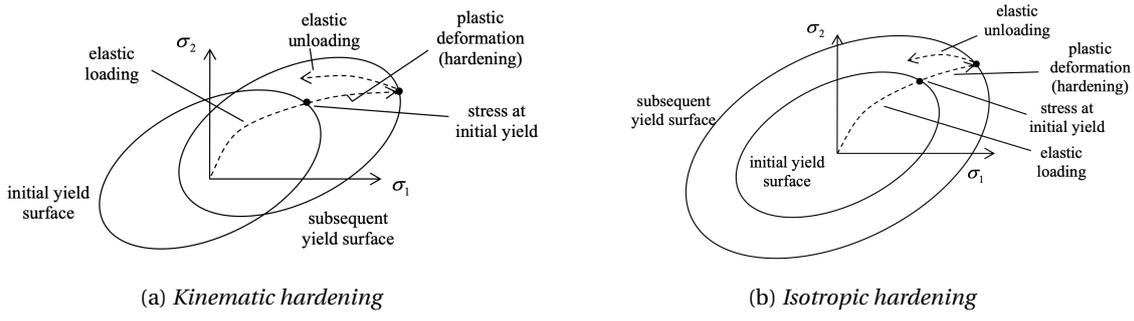


Figure 5.1.1: Graphical representation of hardening rules [1]

The plastic strain is perpendicular to the Yield surface with the Maximum Work principle. For dense metals, the plastic flow rule is associated to the yield criterion (normality rule), which can be mathematically related with the physical mechanism of dislocation glide on planar systems at definite thresholds. The yield surface is itself associated with the plastic deformation as the dislocation glides on planar systems at definite thresholds. Hence, the material flow condition is defined with the yield surface \bar{f} as follows:

$$\dot{\mathbf{e}}^p = \lambda^{pl} \frac{\partial \bar{f}}{\partial \boldsymbol{\sigma}} = \lambda^{pl} \mathbf{s}; \quad \lambda^{pl} = \frac{3\dot{\mathbf{e}}}{2\sigma_y} \quad (5.1.5)$$

In the above equation (5.1.5), λ^{pl} is the plasticity multiplier variable. For the resolution of the Prandtl-Reuss equations, the hydrostatic and deviatoric decomposition of the stress-rate is done and treated individually. The resolution of hydrostatic part is possible with the following relation:

$$\dot{p} = -\frac{1}{3} tr \dot{\boldsymbol{\sigma}} = -\frac{(2\mu^e + 3\lambda^e)}{3} tr \dot{\mathbf{e}}^e \quad (5.1.6)$$

Since the von Mises plasticity is isochoric (constant volume), we find that:

$$tr \dot{\mathbf{e}}^p = 0 \quad (5.1.7a)$$

$$tr \dot{\mathbf{e}} = tr \dot{\mathbf{e}}^e \quad (5.1.7b)$$

This implies that the computation of pressure rate in equation (5.1.6) is trivial as it can be computed from the strain-rate from the equation (5.1.8a). The strain-rate is found directly from the computed velocities with the relation (5.1.8b).

$$\dot{p} = -\frac{(2\mu^e + 3\lambda^e)}{3} tr \dot{\mathbf{e}} \quad (5.1.8a)$$

$$\dot{\mathbf{e}} = \frac{1}{2} (\nabla \vec{v} + \nabla^T \vec{v}) \quad (5.1.8b)$$

Similarly for finding the rate of the deviatoric part of the stress tensor, we invoke the decomposition of the deviatoric strain-rate to its elastic and plastic components, as done for the total strain-rate in equation (5.1.1).

$$\dot{\mathbf{e}} = \dot{\mathbf{e}}^e + \dot{\mathbf{e}}^p \quad (5.1.9)$$

The elastic and plastic parts of the deviatoric strain-rate are related to the deviatoric stress and its rate as follows:

$$\dot{\mathbf{s}} = \dot{\boldsymbol{\sigma}} - \frac{1}{3} tr \dot{\boldsymbol{\sigma}} \mathbf{1} = 2\mu^e \dot{\mathbf{e}}^e - \frac{2\mu^e}{3} tr (\dot{\mathbf{e}}^e) \mathbf{1} = 2\mu^e \dot{\mathbf{e}}^e \quad (5.1.10a)$$

$$\dot{\mathbf{e}}^p = \dot{\mathbf{e}}^p = \lambda^{pl} \mathbf{s} \quad (5.1.10b)$$

Next, we can replace these relations in the additive decomposition of the deviatoric strain-rate in equation (5.1.9) and thus find the following ordinary differential equation which can be resolved to compute the evolution of the deviatoric stress.

$$\dot{\mathbf{s}} + 2\mu^e \frac{3\dot{\bar{\epsilon}}}{2\sigma_y} \mathbf{s} = 2\mu^e \dot{\epsilon} \quad (5.1.11)$$

The equation above validates the claim about the history-dependence of the problem due to the added elastic contribution. This means that the stress at a given state is obtained from a previously known state by the integration of this equation. It is also to be noted that the above-mentioned equation needs the resolution of six different nonlinear equations to compute the deviatoric stress tensor. In another possibility, we can also find a unique scalar (nonlinear) equation to find the complete stress-state. This method is discussed in the following section describing the incremental form of elastoplasticity.

Prandtl-Reuss equations with elasto-viscoplasticity

In general, the cold forming problem can be modeled with the rate-dependent plasticity or elasto-viscoplasticity. In this form, the stress-strain plastic flow behavior changes with the loading rate. The elasto-viscoplastic strain-rate can be decomposed (Prandtl-Reuss or additive) to the respective elastic and viscoplastic parts.

$$\dot{\epsilon} = \dot{\epsilon}^e + \dot{\epsilon}^{vp} \quad (5.1.12)$$

The elastic part of the strain-rate tensor is related to the stress tensor $\boldsymbol{\sigma}$ with the relation (5.1.2) and the viscoplastic part is related to the deviatoric stress tensor \mathbf{s} with the relation (5.1.13) found from the derivative of the viscoplastic potential Φ^{vp} with respect to the stress. The relation obtained is similar to the one with plastic strain-rate in equation (5.1.10b). So, in principle, the problem formulation is the same for both elastoplastic and elasto-viscoplastic material models. In the end, the same time-dependent equations (5.1.8) and (5.1.11) must be resolved to compute the elasto-plastic/elasto-viscoplastic stress state.

$$\dot{\epsilon}^{vp} = \frac{\partial \Phi^{vp}}{\partial \boldsymbol{\sigma}} = \frac{3}{2} \frac{\dot{\bar{\epsilon}}}{\bar{\sigma}} \mathbf{s} = \lambda^{pl} \mathbf{s} \quad (5.1.13)$$

5.1.2 Incremental Prandtl-Reuss equations

With the Finite Element method, it is more convenient to describe the Prandtl-Reuss equations in the incremental form to find the incremental elastoplastic stress for the time-step Δt . The shape, elastoplastic stress field and the deformation field at time $t + \Delta t$ as follows:

$$\bar{\mathbf{x}}_{t+\Delta t} = \bar{\mathbf{x}}_t + \Delta \bar{\mathbf{x}}_{t+\Delta t} \quad (5.1.14a)$$

$$\boldsymbol{\sigma}_{t+\Delta t} = \boldsymbol{\sigma}_t + \Delta \boldsymbol{\sigma}_{t+\Delta t} \quad (5.1.14b)$$

$$\bar{\boldsymbol{\epsilon}}_{t+\Delta t} = \bar{\boldsymbol{\epsilon}}_t + \Delta \bar{\boldsymbol{\epsilon}}_{t+\Delta t} \quad (5.1.14c)$$

In the Lagrangian (incremental) formulations, the stress at the time $t + \Delta t$ at the integration points is computed from stress at time t with equation (5.1.14b). However, the material point evolves and as the shape is updated, the integration point moves in the flow direction with its new position defined with the equation (5.1.14a). When the steady-state is achieved, the pathlines coincide with the streamlines, hence, all the material points have the same trajectory between the time t and $t + \Delta t$ (see Figure 5.1.2).

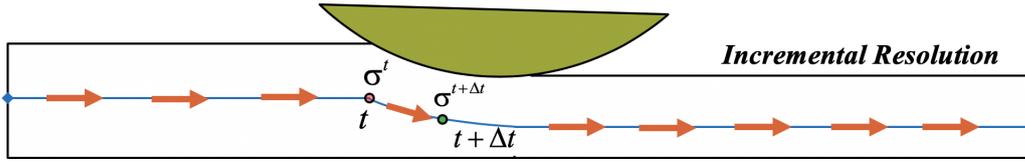


Figure 5.1.2: Incremental computation of elastoplastic stress

In the equation (5.1.14b), the stress increment $\Delta\boldsymbol{\sigma}$ is computed with the Prandtl-Reuss equations, but in the incremental form. In this form, the increment $\Delta\boldsymbol{f}$ of each state-variable \boldsymbol{f} is computed with equation (2.3.11) for the time-increment Δt . To do so, firstly, we can describe the stress update $\Delta\boldsymbol{\sigma}$ with the decomposition of the stress into deviatoric and hydrostatic parts as seen in equation (3.2.2a).

$$\Delta\boldsymbol{\sigma} = \Delta\boldsymbol{s} - \Delta p\mathbb{1} \quad (5.1.15)$$

We can model the incremental hydrostatic pressure Δp and deviatoric stress $\Delta\boldsymbol{s}$ terms to model the incremental elastoplastic stress.

Hydrostatic pressure

The computation of the hydrostatic pressure increment is trivial following the relation obtained in equation (5.1.10).

$$\Delta p = -\frac{(2\mu^e + 3\lambda^e)}{3} \text{tr} \Delta\boldsymbol{\epsilon} \quad (5.1.16)$$

Deviatoric stress

The incremental deviatoric stress can be computed following the steps for the temporal (rate) form between equations (5.1.7) to equation (5.1.11). The isochoric nature of the plastic strain increment leads to the following relation

$$\text{tr} \Delta\boldsymbol{\epsilon}^p = 0 \quad (5.1.17a)$$

$$\text{tr} \Delta\boldsymbol{\epsilon} = \text{tr} \Delta\boldsymbol{\epsilon}^e \quad (5.1.17b)$$

Next, we decompose the incremental deviatoric strain tensor into the respective elastic and plastic parts (5.1.18). The deviatoric elastic strain increment is related to the deviatoric stress with the equation (5.1.19a) and the deviatoric plastic strain is found from the updated deviatoric stress $\boldsymbol{s} + \Delta\boldsymbol{s}$ with (5.1.19b), which comes as a result of the choice of backward Euler integration. The plasticity multiplier λ^{pl} is computed from the equivalent strain increment with equation (5.1.19c).

$$\Delta\boldsymbol{\epsilon} = \Delta\boldsymbol{\epsilon}^e + \Delta\boldsymbol{\epsilon}^p \quad (5.1.18)$$

$$\Delta\boldsymbol{s} = \Delta\boldsymbol{\sigma} + \Delta p\mathbb{1} = 2\mu^e \Delta\boldsymbol{\epsilon}^e - \frac{2\mu^e}{3} \text{tr}(\Delta\boldsymbol{\epsilon}^e) = 2\mu^e \Delta\boldsymbol{\epsilon}^e \quad (5.1.19a)$$

$$\Delta\boldsymbol{\epsilon}^p = \Delta\boldsymbol{\epsilon}^p = \lambda^{pl} (\boldsymbol{s} + \Delta\boldsymbol{s}) \quad (5.1.19b)$$

$$\lambda^{pl} = \frac{3\Delta\bar{\epsilon}}{2\sigma_y(\bar{\epsilon} + \Delta\bar{\epsilon})} \quad (5.1.19c)$$

Once the components of the deviatoric strain increment are known, we can describe the deviatoric stress increment with the total deviatoric strain increment and the plasticity multiplier λ^{pl} .

$$\Delta\boldsymbol{s} = 2\mu^e \left(\Delta\boldsymbol{\epsilon} - \lambda^{pl} (\boldsymbol{s} + \Delta\boldsymbol{s}) \right) = 2\mu^e \frac{\Delta\boldsymbol{\epsilon} - \lambda^{pl} \boldsymbol{s}}{1 + 2\mu^e \lambda^{pl}} \quad (5.1.20)$$

Using the above derived relations, we can find the updated deviatoric stress $\mathbf{s} + \Delta\mathbf{s}$ and deviatoric plastic strain increment $\Delta\mathbf{e}^p$ in terms of the plasticity multiplier λ^{pl} .

$$\mathbf{s} + \Delta\mathbf{s} = \frac{\mathbf{s} + 2\mu^e \Delta\mathbf{e}}{1 + 2\mu^e \lambda^{pl}} \quad (5.1.21)$$

$$\Delta\mathbf{e}^p = \lambda^{pl} \frac{\mathbf{s} + 2\mu^e \Delta\mathbf{e}}{1 + 2\mu^e \lambda^{pl}} \quad (5.1.22)$$

The deviatoric plastic strain increment must comply to the consistency condition as following:

$$\Delta\bar{\epsilon} = \sqrt{\frac{2}{3} \Delta\mathbf{e}^p : \Delta\mathbf{e}^p} \quad (5.1.23)$$

Using the equations (5.1.21) and (5.1.22) and replacing into equation (5.1.23), we can describe the consistency condition with the following nonlinear scalar equation which can be solved to compute the unknown equivalent deformation $\Delta\bar{\epsilon}$ increment:

$$\sigma_y(\bar{\epsilon} + \Delta\bar{\epsilon}) + 3\mu^e \Delta\bar{\epsilon} = \sqrt{\frac{3}{2} (\mathbf{s} + 2\mu^e \Delta\mathbf{e}) : (\mathbf{s} + 2\mu^e \Delta\mathbf{e})} \quad (5.1.24)$$

In the equation (5.1.24), the term $\mathbf{s} + 2\mu^e \Delta\mathbf{e}$ is the elastic trial stress at initialization. The scalar equation computes the equivalent deformation $\Delta\bar{\epsilon}$ increment on the dilated yield surface. On the computation of equivalent deformation $\Delta\bar{\epsilon}$ increment, the plasticity multiplier λ^{pl} can be computed from the equation (5.1.19c). As we have already the description of plastic strain increment and the updated deviatoric stress in terms of the plasticity multiplier, they can be easily computed with the equations (5.1.22) and (5.1.21) respectively.

5.1.3 Resolution of elastoplasticity in steady-state formulations

For the resolution of the elastoplasticity problem in a steady-state, two approaches are commonly used in the literature. The first approach aims at describing a multi-field formulation in $\vec{v}, p, \sigma, \bar{\epsilon}, T, \iota$ which can be solved with direct methods. This approach entails multiple convection equations to be resolved iteratively until the convergence of the state variables of the problem. However, given the iterative nature of the ForgeNxt[®]2016 algorithm, we adopt the iterative HPTS method from [3] for the present work.

5.1.3.1 HPTS Approach [3]

The basis of this strategy is to use the incremental form of Prandtl-Reuss equations described in Section 5.1.2. This makes it possible to reuse the developments for the incremental (time-dependent) resolution of the same problem for the steady-state (time-independent) solver. This method proposes a two-step iterative strategy for the resolution of the elastoplastic constitutive model in the steady-state iterative algorithm, given as follows:

- Build streamlines from the velocities computed from the resolution of the mechanical problem
- Integrate the incremental Prandtl-Reuss constitutive model along the streamline

For the first step, the nodes and integration points the streamlines are built with a known velocity field. The nodes and the integration points are aligned with the streamlines as a result of the shape update (in the previous fixed-point iteration). It is to be reminded that [3] took advantage of the structured mesh for the alignment of the nodes/integration points along the streamline with the streamline equation (2.3.31b). In this method, the number of streamlines are equivalent to the number of surface elements on the input plane Γ_{in} . These streamlines connect consecutive integration points until they meet the output plane Γ_{out} of the control volume. This method assures that all the integration points have an associated streamline.

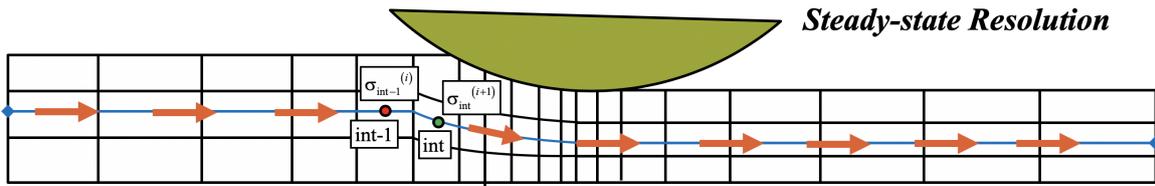


Figure 5.1.3: Steady-state computation of elastoplastic stress with structured mesh

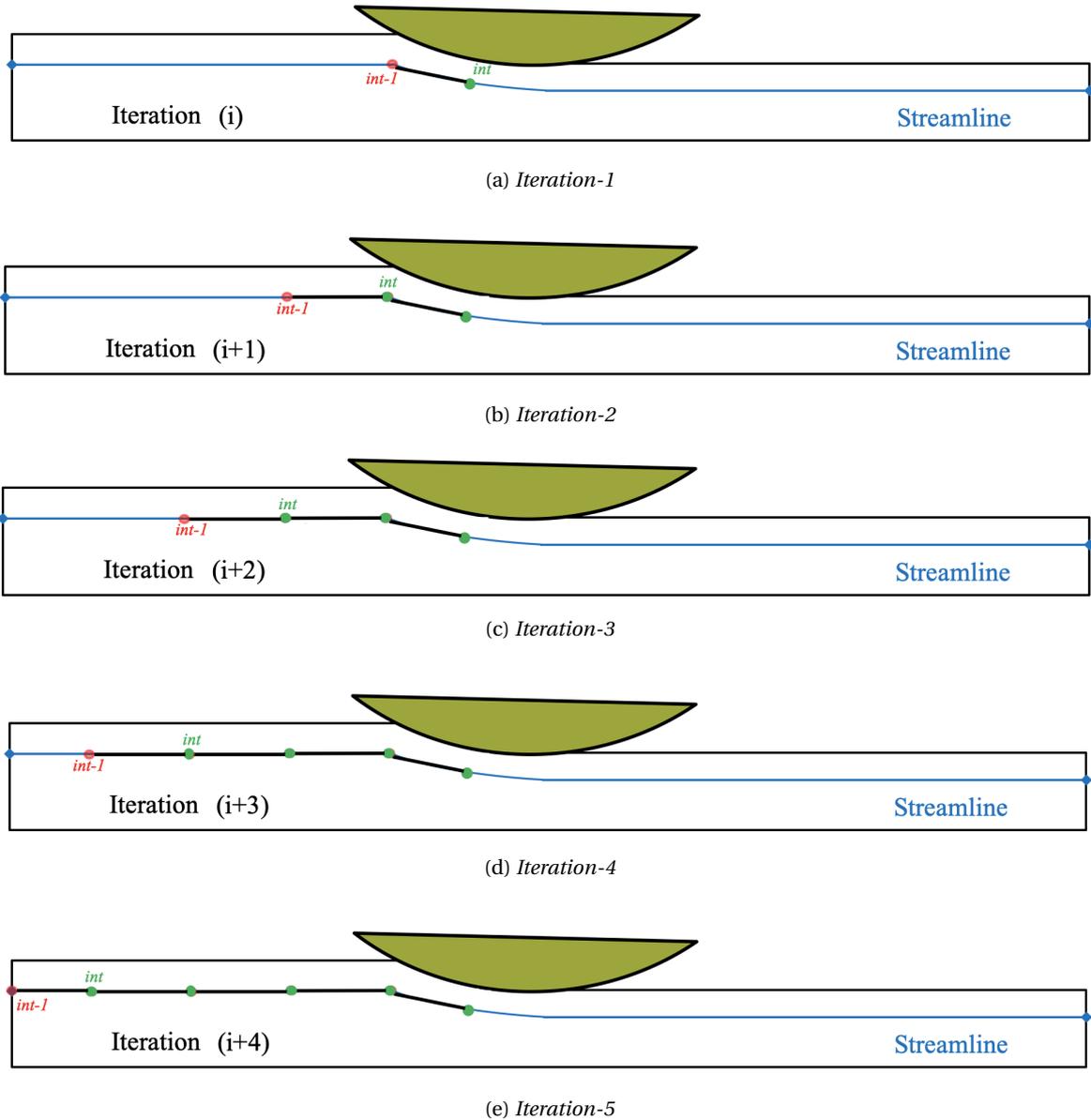


Figure 5.1.4: Iterative resolution of elastoplastic constitutive law for integration on a streamline

The integration point $int-1$ in each streamline is the representative of the previous state of the integration point int as shown in the Figure 5.1.3. Thus, the position of the integration point int in the steady-state formulation is given as following:

$$\vec{x}_{int} = \vec{x}_{int-1} + \vec{v}_{int} \Delta \tilde{t} = \vec{x}_{int-1} + \Delta \vec{x}_{int} \quad (5.1.25)$$

In the equation (5.1.25), $\Delta \tilde{t}$ is the pseudo-time-step and is described with the relation (2.3.12). This is a notional representation of time in time-independent equations. In addition, the time-

step is locally computed for each integration point, and is dependent on the distance between the two integration points. The time-step is thus heterogeneous in size and motivates the name of the method. The second step of the algorithm concerns the integration of field-variables on the streamlines. Firstly, the increment of state-variables ($\Delta\bar{\epsilon}$, $\Delta\mathbf{s}$) are computed by solving the Prandtl-Reuss equations (5.1.24) and (5.1.20) respectively. The respective pressure, deviatoric stress and strain increments find the state-variables at the integration point int with the following relations:

$$p_{int} = p_{int-1} + \Delta p_{int} \quad (5.1.26a)$$

$$\mathbf{s}_{int} = \mathbf{s}_{int-1} + \Delta \mathbf{s}_{int} \quad (5.1.26b)$$

$$\bar{\epsilon}_{int} = \bar{\epsilon}_{int-1} + \Delta \bar{\epsilon}_{int} \quad (5.1.26c)$$

In elastoplasticity, the equations in the Mechanical problem are strongly coupled in the stress and the velocity fields. The updated stress field must as well verify the equilibrium conditions, which yields a new velocity field. Thus a **Sub-iter loop** within the mechanical solver is necessary (as shown in Figure 2.3.2) for a converged solution. It is to be reminded here that the stress solution at an integration point is dependent on the stress solution at the integration point in its upwind. Hence, the stress field must converge/stabilize along the complete length of each streamline as shown in Figure 5.1.4. This requires multiple iterations equivalent to the number of integration points on each streamline.

5.2 New ForgeNxt[®] 2016 EVP resolution algorithm

The Pseudo-time-step-approach is interesting and easy to adapt for the ForgeNxt[®] 2016 solver with its iterative algorithm. In addition, the incremental Prandtl-Reuss equations in the ForgeNxt[®] inc solver can directly be reused for the resolution of the elastoplasticity problem with the steady-state solver. This strategy is proven to provide an excellent prediction and speed-up in comparison to the incremental solution. This approach is thus the first direction in which the ForgeNxt[®] 2016 EVP solver is being studied. The direct adoption of HPTS approach is not possible due to the requirements of structured mesh. In the unstructured mesh, the integration points $int-1$ and int are not necessarily aligned on the same streamline (see Figure 5.2.1). Hence, the advantage of structured mesh, with naturally aligned integration points along streamlines, is lost.

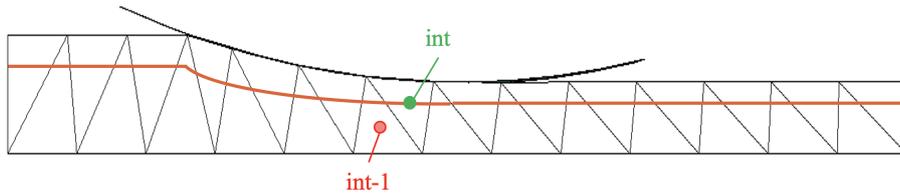


Figure 5.2.1: Problem with adoption of Lam3 algorithm with unstructured mesh

For the moment, the modification of the existing ForgeNxt[®] 2016 algorithm is envisaged to adopt the HPTS approach. This modification requires to build a streamline-framework for elastoplasticity computation in the unstructured mesh.

5.2.1 Building streamline in ForgeNxt[®] 2016 steady-state solver

For building the streamlines in the unstructured mesh, we could use different methods. The first option is the *long streamline method* (see Figure 5.2.2) in which all the integration points in the domain are projected backwards until they meet the input plane Γ_{in} . In this method, similar to [3] approach, every integration point has an associated streamline. The streamlines are built in an iterative loop, with consecutive projection and interpolation (of velocity) steps using a very

small pseudo-time-step $\Delta \tilde{t}$. After building the streamlines, the state-variables must be interpolated continuously on each streamline point from the elements and vice-versa after each **Sub-iter loop** iteration. Apart from the large computation requirements, this method also has a huge storage requirement for tracing the location and state-variables at every streamline.

$$\vec{x}_{int,I-1} = \vec{x}_{int,I} - \vec{v}_{int,I} \times \Delta \tilde{t} \quad (5.2.1)$$

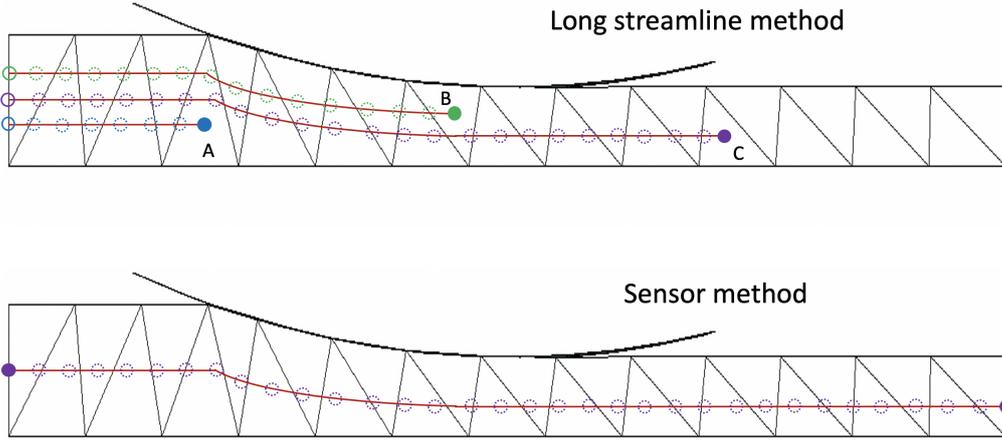


Figure 5.2.2: Comparison of different strategies for building streamlines in unstructured meshes

The second option, with comparatively lesser computation requirements is the *Lagrangian particle or sensor method*. It has been successfully used in the ALE formulation of the ring rolling process in ForgeNxt[®]inc. In this method, the nodal and element state-variables are transferred from one mesh to the particles before the meshing process. After the Lagrangian step and the ALE mesh update are performed, the state-variables are transferred from the particles to the new mesh. In this method, the Lagrangian points are defined at the input plane of the control volume before the problem resolution. A forward projection of these particles \mathbf{p} is done in the downwind direction with the equation (5.2.2) until the built streamlines encounter the output plane. This method reduces the enormous storage required with the *long streamline method* and hence reduces the computation time enormously. However, with this method, we cannot ensure that every integration point is associated with a streamline. Thus it is quite possible that these streamlines may miss high-stress gradient locations which could be critical for the elastoplasticity resolution.

$$\vec{x}_{p,I+1} = \vec{x}_{p,I} + \vec{v}_{p,I} \times \Delta \tilde{t} \quad (5.2.2)$$

A third method is proposed which is practical, ensures association of every integration point to a streamline, and has lesser storage requirements and not computationally intensive. This method defined as the *short streamline method* requires projection of each integration point *int* in the upwind direction (to the material flow). This projection is done only until a point *pre* with the equation (5.2.3) such that both the integration point and its projection are on the same streamline (as shown in Figure 5.2.3). However, it is to be noted that the streamline built is local in nature and thus, the cost and storage requirement of building streamlines is much lesser. In addition, the projection can be done without an iterative loop with a large pseudo-time-step $\Delta \tilde{t}$, thus reduces the streamline building time further.

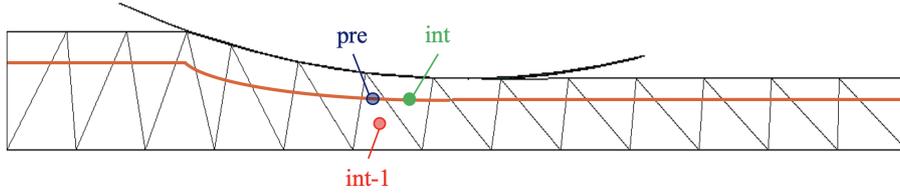


Figure 5.2.3: Methodology of adaption of Lam3 algorithm in ForgeNxt

$$\vec{x}_{pre} = \vec{x}_{int} - \vec{v}_{int} \Delta \tilde{t} \quad (5.2.3)$$

5.2.2 Constant pseudo-time-step for unstructured mesh

In the equation (5.2.3), $\Delta \tilde{t}$ denotes the pseudo-time-step for finding the projection point *pre*. A smaller $\Delta \tilde{t}$ may result in a projection point *pre* in the same element as *int*. This must be avoided as the interpolation of state-variables (P0 fields) on the *pre* is required for streamline integration. In HPTS approach, $\Delta \tilde{t}$ referred to a local, heterogeneous, definition of the time-step. This ensured that the integration points *int-1* and *int* are located in two consecutive adjoining elements and hence resulted in a streamline connecting input plane with the output plane. For the unstructured mesh, it is not necessary to use a constant pseudo-time-step as *pre* can be located anywhere in the upwind, except for in the same element. The constant pseudo-time-step could be defined with either of the following criteria (5.2.4). However, the choice between the two proposed criteria is trivial as both of them assure that the *pre* is not in the same element as *int*.

$$\Delta t_{\min} = \frac{\max_e h_e}{\min_e |\vec{v}|} \quad \text{or} \quad \max_e \frac{h_e}{|\vec{v}|} \quad (5.2.4)$$

It must be noted that as a consequence of a constant minimum pseudo-time step, the element containing *pre* may not be contiguous with the element containing *int* as shown in the Figure 5.2.4a in the flow downwind to the roll-die. Even with homogeneous (similar size) meshes, the streamlines are never of the same length but depend on the local velocity, with longer streamlines post rolling and shorter streamlines in the upwind of the tool. The differential length is not a problem as the gradients are generally higher only under the tool. Using equations (5.2.3) and (5.2.4), we can thus build the local streamlines in the unstructured mesh.

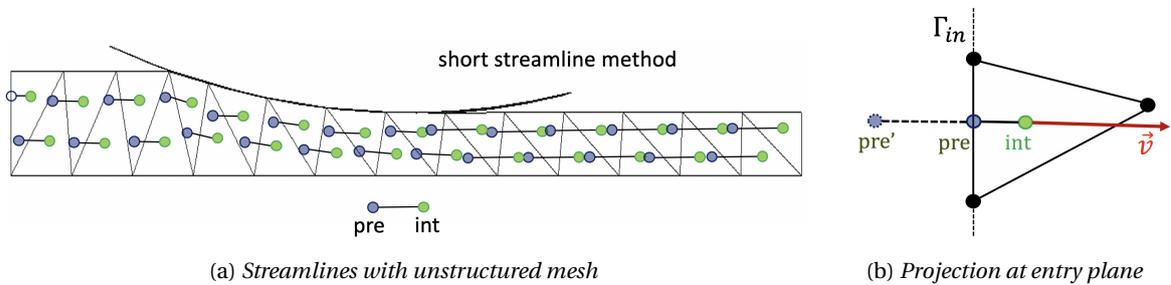


Figure 5.2.4: Building streamlines in a domain with unstructured mesh

We must also consider some special scenarios while building the local streamlines. Such scenarios occur when the *pre* are located outside the domain boundary. For such cases, the point *pre* is projected on the domain boundary, for example on the entry plane (see Figure 5.2.4b). The initial (residual) stress/strain is applied at the entry plane nodes and hence is applied to the projection points projected on this plane.

5.2.3 Basic ForgeNxt[®]2016 EVP algorithm

The basic iterative algorithm for the resolution of elastoplasticity is shown in the Figure 5.2.5 and is similar to the existing ForgeNxt[®]2016 algorithm for viscoplastic materials with additional computations in the Step-1 to solve the EVP problem. First, the streamlines are built in the domain with the known/assumed velocity field followed by the subsequent streamline integration that involves the following steps. These steps constitute the part of the Sub-iter loop.

1. Interpolate state-variables at *pre* to model the previous known state (see Section 5.2.4)
2. Newton-Raphson resolution of the Stokes problem for \vec{v} - p fields (see Section 5.2.5)
3. Prandtl-Reuss resolution for elastoplastic state-variables (see Section 5.1.2) and update on the streamlines
4. Test convergence of stress field, exit if yes, go to 1 if no

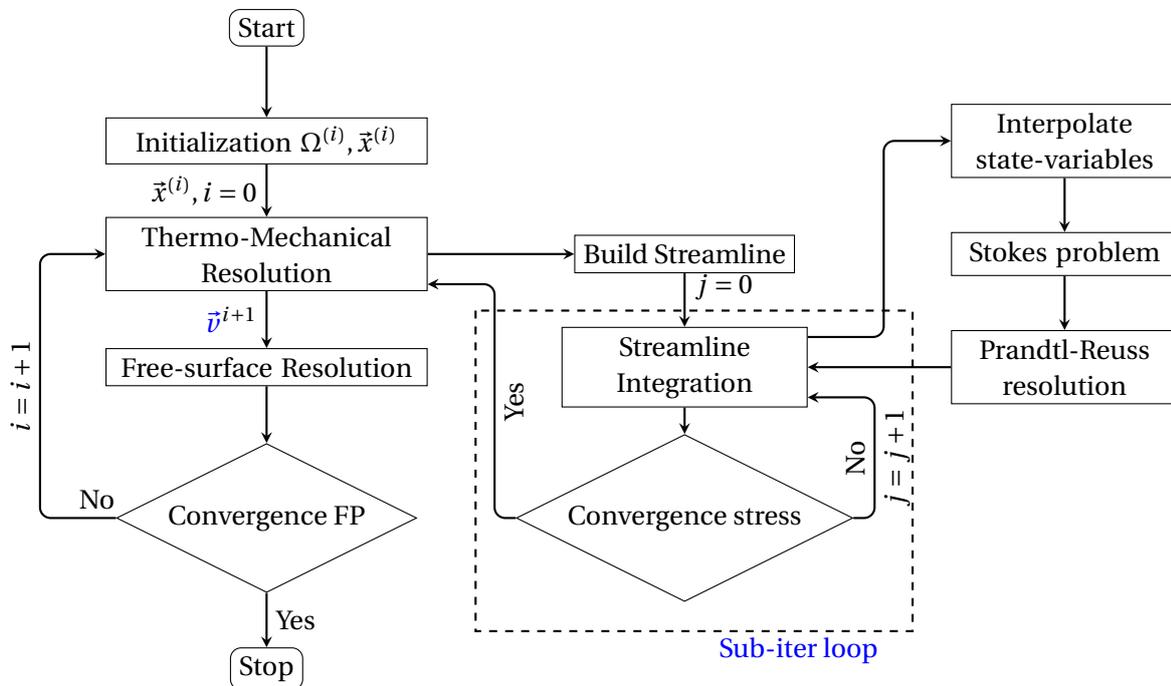


Figure 5.2.5: Proposed basic algorithm for ForgeNxt[®]2016 EVP adapted from [3]

5.2.4 Detailed Sub-iter loop for the elasto-viscoplastic resolution in ForgeNxt[®]2016 EVP

As mentioned in Section 5.3.3, the choice of pseudo-time-step ensures that the point *pre* is in a different element than *int*. For the streamline integration of the constitutive model, the state-variables must first be precisely known at *pre*. The state variables, such as stress and equivalent strain, are P0 fields (constant and discontinuous per element), computed and stored at the integration points (one for each element for the present formulation). The easiest method to remap these variables is a *direct P0 transfer*. The value of the P0 variable at *pre* is approximated as the value of the nearest integration point in local element (to which *pre* belongs). With this approach, the mapped stress tensor automatically satisfies the balance equations. This technique is fast and works well with refined mesh, however, the error becomes large with large mesh sizes with its magnitude proportional to the gradient of remapped values for the mesh size. Such operations (transfer of P0 variables from one mesh to other) are common with multimesh [5] or ALE methods, and often inaccuracies are reported. In the present problem, the transfer of P0 variables is done from mesh with *int* to the mesh with *pre*. It is to be noted that depending on whether the point *pre*

lies inside the element or on the shared boundary between two elements, the state-variable could have unique or multiple values respectively. Moreover, in general practice, in the steady-state resolution, finer meshes are necessary only to capture large gradients, which are generally under the tool contact. Hence, a coarser mesh is enough away from the contact. In such scenarios, it is logical to interpolate the P0 state-variable field at *pre* from a smoother nodal solution. However, the smoothing process is highly diffusive and special care must be taken to minimize the diffusion. This idea of mapping solutions from smoother solution are also necessary while mapping results from one mesh to other, generally with bimesh and ALE methods in Fg3[®]. Hence, there exist some tools in ForgeNxt to keep this transfer with minimum diffusion.

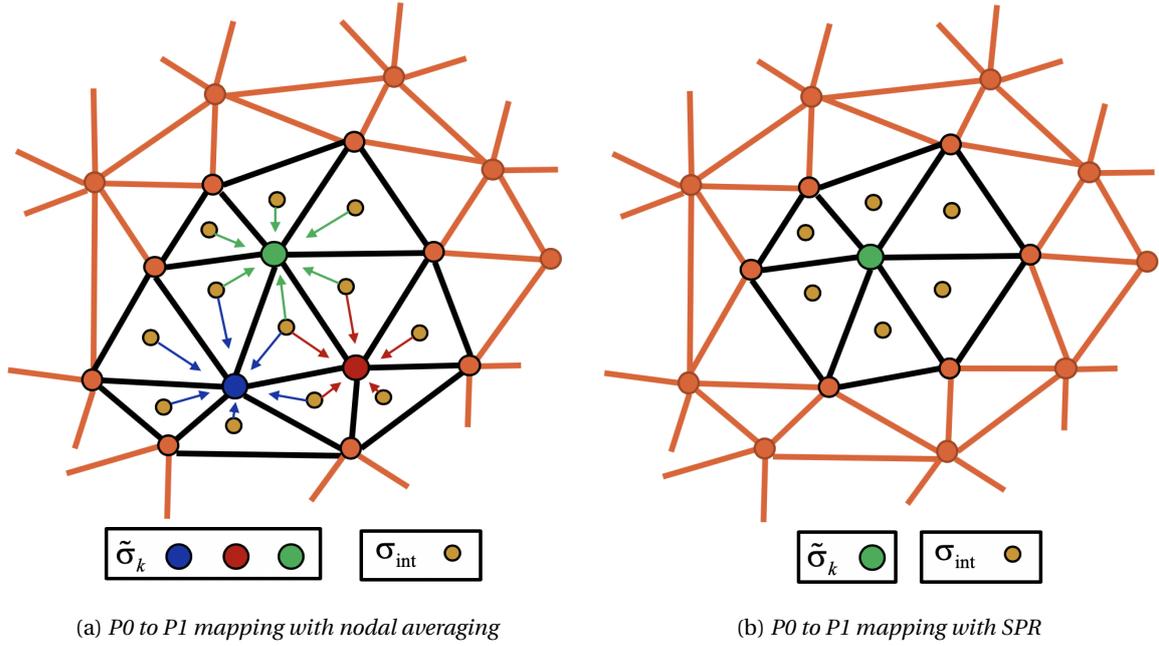


Figure 5.2.6: P0 to P1 mapping of the state-variables

Smoothing of state-variables: mapping from P0 to P1 Two possibilities are envisaged for the smoothing of the P0 state-variables to P1. The first method is *nodal averaging*, in which the P0 solution σ_{int} is known at the integration points of the elements, thus a weighted average of the element solution is done at the nodes to find the P1 field $\tilde{\sigma}$ (see Figure 5.2.6a). However, this method tends to be quite diffusive, and it must be limited so that the equilibrium equation must not be strongly violated.

On the other hand, it is also possible to use a *SPR method*, which was first proposed by [6] for error estimation in adaptive meshing, and can be used for mapping P0 field at the nodes. This method was used for remapping the state-variables in the new mesh (from adaptive remeshing) in ALE by [2] and proven to be less diffusive. In this technique, we aim to find the recovered solution (see Figure 5.2.6b) at the nodes $\tilde{\sigma}$ with the following equation:

$$\tilde{\sigma}_k(\vec{x}) = \mathbf{P}^k(\vec{x}) \mathbf{a}_k = a_k^0 + a_k^1(x - x_k) + a_k^2(y - y_k) + a_k^3(z - z_k) \quad (5.2.5)$$

In the above equation \mathbf{a}_k is the vector of coefficients (constant for linear elements), \mathbf{P}_k is the polynomial basis. The expression (5.2.5) is written with the first-order expansion. The four unknown coefficients \mathbf{a}_k can be computed by minimizing the following least-square expression.

$$\Pi(\mathbf{a}_k) = \frac{1}{2} \sum_{int=1}^{nint} (\tilde{\sigma}_k(\vec{x}_{int}) - \sigma_{int})^2; \quad \forall n = 0, 3 \quad \frac{\partial \Pi(\mathbf{a}_k)}{\partial a_k^n} = 0 \quad (5.2.6)$$

Here $nint$ refers to the total number of integration points in the patch. In the event of an insufficient number of neighbors for node k to find the coefficient vector \mathbf{a}_k , the patch is extended to the second-order neighbors and additional stabilization terms are added to the expression.

Comparison of P0-P1 mapping strategies For the assessment of the two methods presented for the P0-P1 mapping in terms of the best technique, we propose the following strategy:

1. Initialize a discretized geometry model with a known P1 field \tilde{q}
2. Interpolate field \tilde{q} at the integration points to find the P0 field \bar{q}
3. Use *Nodal Averaging* method to find the P1 solution \tilde{q}_{NA} from the P0 field \bar{q}
4. Use *Superconvergent Patch Recovery* method to find the P1 solution \tilde{q}_{SPR} from the P0 field \bar{q}
5. Find and compare the respective P1 field errors with the two methods $\epsilon_{NA} = |\tilde{q}_{NA} - \tilde{q}|$ and $\epsilon_{SPR} = |\tilde{q}_{SPR} - \tilde{q}|$

We test the above-mentioned strategy with the equivalent strain, hence $q = \epsilon$. The starting smooth equivalent strain ($\tilde{\epsilon}$) which is a P1 field and found from the resolution of convection equation (3.2.42). The model and the converged strain field from *VP-Test-Case-01* in Section 3.5.1 is used for the current test. The respective $\tilde{\epsilon}$, $\bar{\epsilon}$, $\tilde{\epsilon}_{NA}$ and $\tilde{\epsilon}_{SPR}$ fields are shown in the Figure 5.2.7. The P1 field obtained from *SPR* looks much similar to the field obtained with the nodal averaging method. It is only with the error between the starting and mapped P1 solutions in Figure 5.2.8 that we see the difference clearly.

The error is much less with the solution obtained with *SPR* method in comparison to the nodal averaging both outside and inside the domain. Hence, it can be concluded that the *SPR* method is much less diffusive in comparison to the nodal averaging method and hence would be used for the interpolation of the state-variables at projection points *pre*.

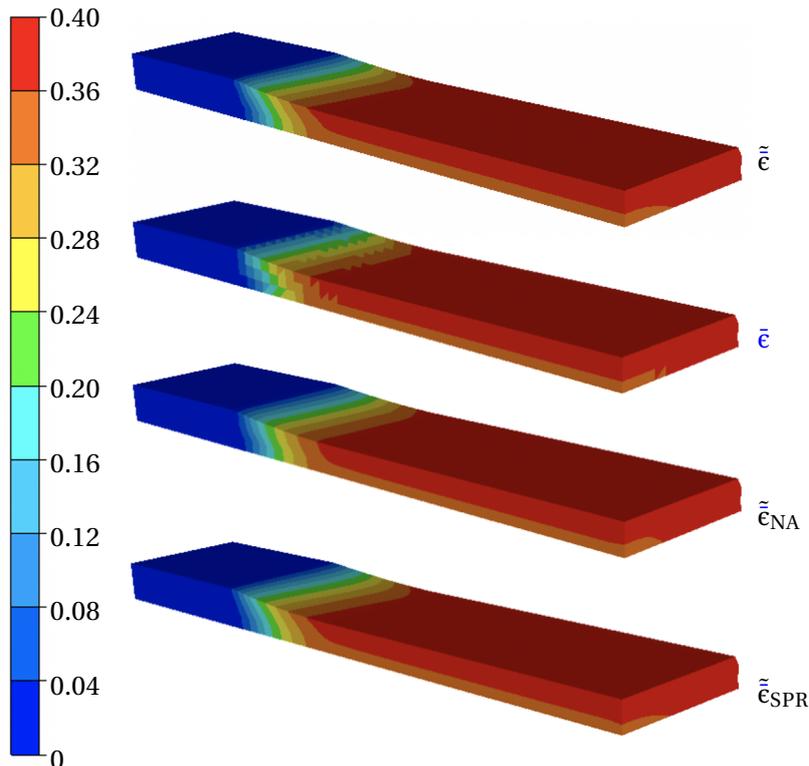


Figure 5.2.7: Comparison of P0-P1 field mapping methods- P1 solutions

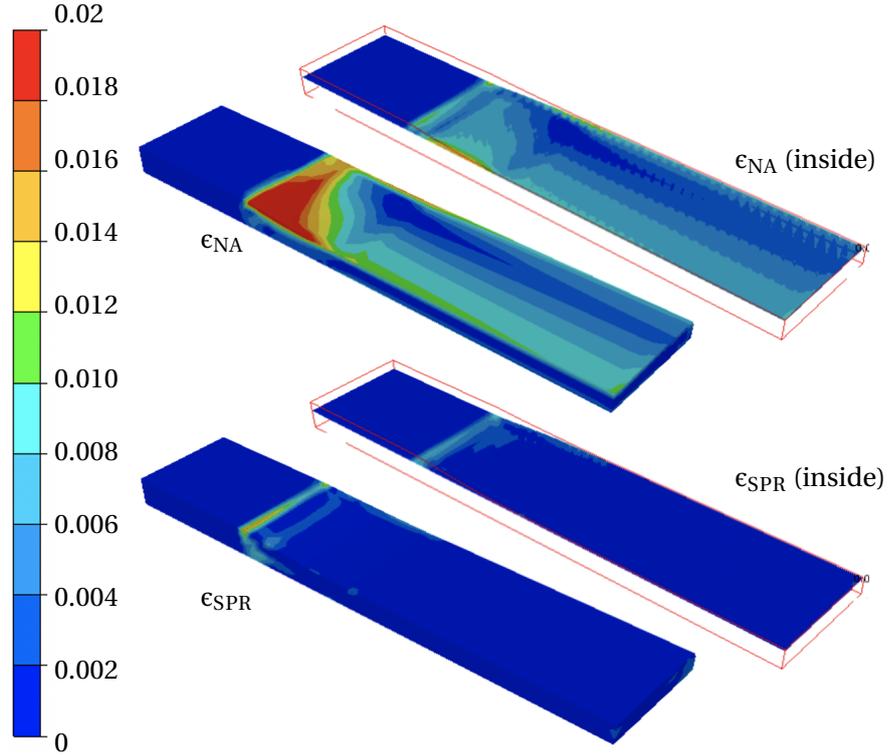
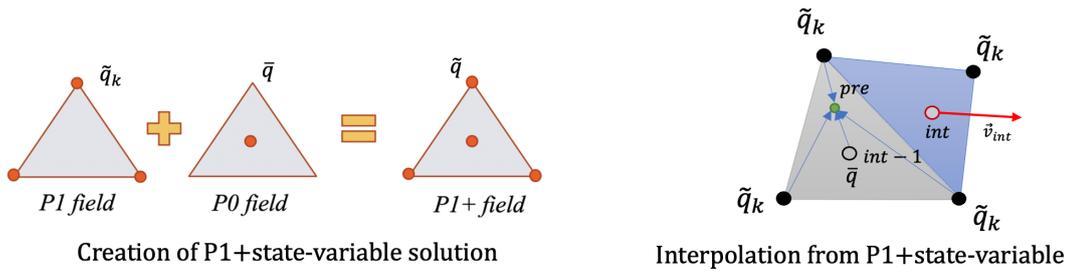


Figure 5.2.8: Comparison of P0-P1 field mapping methods- P1 error field

P1+ Interpolation of state-variables at *pre* The next step is to interpolate the state-variable fields at the projection points *pre* from the smooth solution of the state-variables. It is to be highlighted here that an enriched state-variable field is built using the solution known at the nodes (smoothed solution) and the integration points of the mesh as shown in the Figure 5.2.9. To restrict the interpolation diffusion to a minimum, it is proposed to use this enriched solution for the interpolation.


 Figure 5.2.9: Building P1+ field from P0 field known at integration point (centroid of element) and P1 field known at the nodes and interpolation at *pre*

With the prescribed definition of the P1+ field $\tilde{q} \oplus \bar{q}$, we can interpolate the state-variable at any coordinate \vec{x} in the discretized domain Ω_{xh} with the following relation. In this equation, k is the nodal index, N_k is the piecewise linear Galerkin shape function, and N_e is the bubble function. The bubble function is also used for the velocity interpolation in Section 3.2.3.4. For our application, a piecewise linear function with value 1 at the centroid of the element and 0 at the boundary in the subspace of the tetrahedral element is chosen. The derivation, of N_e and the distance function to locate the sub-tetrahedron (subspace) the coordinate \vec{x} belongs to, is presented in Appendix A.1.

$$\tilde{q}(\vec{x}) = \sum_{k=1,4} N_k(\vec{x}) \tilde{q}_k(\vec{x}) + N_e(\vec{x}) \left(\bar{q} - \sum_{k=1,4} N_k(int) \tilde{q}_k \right) \quad (5.2.7)$$

The equation can be used to interpolate the state-variables at the projection point *pre*. Hence, the

state-variables including the deviatoric stress \mathbf{s}_{pre} , and equivalent strain $\bar{\epsilon}_{pre}$ are interpolated from the following relations using the respective P1+ fields.

$$\mathbf{s}_{pre} = \tilde{\mathbf{s}}(\tilde{\mathbf{x}}_{pre}) \quad (5.2.8a)$$

$$\bar{\epsilon}_{pre} = \tilde{\bar{\epsilon}}(\tilde{\mathbf{x}}_{pre}) \quad (5.2.8b)$$

It can be noted that if *pre* coincides with the local integration point *int-1* of the element in which it is located, the above description yields the field value at the integration point as follows:

$$\tilde{q}(\mathit{int-1}) = \sum_{k=1,4} \mathbf{N}_k(\mathit{int}) \tilde{q}_k(\mathit{int-1}) + \underbrace{\mathbf{N}_e(\mathit{int-1})}_{=1} \left(\bar{q} - \sum_{k=1,4} \mathbf{N}_e(\mathit{int-1}) \tilde{q}_k \right) = \bar{q} \quad (5.2.9)$$

5.2.5 Newton-Raphson resolution of Stokes Problem

Once the streamlines have been built with the state variables interpolated at the projection points *pre*, the Stokes problem is resolved to compute the unknown velocity \vec{v} and pressure p fields. In the said problem, the linear momentum conservation (3.2.1a) still continues to be valid, however, the incompressibility condition (3.2.1b) is no more applicable. Instead, we use the following condition for the elastic compressibility:

$$\vec{\nabla} \cdot (\vec{v}) = \frac{-\dot{p}}{K_b} \quad \text{on} \quad \Omega_x \quad \text{with} \quad K_b = \lambda^e + \frac{2\mu^e}{3} \quad (5.2.10)$$

In the above equation, the material is incompressible when the Bulk modulus $K_b \rightarrow \infty$, and in any practical application, the formulation results in finite compressibility. So, the weak form of the mechanical problem in equation (3.2.20) is replaced by the following: Find $(\vec{v}, p) \in \mathbb{V}^v \times \mathbb{V}^p$ such that, $\forall (\vec{v}^*, p^*) \in \mathbb{V}_0^v \times \mathbb{V}^p$

$$\int_{\Omega_x} \mathbf{s}(\vec{v}) : \dot{\boldsymbol{\epsilon}}(\vec{v}^*) d\omega - \int_{\Omega_x} p \operatorname{div}(\vec{v}^*) d\omega - \int_{\Gamma_c} \vec{\tau}_f(\vec{v}) \cdot \vec{v}^* ds - \int_{\Gamma_c} \sigma_n \vec{n} \cdot \vec{v}^* ds = 0 \quad (5.2.11a)$$

$$- \int_{\Omega_x} p^* \left(\vec{\nabla} \cdot (\vec{v}) + \frac{\dot{p}}{K_b} \right) d\omega = 0 \quad (5.2.11b)$$

In the incremental (time-dependent) formulation, the pressure-rate \dot{p} is defined with equation (5.2.12). Here p_t is the pressure field in the previous time step and Δt being the time step.

$$\dot{p}_h = \frac{p^{t+\Delta t} - p^t}{\Delta t} \quad (5.2.12)$$

In the steady-state context, the pressure field (P1) at the previous pseudo-time-step cannot be computed with the existing streamline framework for modeling the P0 variables at previous pseudo-time-step. Hence, a new framework is devised to model the pressure field at the previous time step.

1. Build P1 streamlines for each node $k \in \Omega_h$ and its projection \tilde{pre} computed from equation (5.2.13). In this equation, the pseudo-time-step $\Delta \tilde{t}_{\min}$ is the same as used for building the streamlines for the integration points in equation (5.2.2)

$$\forall k \in \Omega_h, \quad \tilde{\mathbf{x}}_{\tilde{pre}} = \tilde{\mathbf{x}}_k - \vec{v}_k \Delta \tilde{t}_{\min} \quad (5.2.13)$$

2. The pressure is computed at the node projection \tilde{pre} from the following interpolation

$$p_{\tilde{pre}} = \sum_{k=1,4} \mathbf{N}_k(\tilde{\mathbf{x}}_{\tilde{pre}}) p_k \quad (5.2.14)$$

After the FE discretization with the P1+/P1 mini elements (see equation (3.2.28)), we can describe the discretized weak form of the mechanical problem in the following manner. Find $(\vec{v}_h^l, \vec{v}_h^b, p_h) \in \mathbb{V}_h^{v,l} \times \mathbb{V}_h^{v,b} \times \mathbb{V}_h^p$, with $m \in [1, 4]$, $\mu^e \in [1, 3]$ such that:

$$R_{\mu m}^l(\vec{v}_h^l, \vec{v}_h^b, p_h) = \int_{\Omega_h} \mathbf{s}_h(\vec{v}_h^l) : \mathbf{B}_{\mu m}^l d\omega_h - \int_{\Omega_h} p_h \operatorname{div}(\mathbf{B}_{\mu m}^l) d\omega_h - \dots = 0 \quad (5.2.15a)$$

$$R_{\mu}^b(\vec{v}_h^l, \vec{v}_h^b, p_h) = \int_{\Omega_h} \mathbf{s}_h(\vec{v}_h^b) : \mathbf{B}_{\mu}^b d\omega_h - \int_{\Omega_h} p_h \operatorname{div}(\mathbf{B}_{\mu}^b) d\omega_h = 0 \quad (5.2.15b)$$

$$R_m^p(\vec{v}_h^l, \vec{v}_h^b, p_h) = - \int_{\Omega_h} N_m \left(\operatorname{div}(\vec{v}_h^l + \vec{v}_h^b) + \frac{p_h - p_h^{\bar{p}e}}{K_b \Delta \tilde{\mathbf{t}}} \right) d\omega_h = 0 \quad (5.2.15c)$$

For the stiffness matrix computation, the derivative of the deviatoric stress $\Delta \mathbf{s}$ with the velocity \vec{v} must be described. The chain rule necessitates the computation of the derivative of the deviatoric stress with the deviatoric strain $\Delta \mathbf{e}$ given the relation (5.2.16a). The deviatoric strain increment is the deviatoric part of the incremental strain $\Delta \mathbf{e}$, which is computed from the velocity field with the equation (5.1.8b).

$$\frac{\partial \Delta \mathbf{s}}{\partial \vec{v}} = \frac{\partial \Delta \mathbf{s}}{\partial \Delta \mathbf{e}} \frac{\partial \Delta \mathbf{e}}{\partial \Delta \boldsymbol{\epsilon}} \frac{\partial \Delta \boldsymbol{\epsilon}}{\partial \vec{v}} \quad (5.2.16a)$$

$$\frac{\partial \Delta \mathbf{s}}{\partial \Delta \mathbf{e}} = \frac{2\mu^e}{1 + \frac{3\mu^e \Delta \bar{\epsilon}}{\sigma_y(\bar{\epsilon} + \Delta \bar{\epsilon})}} \left(\mathbf{Id}_4 - \frac{1 - \frac{\Delta \bar{\epsilon}}{\sigma_y(\bar{\epsilon} + \Delta \bar{\epsilon})} \frac{d\sigma_y}{d\bar{\epsilon}}}{1 + \frac{1}{3\mu^e} \frac{d\sigma_y}{d\bar{\epsilon}}} \frac{(\mathbf{s} + 2\mu^e \Delta \mathbf{e}) \times (\mathbf{s} + 2\mu^e \Delta \mathbf{e})}{(\mathbf{s} + 2\mu^e \Delta \mathbf{e}) : (\mathbf{s} + 2\mu^e \Delta \mathbf{e})} \right) \quad (5.2.16b)$$

In addition, the H^{pp} in the system (3.2.31) is non-zero for elasto-viscoplasticity and given as following:

$$H_{mk}^{pp} = - \int_{\Omega_{xh}} N_m \frac{N_k}{K_b \Delta \tilde{\mathbf{t}}} d\omega_h \quad (5.2.17)$$

The orthogonality of the linear and bubble shape functions in the P1+ interpolation, is still valid. So, the terms $H_k^{lb,(i)}$ and $H_k^{bl,(i)}$ are still zero.

The new velocity field computed from the Newton-Raphson resolution is used to update the state-variables from the Prandtl-Reuss equations described in Section 5.1.2. As the new state-variables $\mathbf{s}, \bar{\epsilon}$ are computed at the integration points *int*, the same must be used to compute the updated velocity field, thanks to the strong coupling between the velocity and the stress field. As mentioned before, these steps are repeated in the *Sub-iter loop* until the stabilization of the stress field. *In this present algorithm, the streamlines are not updated as soon as the velocity field is updated in order to remain consistent with A. Hacquin's approach in which the streamlines are updated during geometry computation with the nodal repositioning.*

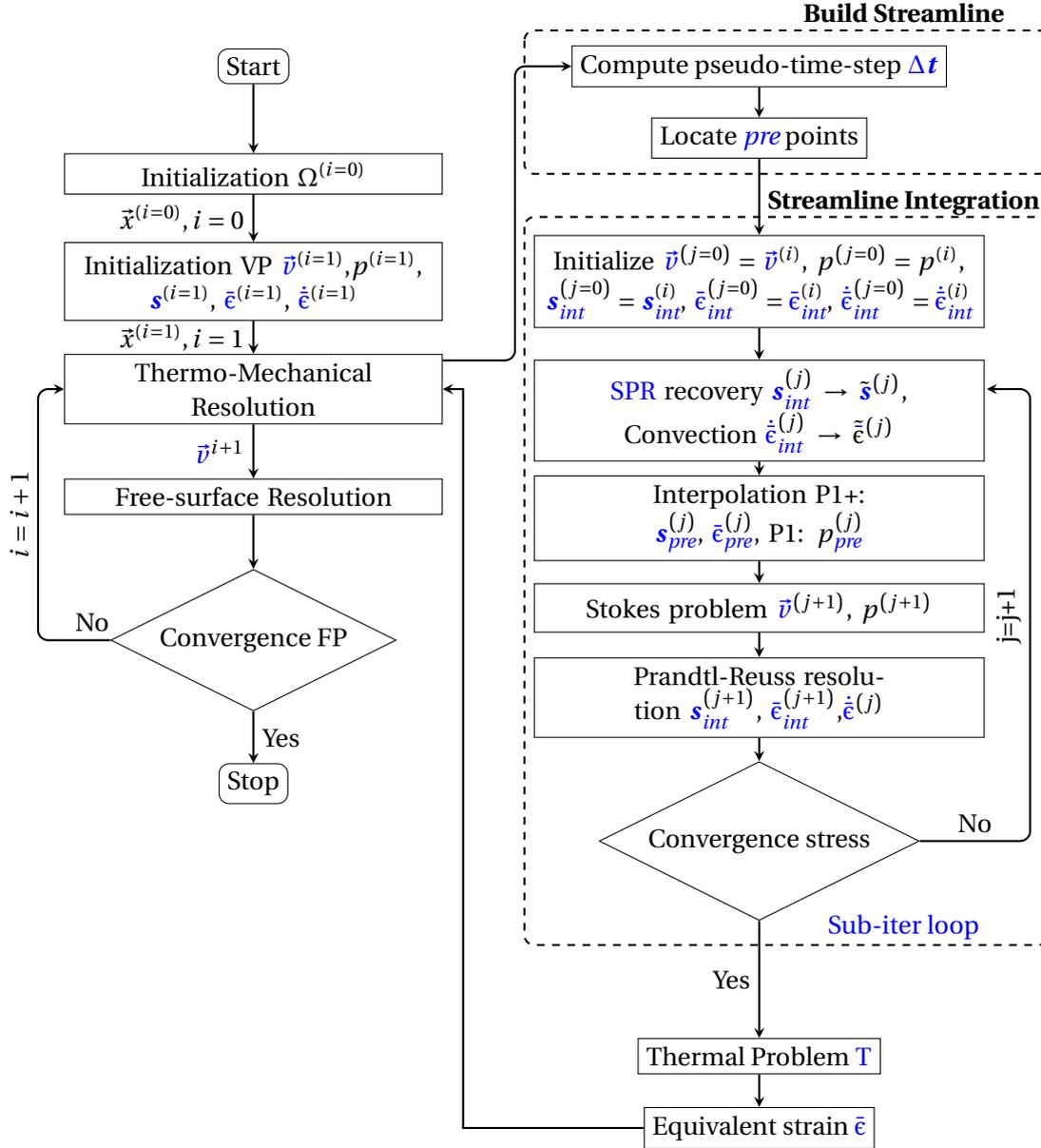
5.2.6 Detailed ForgeNxt[®]2016 EVP Algorithm-1

 Figure 5.2.10: Detailed ForgeNxt[®]2016 EVP Algorithm-1 with Sub-iter loop

Figure 5.2.5 briefly listed the steps involved in the resolution algorithm for the steady-state elastoplasticity. The present section is dedicated to organize the steps presented in the previous sections necessary for elastoplasticity resolution in the proposed algorithm. For initialization, the streamline building is not possible as the velocity field is unknown. Hence, the first iteration is computed with the steady-state algorithm with a viscoplastic material model, in which no streamline building required. Once initialized, the pseudo-time-step $\Delta \tilde{t}$ is computed followed by the computation of projection points *pre* in the mesh as explained in Section 5.2.1. This initial viscoplastic solution for velocity \bar{v} , pressure p , deviatoric stress s and equivalent strain $\bar{\epsilon}$ fields is necessary for the initialization of the elasto-viscoplastic resolution with the proposed algorithm. Next the steps of the **Sub-iter loop** are recalled for streamline integration of the state-variables. Inside this loop, the iteration is represented with the index j . Firstly, the state-variables are initialized followed by the smoothing of the deviatoric stress with the SPR method. The smoothed equivalent strain is computed with the resolution of the convection equation (3.2.42). Using the P1+ state-variables fields, and P1 pressure field, they are interpolated at the *pre*. The initialized fields (at *pre*) are used

to compute the elasto-viscoplastic state-variables from Prandtl-Reuss equations. The steps in the **Sub-iter loop** are repeated until the convergence of the stress field with the criterion explained below in Section 5.2.7. On convergence of the mechanical problem with the **Sub-iter loop**, the updated velocity field is used for the free-surface resolution. It is also to be pointed out that the new velocity field computed inside the **Sub-iter loop** is not used to update the streamlines as a choice adopted from the original [3] approach. The convergence of the streamlines is rather assured from the converged velocity and shape fields in consecutive fixed-point iterations.

5.2.7 Convergence criterion/a for the **Sub-iter loop**

The convergence of the **Sub-iter loop** is realized when the deviatoric stress is stable in the complete domain. This is ensured with the infinite error norm of the deviatoric stress, which tends to bring the maximum difference between the deviatoric stress (all 6 components) in an element between two **Sub-iter loop** iterations below a defined criterion.

$$d \in \{1, 6\} \quad \frac{\max_{int} |s_d^{j+1}(int) - s_d^j(int)|}{\max_{int} |s_d^{j+1}(int)| - \min_{int} |s_d^{j+1}(int)|} \leq \epsilon_{si} \quad (5.2.18)$$

The criterion chosen here is $\epsilon_{si} = 5\%$ which ensures the convergence of the stress field in the complete domain. [3, 4] highlighted that the stress update with such a criterion is communicated one section per iteration in the structured mesh. Hence the number of iterations required for the complete stability is equivalent to the number of sections. With the unstructured mesh, where we do not have well-prescribed sections, the number of iterations for the convergence could be estimated with the ratio of the total length of the workpiece with the minimum communication distance.

5.3 Validation of **ForgeNxt[®]2016 EVP Algorithm-1** with a simple cold rolling problem

The proposed **ForgeNxt[®]2016 EVP Algorithm-1** in Section 5.2.6 is evaluated for its performance with some test-cases using elasto-viscoplastic constitutive material model. The algorithm is tested in an order of increasing complexity. The convergence of the **Sub-iter loop** has also been studied with the chosen criterion to see if a stable stress field is obtained. The results from these test-cases are validated with the incremental resolution of the same problem (and same mesh size) in **ForgeNxt[®]inc** until the steady-state is reached.

5.3.1 **EVP-Test-Case-01** model details

Firstly, a simple rolling test case is tested with the proposed **ForgeNxt[®]2016 EVP** algorithm. The model geometry is shown in Figure 5.3.1. At the input plane Γ_{in} , the sheet measures 10 mm x 1 mm and at the output plane it measures 9 mm x 1 mm. $2(x, y)$ symmetric planes are applied to constraint the lateral flow of the material. This problem is tested with different models (structured/unstructured mesh, with homogeneous/heterogeneous mesh-size). The viscoplastic initialization (Figure 5.2.10) is performed either with Hensel-Spittel or Norton-Hoff constitutive models.

Mesh models

Three different mesh models have been tested for the problem (see Figure 5.3.1). First, a structured mesh of 0.5mm size throughout the domain. The second model consists of unstructured mesh with size 0.5 mm. The third model consists of non-homogeneous mesh in the workpiece domain with 0.5 mm below the tool and 1.0 mm away from the tool.

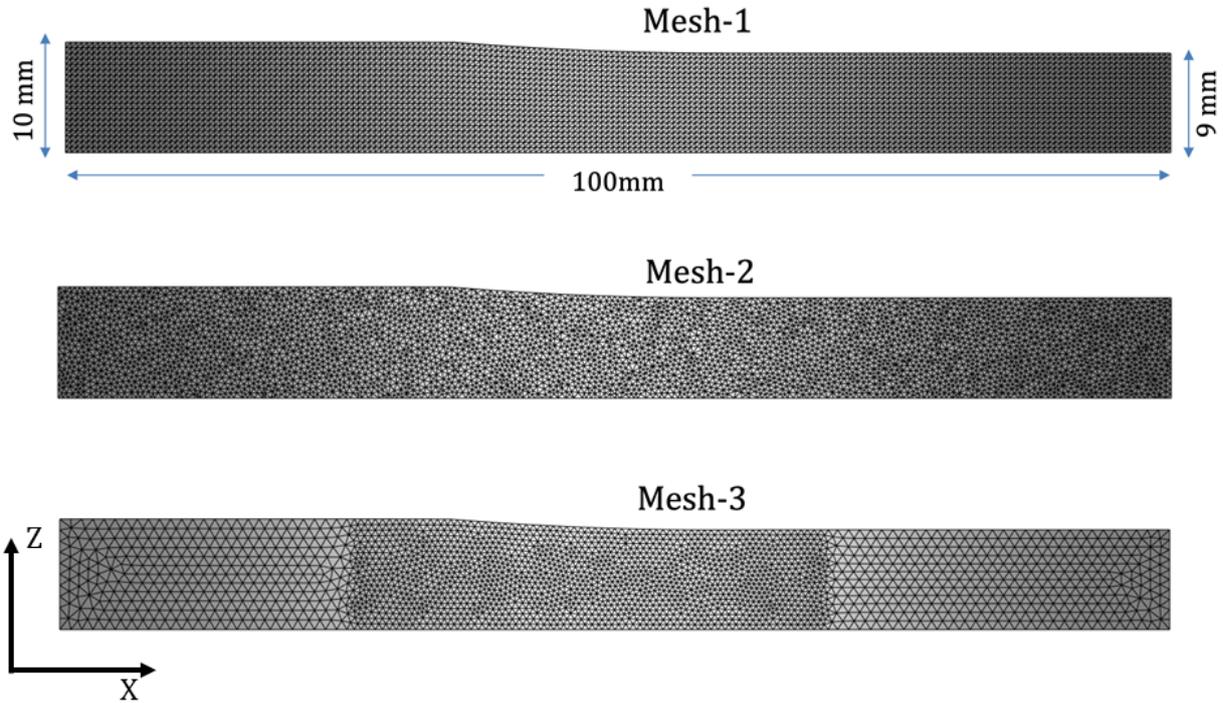


Figure 5.3.1: Mesh models tested for the pseudo-2D cold rolling (EVP) test-case

Material constitutive model

The Hensel-Spittel material model is used to describe the elasto-viscoplastic behavior. This model is described generally in the following form. For modeling elasticity, the Young's modulus E and Poisson's ratio as ν are also specified.

$$\bar{\sigma} = A e^{m_1 T} \bar{\epsilon}^{m_2} \dot{\bar{\epsilon}}^{m_3} e^{m_4 / \bar{\epsilon}} \quad (5.3.1)$$

On comparison with the Norton-Hoff flow equation (3.2.7), we can see the similarities between the two models; with Hensel-Spittel it is possible to model much complex material behavior. For the present case we assume an elasto-viscoplastic fictitious material with specifications of the coefficient A , m_2 and m_3 respectively shown in Table 5.3.1. The other coefficients are assumed to be null. However, we do not assume any strain hardening for simplicity.

Type	Parameters
Hensel-Spittel EVP	$E = 200\text{GPa}$, $\nu = 0.3$, $A = 1000\text{MPa}$, $m_2 = 0.2$, $m_3 = 0.05$, $m_4 = 0.05$

Table 5.3.1: Hensel-Spittel properties used for the EVP resolution of [EVP-Test-Case-01](#)

5.3.2 Initialization with viscoplastic solution

An initialized solution is necessary for the resolution of the problem with the steady-state EVP algorithm. So, first we solve the problem with the viscoplastic material model using the [ForgeNxt® steady-state solver with implicit contact constraint \(Steady-State Implicit\)](#) to initialize the velocity, pressure, stress, and equivalent strain fields. The starting geometry for elasto-viscoplastic resolution is the converged geometry from viscoplastic resolution. Thus only a few iterations are required for the convergence of the geometry with the elasto-viscoplastic material model. The Norton-Hoff model and Hensel-Spittel models with following properties are chosen for the initialization of the solution.

Type	Parameters
Hensel-Spittel VP	$A = 1000, m_2 = 0.2, m_3 = 0.05, \text{ other coeffs}=0$
Norton Hoff VP	$K = 220, m = 0.5$

Table 5.3.2: Different models and properties used for the VP initialization

5.3.3 Different EVP-Test-Case-01 model variants tested

The details of the models tested for the present test-case are shown in Table 5.3.3. For example, the **EVP-Test-Case-01-a** model uses the structured Mesh-1 (see Figure 5.3.1) and is initialized with the Hensel-Spittel viscoplastic model and so on.

Model Name	Mesh	Viscoplastic model
EVP-Test-Case-01-a	Mesh-1 (structured, 0.5mm everywhere)	Hensel-Spittel
EVP-Test-Case-01-b	Mesh-2 (unstructured, 0.5mm everywhere)	Hensel-Spittel
EVP-Test-Case-01-c	Mesh-3 (unstructured, 0.5mm near tool, 1mm away)	Hensel-Spittel
EVP-Test-Case-01-d	Mesh-3 (unstructured, 0.5mm near tool, 1mm away)	Norton-Hoff
Incremental	Mesh-2 (unstructured, 0.5mm everywhere)	-

Table 5.3.3: Different **EVP-Test-Case-01** variants tested

5.3.4 Comparison of EVP-Test-Case-01 results

Firstly, we compare the convergence of the deviatoric stress inside the **Sub-iter loop**. The residual of the different components of the deviatoric stress, as presented in the equation (5.2.18), is plotted for the present test-case in Figure 5.3.2. For each component, the residual convergence is compared for the four test-models (in Table 5.3.2). The maximum number of iterations chosen for the **Sub-iter loop** are 30 (within each fixed point iteration). Since multiple projections and re-projections of the state-variables is involved, the diffusive effect must be curtailed by limiting the maximum number of iterations. In each plot, the abscissa is the global step number which corresponds to the total number of the **Sub-iter loop** iterations since the beginning of the simulation. From the results as seen in Figure 5.3.2, we observe that:

1. **EVP-Test-Case-01-a** takes the maximum number of steps for the convergence due to more element sections (fine mesh size) and thus needs more iterations for stress field stability. Convergence for s_{xx} component is not observed in the first fixed-point iteration.
2. **EVP-Test-Case-01-c** and **EVP-Test-Case-01-d** take the least (and same) number of steps for convergence, with almost same residual at the convergence
3. a new fixed-point iteration follows if the maximum number of iterations is reached or the deviatoric stress converges
4. *The last iteration of the state-variables from the **Sub-iter loop** in the iteration i is used for the initialization of state-variables in the iteration $i+1$. In this iteration, the streamline is updated, hence the stress-field observes a large change and hence is seen as a peak in the convergence residual. This peak gradually decreases as we move closer to convergence*
5. The first convergence of the deviatoric stress field takes the longest. Once a stable elasto-plastic field is computed, it converges much faster.

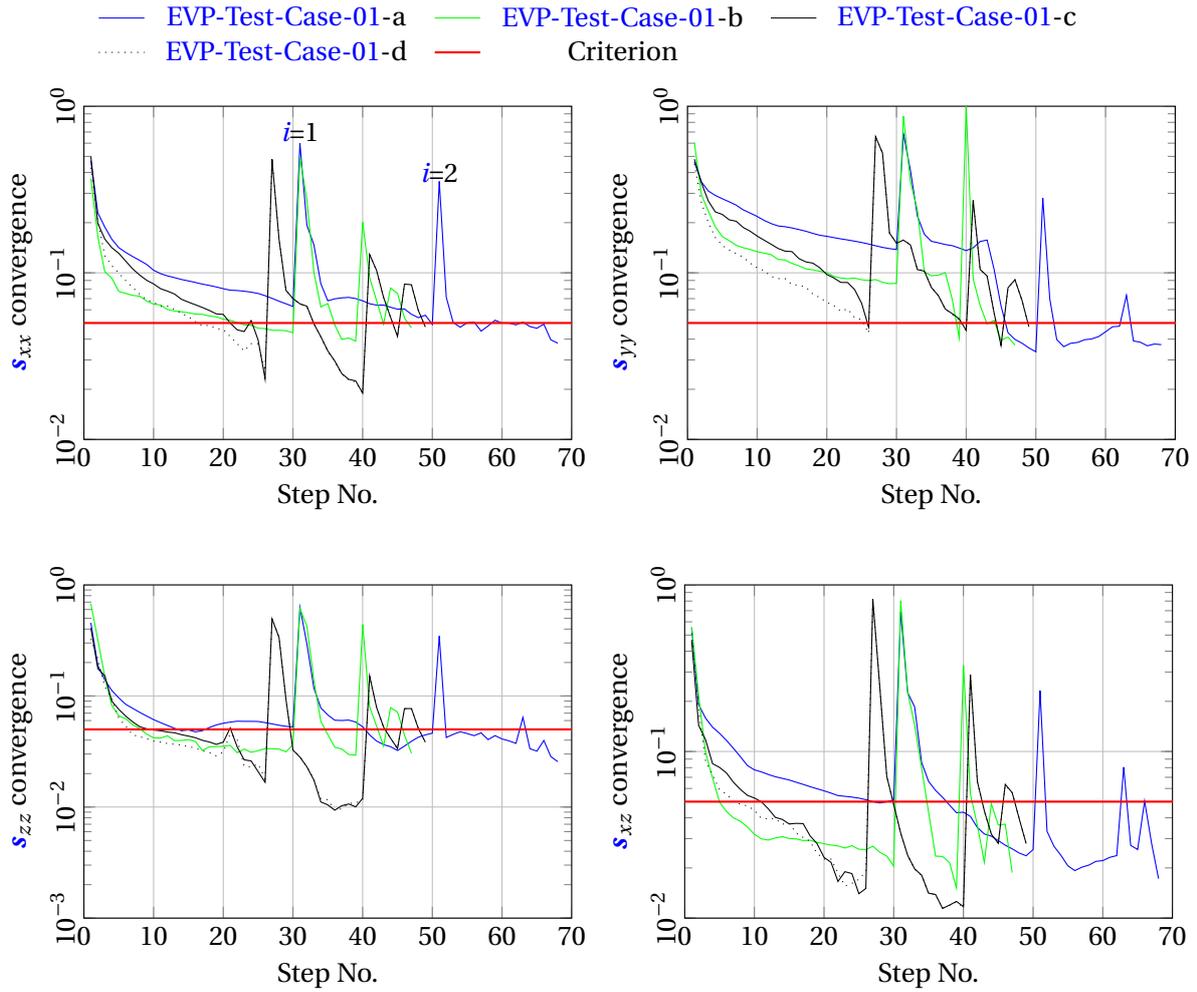


Figure 5.3.2: Convergence of different deviatoric stress components for EVP-Test-Case-01

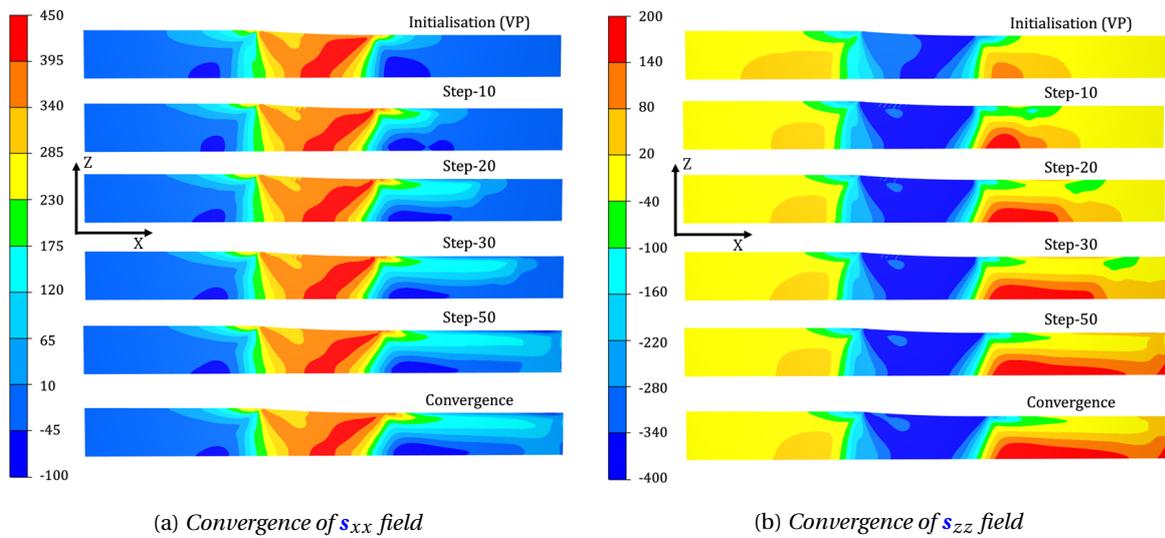


Figure 5.3.3: Convergence of deviatoric stress components

Subsequently, we can also visualize the convergence of the deviatoric stress field components s_{xx} and s_{zz} for *viscoplastic initialization*, *step-10*, *step-20*, *step-30*, *step-50* and *convergence* in the Figure 5.3.3. A relaxation of the stress field is clearly seen in the viscoplastic solution in the down-

wind due to viscous effects. However, when the material behavior is of elasto-viscoplastic nature, no such relaxation is visible. We see the largest stress norm below the tool and gradually reducing in the upwind and downwind region of the tool until a stabilized stress field is obtained. This "St.Venants" like zone is roughly 1-2 times the thickness of the workpiece. The stress field must stabilize after a distance (2-3 times the thickness of the workpiece) after the tool-bite region. However, at the output plane Γ_{out} , we have a free-surface condition (as seen in equation (3.2.10)) which results in the perturbation of the residual stress. This perturbation effect is also observed for a distance equivalent to 2-3 times the workpiece thickness (from the output plane Γ_{out}). The free surface condition was reasonable for the viscoplastic material models, where we observed stress relaxation, but for the elasto-viscoplastic materials, suitable boundary conditions must be used for the output plane Γ_{out} as highlighted by [3].

The **global results** from the simulations carried out with the different models presented are compared in Table 5.3.4. The validation of the steady-state solution is done with the incremental resolution of the same problem with the Mesh-2 in ForgeNxt[®] inc. In addition, the results from the initialized viscoplastic solution with Hensel-Spittel (HS) and Norton-Hoff (NH) models are also presented with each of the test-case. We have the following main observations from the results:

1. There is good agreement between the results found from the Hensel-Spittel viscoplastic model, hence, assuring the same start for each of the EVP test case simulations
2. The Tool Force and Torque computed with all four simulations with the steady-state algorithm have a very good agreement with the incremental solution
3. The speedup achieved with the steady-state formulation is between 15-23 times depending on the mesh used for the simulation
4. The solution with the Norton-Hoff viscoplastic initialization (EVP-Test-Case-01-d) has a very good agreement with the other steady-state as well as the incremental solutions, hence proving that the algorithm is robust enough to find the same solution with different initializations
5. The non-homogeneous mesh reduces the solution time by 1.35 times, without affecting much the solution accuracy

Resolution Method	Material Model	No. of steps	Time for resolution (min)	Tool Force (Tonnes)	Tool Torque (N-m)
EVP-Test-Case-01-a	VP-HS	-	-	2.05	251
	EVP	68	4.78	2.18	243.6
EVP-Test-Case-01-b	VP-HS	-	-	2.01	247.5
	EVP	47	4.4	2.17	243.9
EVP-Test-Case-01-c	VP-HS	-	-	2.00	246
	EVP	49	3.46	2.18	247.2
EVP-Test-Case-01-d	VP-NH	-	-	1.54	209.3
	EVP	49	3.25	2.18	247.2
Incremental	EVP	599	75	2.19	242.2

Table 5.3.4: Comparison of global results for EVP-Test-Case-01 from ForgeNxt[®]2016 EVP Algorithm-1 in comparison to the ForgeNxt[®] inc resolution (on 1 core)

The rolling direction velocity and pressure fields with the incremental and steady-state results are presented in Figure 5.3.4a and 5.3.4b respectively. *The part near the engagement phase in the incremental model is trimmed to remove the transient effects.* The velocity computed from both the formulations have an excellent agreement. The pressure is accurately computed under and the upwind of the tool, but we observe a difference on the free surface in the downwind direction to

the tool because of the end effect. We also see an excellent prediction of rolling direction stress σ_{xx} below the tool and vertical direction stress σ_{zz} in Figures 5.3.5a and 5.3.5b respectively. However, the residual state of the rolling direction stress in the downwind of the tool is slightly under-predicted (with steady-state) near the free surface and agreement is excellent if we trim the zone impacted by the artificial boundary condition downstream, as done for the incremental solution.

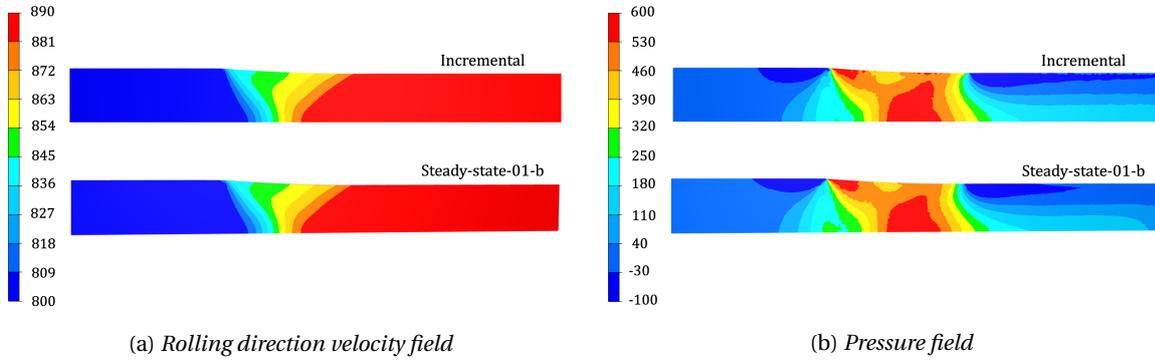


Figure 5.3.4: Comparison of solution for *EVP-Test-Case-01* with steady-state and incremental simulations

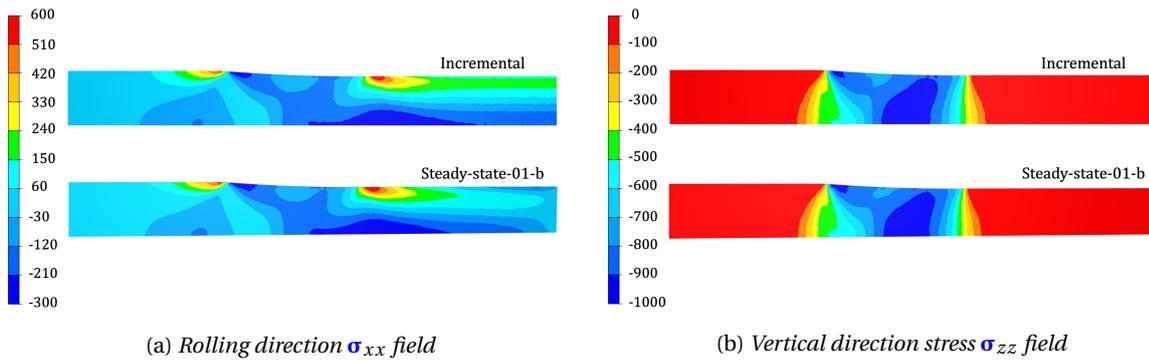


Figure 5.3.5: Comparison of solution for *EVP-Test-Case-01* with steady-state and incremental simulations

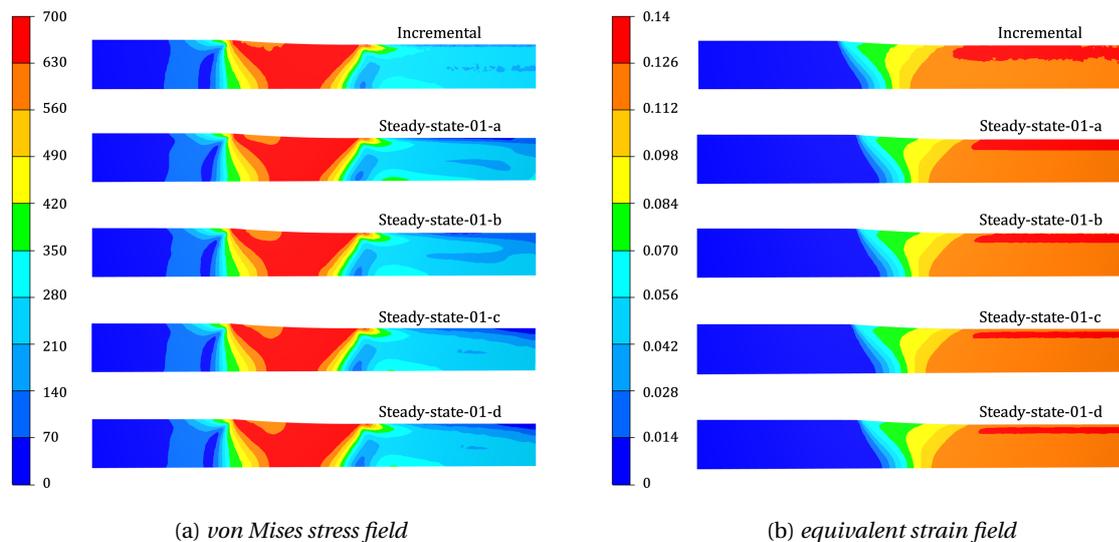


Figure 5.3.6: Comparison of equivalent stress and strain fields with different simulations

The equivalent stress and strain fields computed with the different *EVP-Test-Case-01* model variants (in Table 5.3.2) are presented in the Figure 5.3.6. It can be seen that the steady-state algo-

rithm is able to predict the von Mises stress under and upwind to the tool accurately. The residual stress is lower with all steady-state simulations on the top free-surface in the downwind of the tool-bite which could partly come from the Saint Venant's effect. However, the gradient is also high in the Z-direction, which is unlikely due to the free surface condition. Hence, some other effects are present as well near the output plane that need to be studied and corrected. The equivalent strain below the tool is also in good agreement except with the non-homogeneous meshes (EVP-Test-Case-01 c & d), where we see an under-prediction of the strain at the free surface, and the maximum equivalent strain is seen below the free surface.

5.3.5 Conclusions from the EVP-Test-Case-01 results

Four tests have been conducted on a simple pseudo-2D simple rolling case with the proposed framework for the resolution of elasto-viscoplastic problems using steady-state formulation. The results from the proposed algorithm are validated with the incremental solution of the same problem in ForgeNxt[®] inc. The presented results highlighted mainly the comparison of global results in terms of tool forces and torque, which demonstrated an excellent agreement with the respective validation test-case. The various solution contours have also been compared to test the ability of the algorithm to predict the steady-state solution. Except for the end effects observed in stress field contours, we observe a very good agreement with the incremental solution.

5.4 Improvement of speedup with ForgeNxt[®] 2016 EVP Algorithm-2

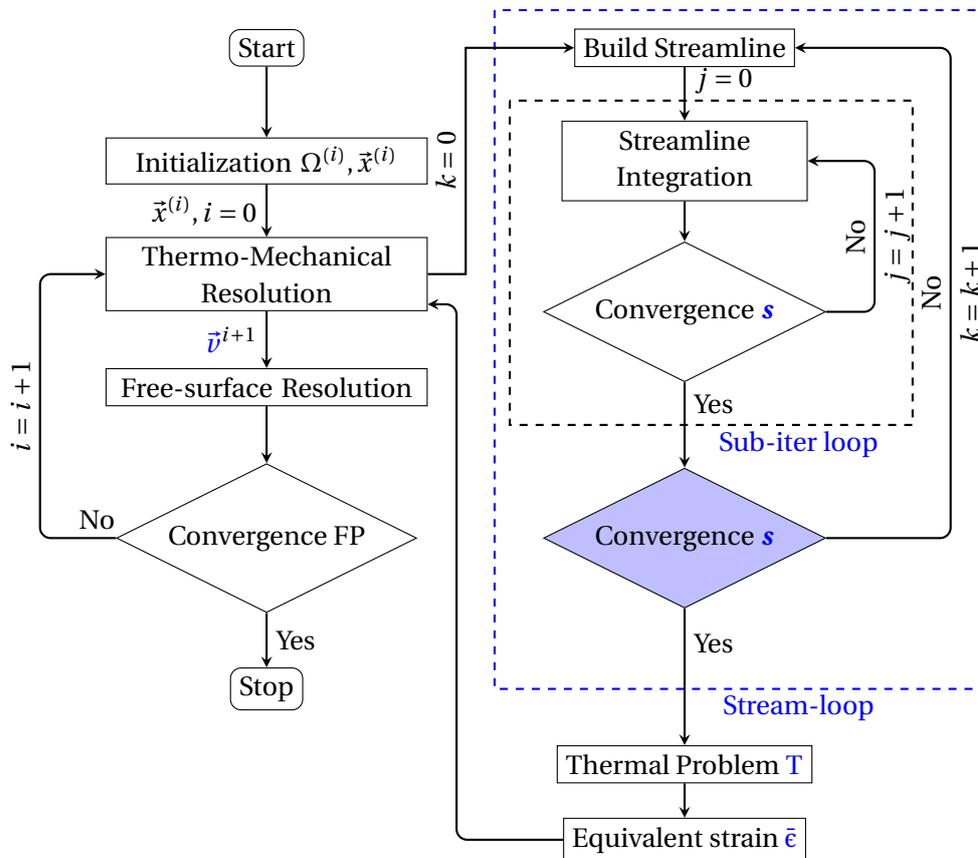


Figure 5.4.1: Algorithm-2 for EVP steady-state resolution with streamline loop in ForgeNxt[®] 2016

In the Sub-iter loop of the present algorithm (Figure 5.2.10), multiple interpolations of the state-variables are required from the mesh to the streamlines and vice-versa. This is a diffusive process and can accumulate error after numerous iterations, especially if a strict convergence criterion is

chosen for the **Sub-iter loop**. In addition, the new velocity field computed inside the **Sub-iter loop** must also entail the recomputation of streamlines. In the presented algorithm, both the steady-state shape and velocity are changed simultaneously, and the stress solution does not conform to the new streamlines (seen as peaks in Figure 5.3.2) and thus retards the convergence. Hence, a change is proposed in the steady-state algorithm (see Figure 5.4.1) by adding another step to update the streamlines in the mechanical computation.

5.4.1 Detailed ForgeNxt®2016 EVP Algorithm-2

A new loop called the **Stream-loop** is introduced into the algorithm. This step aims to update the streamlines after a certain level of stress convergence is achieved. The updated velocity field is used to update the streamlines or the location of projection points *pre* after few iterations of the **Sub-iter loop**. Thus, the **Stream-loop** encompasses the **Sub-iter loop** and aims to improve the convergence speed-up.

5.4.2 Updated Stream-loop and Sub-iter loop convergence criteria

As seen in the Figure 5.4.1, two convergence criteria must be described for the **Stream-loop** and **Sub-iter loop** respectively. The **Sub-iter loop**, unlike the ForgeNxt®2016 EVP Algorithm-1 (Figure 5.2.10), does not require a global convergence of the deviatoric stress, but only in the vicinity of the tool where the deformation takes place. Hence, the new convergence criterion of **Sub-iter loop** looks for the stability of the deviatoric stress field only in the vicinity of the tool with the following criterion. This is ensured if the infinite norm of the deviatoric stress error between two **Sub-iter loop** iterations is stagnant.

$$d \in \{1, 6\} \quad \frac{\max_{int}^{j+1} |s_d^{j+1}(int) - s_d^j(int)| - \max_{int}^j |s_d^j(int) - s_d^{j-1}(int)|}{\max_{int}^{j+1} |s_d^{j+1}(int) - s_d^j(int)|} \leq \epsilon_{si} \quad (5.4.1)$$

The streamlines are updated after the stresses below the tool converge. For ensuring the global convergence of the deviatoric stress field, we apply the following criterion for the convergence of the **Stream-loop**. This is the same criterion for the convergence of deviatoric stress in ForgeNxt®2016 EVP Algorithm-1 as presented in Section 5.2.7.

$$d \in \{1, 6\} \quad \frac{\max_{int} |s_d^{j+1}(int) - s_d^j(int)|}{\max_{int} s_d^{j+1}(int) - \min_{int} s_d^{j+1}(int)} \leq \epsilon_{stream} \quad (5.4.2)$$

The **Sub-iter loop** convergence criterion was found empirically and is chosen as $\epsilon_{si} = 0.2$. The **Stream-loop** convergence criterion is chosen as $\epsilon_{stream} = 0.05$ in line with the convergence criterion for global stress stability in the ForgeNxt®2016 EVP Algorithm-1. The maximum number of iterations in the **Sub-iter loop** are 10 and **Stream-loop** are 3 in order to limit the total number of iterations to 30 consistent with Algorithm-1. With this limit, the streamlines are updated either if the **Sub-iter loop** is converged or maximum number of iterations are reached. In addition, if the convergence criterion for **Stream-loop** is reached, it is also treated as the convergence of inner **Sub-iter loop** loop.

5.4.3 EVP-Test-Case-01 resolution with ForgeNxt® 2016 EVP Algorithm-2

Resolution Method	Material Model	No. of steps	Time for resolution (min)	Tool Force (Tonnes)	Tool Torque (N-m)
Steady-state	EVP-Algo1	49	3.25	2.18	247.2
	EVP-Algo2	37	3.87	2.18	247.2
Incremental	EVP	599	75	2.19	242.2

Table 5.4.1: Comparison of global results for EVP-Test-Case-01-d with the ForgeNxt® 2016 EVP Algorithm-1 and Algorithm-2 in comparison to the ForgeNxt® inc resolution (on 1 core)

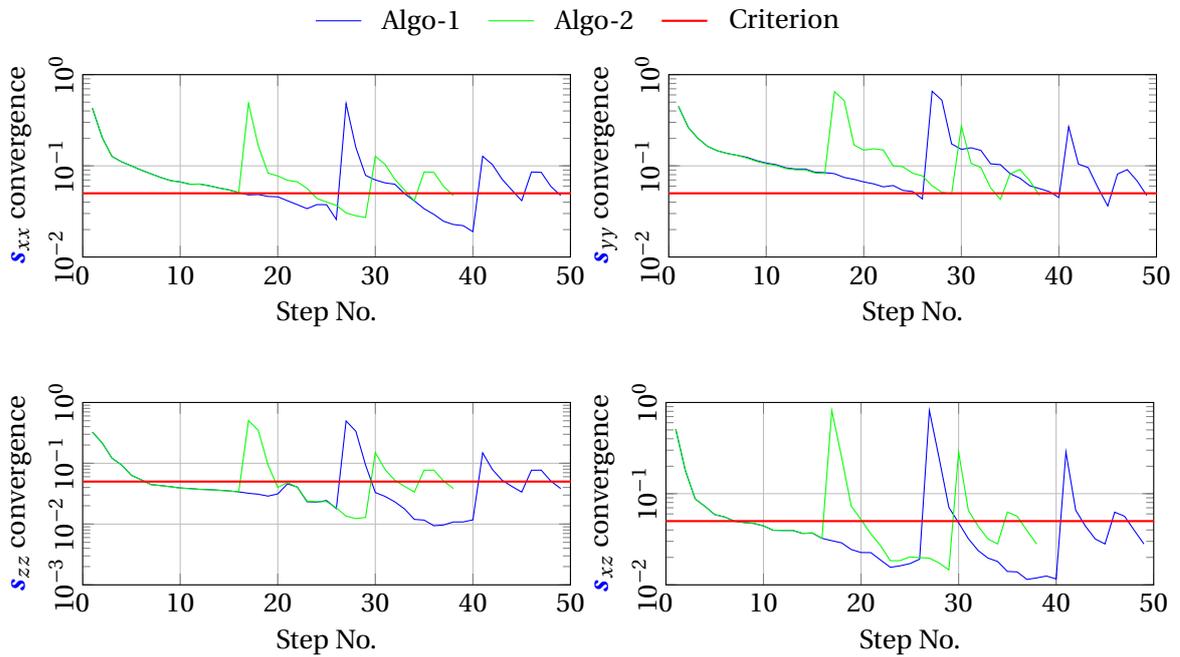


Figure 5.4.2: Convergence of different deviatoric stress components for EVP-Test-Case-01-d with ForgeNxt® 2016 EVP Algorithm-1 & Algorithm-2

The ForgeNxt® 2016 EVP Algorithm-2 is tested with EVP-Test-Case-01-d and it is found that the results for the updated algorithm are the same as the ones obtained from Algorithm-1, hence only the global results are compared in the Table 5.3.2. We see a reduction of the number of iterations for convergence from 49 with Algorithm-1 to 37 with Algorithm-2, but there is hardly any difference in the resolution time. The convergence of the Stream-loop residual for deviatoric stress is seen in Figure 5.4.2. An offset is seen in the peaks representing shape update (at new fixed-point iteration) with the new Algorithm-2. Nonetheless, the convergence curve with Algorithm-2 is seen to have no signature of the streamline update. It could be because there might be a very small change or no change in the position of the *pre* at the maximum *s* error location.

5.5 Validation of ForgeNxt® 2016 EVP algorithm with complex test cases

Next, we test the Algorithm-1 and Algorithm-2 with two more complex test cases: (i) EVP-Test-Case-02 (ii) EVP-Test-Case-03.

5.5.1 EVP-Test-Case-02 resolution with ForgeNxt® 2016 EVP algorithm

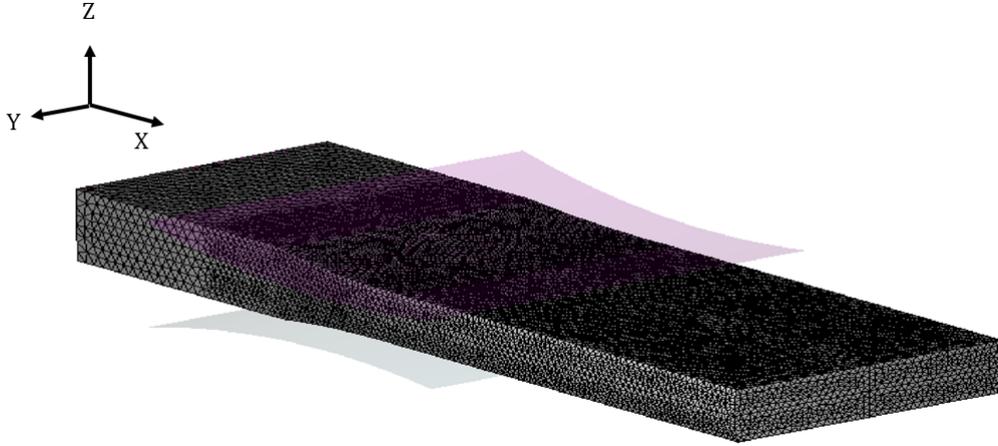


Figure 5.5.1: Mesh model for the EVP-Test-Case-02

EVP-Test-Case-02 is a thick plate rolling test-case which was also tested for contact formulation in the Section 4.5, except that here it is tested with cold conditions using Hensel-Spittel EVP material model (Table 5.3.1). With only two symmetry planes (one horizontal and one vertical), the material flow is allowed in the lateral direction. A constant mesh size of length 1 mm is used for the discretization of the domain (see Figure 5.5.1). The global results (in Table 5.5.1) with steady-state resolution (with Algorithm-1 & 2) are seen to have a very good agreement with the incremental resolution results. An important observation is that there is no change observed in the results when changing from Algorithm-1 to Algorithm-2, similar to what was observed with EVP-Test-Case-01-d. The only difference seen is a small reduction in the number of steps and the time for resolution with the Algorithm-2.

Resolution Method	Material Model	No. of steps	Resol. Time (min)	Force (T)	Torque (kN-m)
Steady-state	EVP-Algo1	37	11.1	212.6	44.93
	EVP-Algo2	33	9.92	212.6	44.93
Incremental	EVP	524	94	214.6	43.66

Table 5.5.1: Comparison of global results for EVP-Test-Case-02 from the ForgeNxt® 2016 EVP Algorithm-1 and Algorithm-2 in comparison to the ForgeNxt® inc resolution (on 1 core)

The converged solutions for the von Mises stress and the equivalent strain with the steady-state and incremental methods are compared in Figure 5.5.2. In general, the stress field is predicted quite well with the steady-state algorithm. Nonetheless, there are differences in the post-bite region after the contact with steady-state algorithm over-predicting the stress. Near the edges, we observe an under-prediction of the stress field. The prediction of equivalent strain is much better.

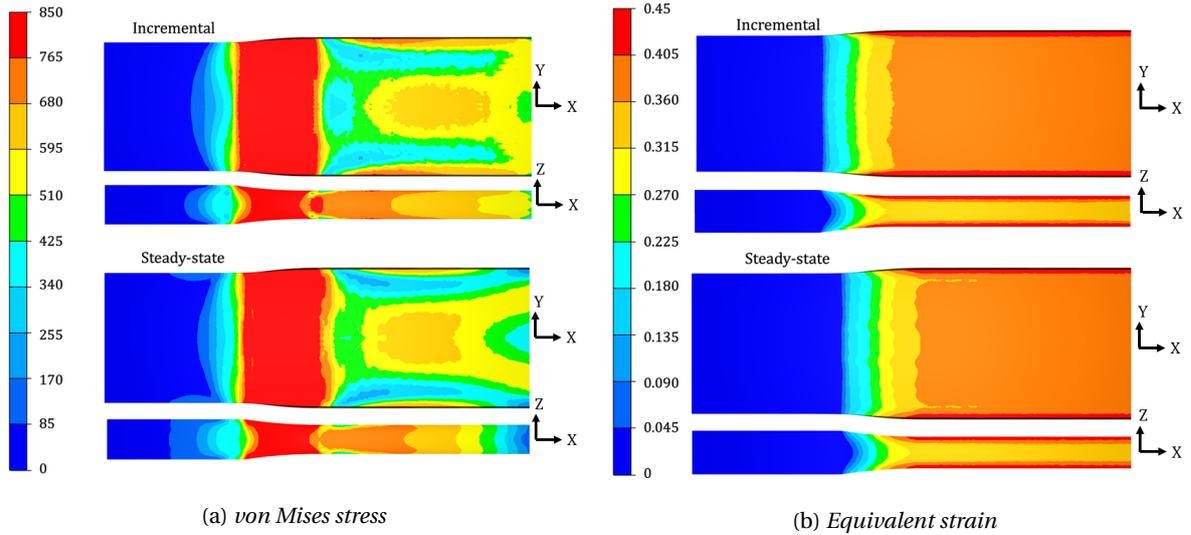


Figure 5.5.2: Comparison of solution for *EVP-Test-Case-02* with steady-state and incremental simulations

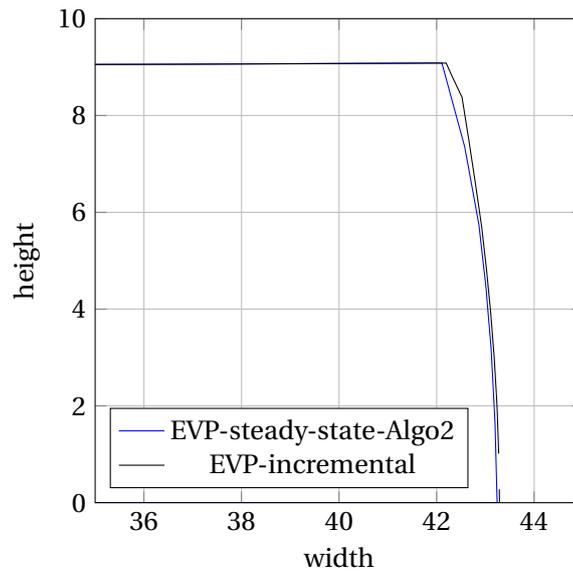


Figure 5.5.3: Comparison of output plane shape (Initial half-width: 40 mm) for *EVP-Test-Case-02* with incremental and steady-state formulations

The comparison of steady-state shape computed with steady-state (Algorithm-2) and incremental algorithms is shown in Figure 5.5.3. It can be observed that both solutions have a good agreement, except that the lateral spread is underpredicted with the steady-state algorithm near the top edge. This could be due to a relatively coarser element at the same location or difference in friction modeling in the incremental resolution. The convergence of the deviatoric stress components can be seen in Figure 5.5.4. We see slight improvement in convergence iterations but the residual is more or less the same with or without streamline update except for components s_{xx} , s_{zz} . The peaks of residual are most superimposed, except that with the Algorithm-2, we observe an extra peak at 10th step. This additional peak corresponds to the streamline update, which happens only one time during the $i = 1$ whereas later the global convergence criterion (for *Stream-loop*) is met, so no streamline update takes place.

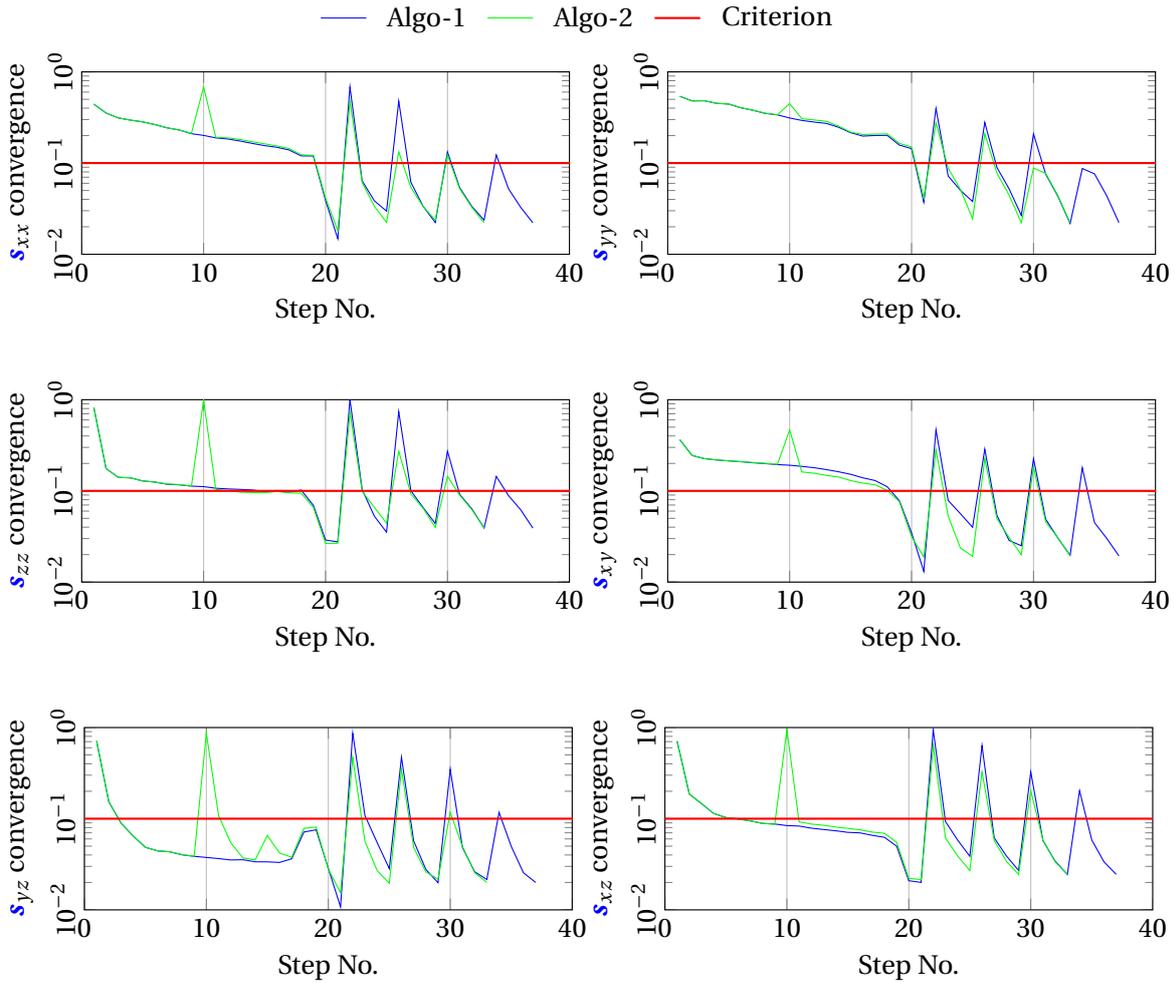


Figure 5.5.4: Convergence of deviatoric stress components for **EVP-Test-Case-02** with steady-state algorithms

5.5.2 **EVP-Test-Case-03** resolution with **ForgeNxt®2016** EVP algorithm

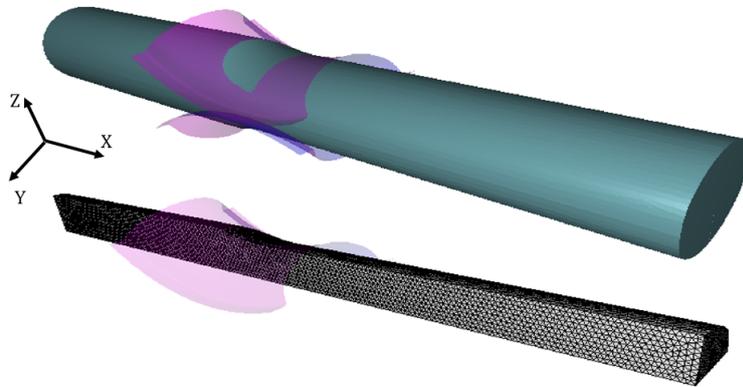


Figure 5.5.5: Mesh models used for the Test-case 03

The third test-case is a shape rolling problem in which a circular section workpiece is rolled to an oval shape. The rolls rotate at 29.1 rpm. The model used for this test is shown in Figure 5.5.5. The Hensel-Spittel material model is used with the same properties as used in the previous two cases (see Table 5.3.1). The global results (in Table 5.5.2), like in the previous test-cases, are predicted

well with the steady-state solver. However, it can be noticed that the steady-state Algorithm-1 takes a much longer time and almost double the number of iterations in comparison to the Algorithm-2.

Resolution Method	Material Model	No. of steps	Time for resolution (min)	Tool Force Z (T)	Tool Torque (kN-m)
Steady-state	EVP (Algo-1)	120	45.5	291.6	100
	EVP (Algo-2)	65	25.12	291.9	100.1
Incremental	EVP	443	212	294.53	100.7

Table 5.5.2: Comparison of global results for EVP-Test-Case-03 from the ForgeNxt®2016 EVP in comparison to the ForgeNxt®inc resolution (on 1 core)

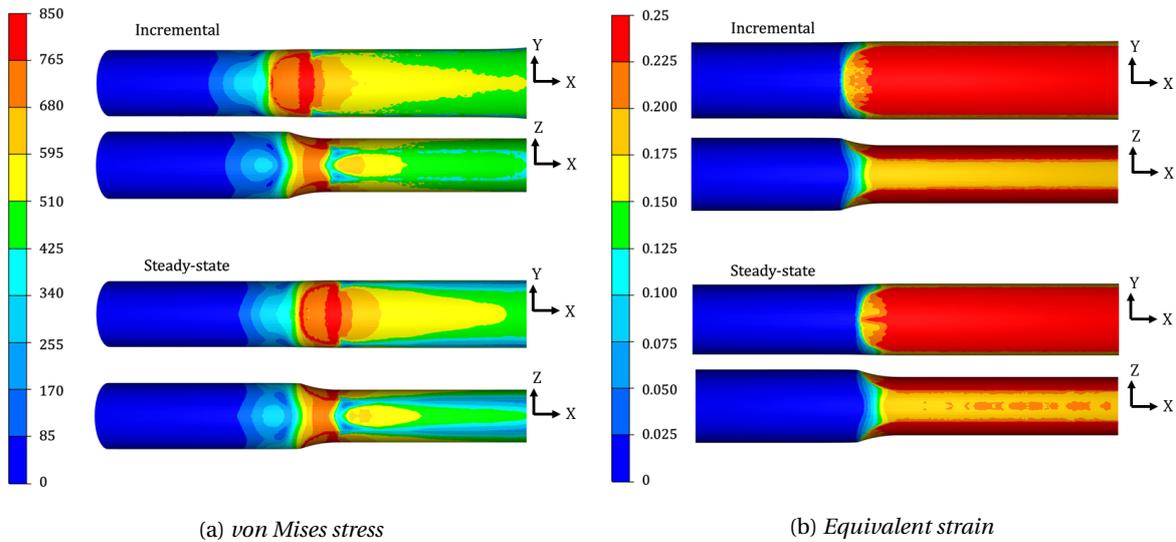


Figure 5.5.6: Comparison of solution for EVP-Test-Case-03 with steady-state and incremental simulations

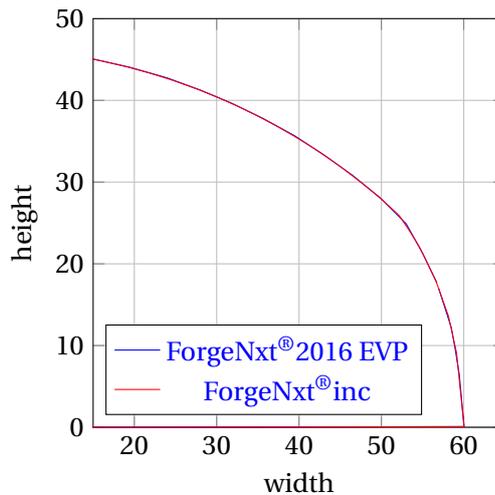


Figure 5.5.7: Comparison of output plane shape for EVP-Test-Case-03 with incremental and steady-state formulations

The converged solution for the von Mises stress and the equivalent strain fields with the incremental and steady-state formulations are compared in Figure 5.5.6. Firstly, we see smoother contours without any oscillations with steady-state formulation. We see an overall excellent prediction of the solution contours for the gain achieved in resolution time. The steady-state shape

computed with the two formulations compared in the Figure 5.5.7 demonstrate an excellent agreement between the two solutions. The convergence of the deviatoric stress components for the **EVP-Test-Case-03** is presented in Figure 5.5.8. This test is a perfect example showcasing how the streamline update can help accelerate the convergence. In the first fixed-point iteration, we see how a slowly converging residual which takes almost 50 steps to see first convergence, is accelerated with Algorithm-2 which takes only 26 iterations for the same. Even further into the computation, the stress convergence with the Algorithm-2 is consistent and rapid.

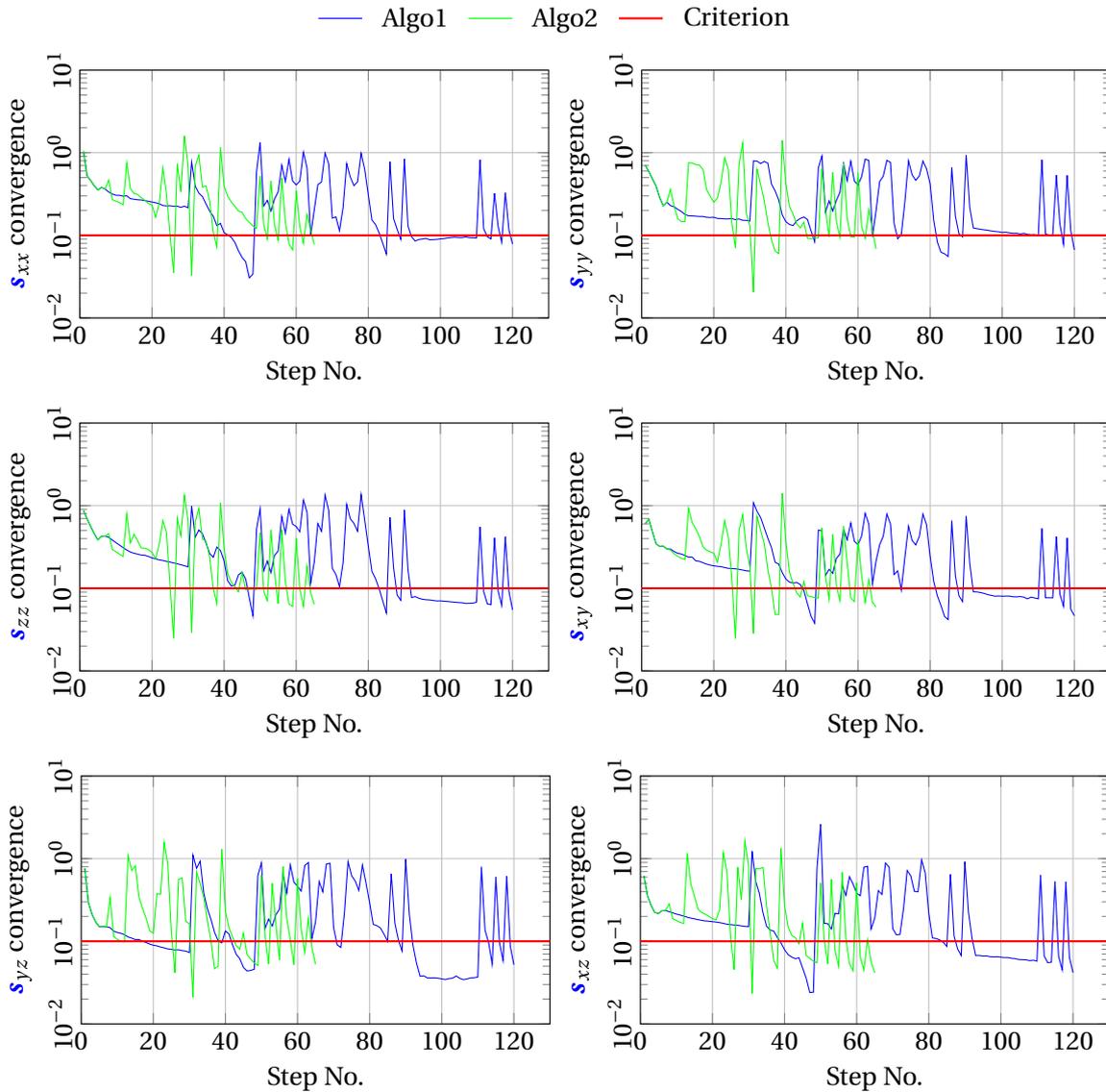


Figure 5.5.8: Convergence of deviatoric stress components for **EVP-Test-Case-03** with steady-state algorithms

5.5.3 Conclusions from the **EVP-Test-Case-02** and **EVP-Test-Case-03** results

Three different test-cases have been tested with the proposed two algorithms for the resolution of elasto-viscoplastic problems using steady-state formulation. The results from the proposed algorithm are validated with the incremental solution of the same problem in **ForgeNxt[®] inc.** The results found with both the algorithms are in excellent agreement with the incremental solution. The steady-state shape is also accurately predicted with the steady-state method. While comparing the results between the two steady-state algorithms proposed, we found that the global results as well as the field contours are nearly identical. However, the Algorithm-2 results in faster conver-

gence of the stress field with a lower number of steps and lesser computation time for resolution of the problem.

5.6 Conclusions from the Chapter

The present chapter focused on finding a methodology to resolve continuous forming problems with EVP materials with the steady-state formulations. From the point of view of the existing iterative solver of the ForgeNxt[®]2016, the HPTS approach inspired the extension of the existing algorithm to solve such problems. The method requires to trace the streamlines in the discretized domain and integrate the elasto-viscoplastic constitutive model on these streamlines. The HPTS approach is advantageous as it uses the incremental Prandtl-Reuss equations to compute the elasto-plastic stress field which can be used both for incremental and steady-state formulations. Though originally, this method had the advantage of the structured mesh for carrying out this task, such a framework is not natural with the unstructured mesh in Fg3[®]. Thus a framework has been proposed to build discrete streamlines in the unstructured mesh. Compared to the heterogeneous time-step in HPTS approach, we use a constant pseudo-time-step in order to compute the projection point. In steady-state formulation, the projection point of an integration point describes its state at the previous time-step. Hence, the state-variables need to be interpolated at the projection point. To make this interpolation non-diffusive, we use the P1+ field solution built from a smoothed (P1) deviatoric stress field found from SPR recovery and the P0 field known at the integration points. A sub-iteration loop is introduced inside the mechanical step for finding a stable stress field which respects the equilibrium conditions in the strongly coupled problem. The proposed algorithm is tested on a simple cold rolling test case with Hensel-Spittel elasto-viscoplastic material model and the predicted results are seen to have a good agreement with the incremental solution of the same problem. A modification of the algorithm is proposed with an additional streamline loop that aims to recompute the streamlines with the updated velocity field after the stress converges below the tool. Both these algorithms are tested with the same simple cold rolling test-case and the one with streamline update is found to reduce the number of iterations for convergence.

Further, two more test-cases: a flat rolling and a shape rolling, are tested with both the presented algorithms, and the results are validated with the incremental one of the same problem. The results are found to have an excellent agreement and promise to bring down the resolution time enormously. The algorithm with a streamline update is observed to be more robust and faster than the other.

Résumé

Ce chapitre se concentre sur la recherche d'une méthodologie pour résoudre les problèmes de laminage à froid pour des matériaux *EVP* avec la formulation stationnaire. Du point de vue du solveur itératif existant de *ForgeNxt*[®] 2016, l'approche *HPTS* a inspiré l'extension de l'algorithme existant pour résoudre de tels problèmes. La méthode nécessite de tracer les lignes de courant dans le domaine discrétisé et d'intégrer le modèle constitutif élasto-viscoplastique sur ces lignes de courant. L'approche *HPTS* est avantageuse car elle utilise les équations incrémentales de Prandtl-Reuss pour calculer le champ de contraintes élastoplastiques qui peut être utilisé pour les formulations incrémentale et stationnaire. Bien qu'à l'origine, cette méthode avait l'avantage du maillage structuré, ce qui simplifie l'intégration des équations sur les lignes de courant formées par l'alignement des noeuds consécutifs, ce n'est pas naturel avec le maillage non structuré comme dans *Fg3*[®]. Ainsi, un cadre a été proposé pour construire des lignes de courant discrètes dans un maillage non structuré. Comparé au pseudo-pas de temps hétérogène dans l'approche *HPTS*, nous utilisons un pseudo-pas de temps constant afin de calculer le point de projection. Dans la formulation stationnaire, le point de projection d'un point d'intégration décrit son état au "pas de temps" précédent. Par conséquent, les variables d'état doivent être interpolées au point de projection. Pour rendre cette interpolation non diffusive, nous utilisons un champ *P1+* construit à partir d'un champ de contrainte déviatorique lissé (*P1*) trouvé à partir de la méthode *SPR* et du champ *P0* connu aux points d'intégration. Une boucle de sous-itération est introduite à l'intérieur de l'étape mécanique pour trouver un champ de contrainte stable qui respecte les conditions d'équilibre dans le problème fortement couplé. L'algorithme proposé est testé sur un cas test de laminage à froid simple et les résultats semblent être en bon accord avec la solution incrémentale du même problème. Une modification de l'algorithme est proposée avec une boucle supplémentaire pour l'actualisation de lignes de courant avec le champ de vitesse actualisé après la convergence de la contrainte sous l'outil. Ces deux algorithmes sont testés avec le même cas test de laminage à froid simple et celui avec l'actualisation de lignes de courant réduit le nombre d'itérations de convergence. De plus, deux autres cas tests, avec un laminage à plat et un laminage de forme, sont testés avec les deux algorithmes présentés, et les résultats sont validés avec la solution incrémentale du même problème. Les résultats montrent un excellent accord (avec les résultats de la méthode incrémentale) et un temps de résolution considérablement réduit. On observe que l'algorithme avec l'actualisation de lignes de courant est plus robuste et plus rapide que le premier.

Bibliography

- [1] C. Aliaga. *Simulation numérique par éléments finis en 3D du comportement thermomécanique au cours du traitement thermique d'aciers : application à la trempe de pièces forgées ou coulées*. PhD thesis, ENSMP, 2000. URL <http://www.theses.fr/2000ENMP0882>.
- [2] S. Guerdoux. *Numerical simulation of the friction stir welding process*. PhD thesis, ENSMP, 2007.
- [3] A. Hacquin. *Modélisation Thermomécanique Tridimensionnelle du Laminage - Couplage Bande/ Cylindres*. PhD thesis, ENSMP, 1996.
- [4] A. Hacquin, P. Montmitonnet, and J.P. Guillerault. A steady state thermo-elastoviscoplastic finite element model of rolling with coupled thermo-elastic roll deformation. *Journal of Materials Processing Technology*, 60(1):109 – 116, 1996. ISSN 0924-0136.
- [5] K. W. Kpodzo. *Accélération des calculs pour la simulation du laminage à pas de pèlerin en utilisant la méthode multimallages*. Theses, Ecole Nationale Supérieure des Mines de Paris, March 2014. URL <https://pastel.archives-ouvertes.fr/pastel-01038003>.
- [6] O.C. Zienkiewicz and J.Z. Zhu. The superconvergent patch recovery (SPR) and adaptive finite element refinement. *Computer Methods in Applied Mechanics and Engineering*, 101(1-3):207–224, 1992.

Chapter 6

Conclusions and Perspectives

Contents

6.1	Conclusions	158
6.1.1	Conclusions-1: Contact formulation	158
6.1.2	Conclusions-2: Steady-state algorithm for history-dependent materials	159
6.2	Perspectives	160
6.2.1	Unsymmetric rolling cases	160
6.2.2	Resolution approach for long interstand distances	162
6.2.3	Boundary condition at output plane for steady-state elastoplasticity resolution	162
6.2.4	Speed-up of the steady-state elastoplasticity resolution	162

This chapter is focused on the summary of important developments made in this thesis and the future perspectives of the steady-state formulation for solving metal forming problems of industrial scale and complexity.

6.1 Conclusions

Starting from the pre-existing version ForgeStat, there were two clear distinct objectives of the present thesis and the conclusions on each subject are detailed below:

6.1.1 Conclusions-1: Contact formulation

Firstly, the unstable solution with the ForgeNxt®2016 is found to be linked with the contact. In the strongly coupled multi-field formulation, the contact surface is also an unknown and an initial assumption is corrected in the two steps of the iterative algorithm. The contact conditions are hence applied both on velocity and shape correction fields, but it is found that these two conditions are not consistent. Both the contact conditions are described with the normal velocity: in the Step-1, it is computed with the scalar product of velocity and tool normal in a nodal form and in the Step-2, it is described with the scalar product of velocity and mesh normal in a weighted-residual form. We seek to improve the stability of the solution by using contact conditions which are consistent. To do so, a nodal form of free-surface equation is derived at first using a *nodal condensation* approach which is applied on the free-surface residual in the Step-2. This nodal free surface residual is defined with the scalar product of velocity and an upwind biased normal or *upwind normal*. The upwind bias comes from the choice of SUPG test function for the averaging of the normal vector. In addition, the SUPG stabilization coefficient used for the test function is found to be $\alpha=1/3$ for the 2D problems whereas it is generally used as $\alpha=1/2$. Before finding the contact condition from the nodal free-surface equation, it is tested for its ability to predict 2D and 3D analytical shape functions. It is found that the method is convergent, but less accurate in comparison to the original weighted residual form. However, the largest error is located at locations with high-velocity gradients which is not a common scenario with metal forming problems. In addition, it is also found that the SUPG stabilization coefficient $\alpha=1/3$ results in better accuracy. We choose to use the nodal form of the free-surface equation only to find a consistent contact condition for the velocity computation in Step-1. An initial explicit contact condition is derived considering the mathematical behavior of the residual for a contact node. Initial tests proved that only this change improves the robustness of the algorithm. The new contact condition is extended to problems with shape singularity, such as edges. However, an implicit form of the contact condition is ideal for further improving the robustness, as a special consideration for the first contact nodes is needed which are the most sensitive and generally the reason behind the unstable solution. This condition on velocity field is applied on the current configuration, which is unknown in the mechanical step. Derived from the material impenetrability constraint on the shape correction, a complete decoupling of velocity and shape fields in the free surface equation with some assumptions leads to the required implicit contact condition. The proposed explicit and implicit contact conditions are implemented in the ForgeNxt®2016 and tested with industrial hot rolling problems of flat and long products. The results were compared with their resolution with incremental method until a steady-state is achieved. The comparison of results demonstrates accurate prediction with improved robustness of the algorithm. It was also demonstrated that the implicit condition provides stable solution with all the problems tested (in comparison to the explicit condition), and also reduction of number of iterations. It is also demonstrated that the computation time is typically reduced by 20 times with the steady-state formulation in comparison to the incremental resolution. Hence, the first objective of the improvement of the robustness of the algorithm was well achieved.

6.1.2 Conclusions-2: Steady-state algorithm for history-dependent materials

The treatment of elastoviscoplastic materials in steady-state algorithms brings in a variety of challenges to be addressed. From the point of view of the existing iterative solver of the *ForgeNxt*^{®2016}, the *HPTS* approach [2, 3] inspired the extension of the existing algorithm to solve such problems. In this approach, a streamline framework is the prerequisite for the resolution of incremental Prandtl-Reuss elastoplasticity equations. In the original approach, this method was applied for structured meshes in which it is easier to trace and store the streamlines. The incremental Prandtl-Reuss equations also make it possible to reuse a major part of the incremental resolution algorithm. The state of a precedent time-step is determined with the state at a precedent integration point, which is true for the steady-state conditions. A pseudo-time-step is used for modeling time in the time-dependent elastoplasticity equations. For the unstructured meshes like in *Fg3*[®], a framework must be developed to trace the streamlines and the associated state-variables which carry the history-dependent information. Short streamlines are defined between each integration point *int* and its upwind projection *pre*. The state-variables at projected point are interpolated from the smoothed P1 fields. To minimize the diffusion, the *Superconvergent Patch Recovery (SPR)* approach is used for P0 to P1 transfer, and the interpolation is done from the enriched P1+ (P1+P0) field. Once these tools are implemented, the state-variables at *pre* (previous pseudo-time instant) are used to compute the state variables at *int* with the Prandtl-Reuss equations. The problem is strongly-coupled in stress and velocity, hence a *Sub-iterative loop for EVP resolution (Sub-iter loop)* is introduced for the integration of state-variables until a stable stress-field is reached. As soon as a new stress is computed, the equilibrium is satisfied and the velocity field is updated. For the convergence of the *Sub-iter loop*, we seek the complete stability of the stress solution. The existing *ForgeNxt*^{®2016} algorithm is modified to include the new steps involving elastoplasticity resolution. The updated algorithm is tested with a simple 2D rolling problem with different meshes using Hansel-Spittel elastoviscoplasticity model. The initialization of the fields is required, which is done with viscoplastic assumption of the material. Two different viscoplastic models are used for initialization. The results obtained from the steady-state algorithm are compared with the incremental resolution of the same problem until the steady-state is reached. The proposed steady-state algorithm provides excellent prediction with all the tested meshes & different initializations. A speedup of 16 times is achieved with the algorithm.

It is observed that if a large number of sub-increment loop iterations are prescribed to ensure that the state variables are globally stable before proceeding to the shape correction step, one may observe large diffusive error induced into the solution. This is due to the fact that as the state-variables are updated, the velocity field also changes, thus the streamlines which are built with the velocity do not correspond to the new state of the solution. Hence, we must also ensure a regular update of the streamlines to make sure that the streamlines, state-variables and velocity field correspond to each other. This update is effectuated by introducing another loop, *Streamline loop for EVP resolution (Stream-loop)*, into the algorithm.

The updated algorithm results in the reduction of the number of resolution steps for the pseudo-2D rolling problem without any change in the results. The two algorithms are further tested with two 3D plate rolling and shape rolling test-cases. The steady-state solutions from these test cases are compared with the same from the incremental formulation in *ForgeNxt*^{®inc}. The algorithm with streamline update is seen to provide an accurate prediction for both the test cases and also more robust in comparison to the algorithm without streamline update. The resolution time achieved with the former is nearly half of the time taken with the latter for the shape rolling problem. Overall, a speed-up of approximately 10 times is seen with the different problems tested with the proposed algorithm.

6.2 Perspectives

6.2.1 Unsymmetric rolling cases

Even though the present steady-state algorithm has become much robust and diverse in comparison to the existing version of the *ForgeNxt*[®]2016, there are still some problems which need to be addressed. Firstly, the forming model to be studied must have at least 2 planes of symmetry, otherwise the solution fails to converge with the steady-state algorithm. If the mesh on the two opposite sides of the symmetry plane is different (see Figure 6.2.1a), it results in an oscillatory contact and an unstable solution. Such scenarios are also possible when the workpiece is in contact with different (non symmetric) tools like tubes. In such examples, like the one presented in Figure 6.2.1b from Vallourec, a five pass tube rolling is attempted to be simulated with the steady-state algorithm. Three rolls form a stand in each pass where the external diameter of the tube is reduced from outside and internally the tube is held on a mandrel.

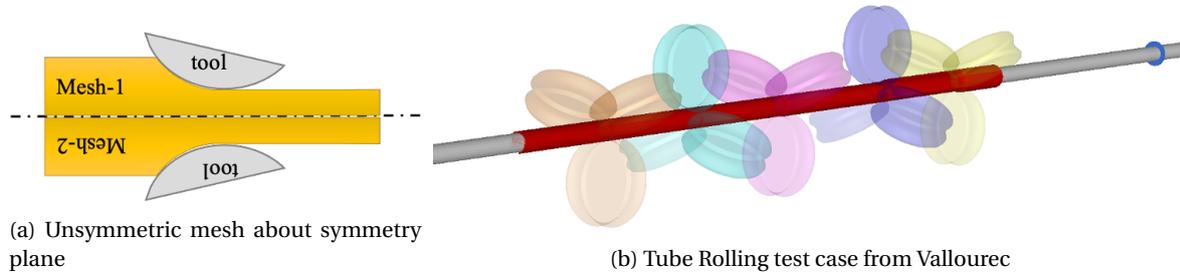


Figure 6.2.1: Unsymmetric rolling cases

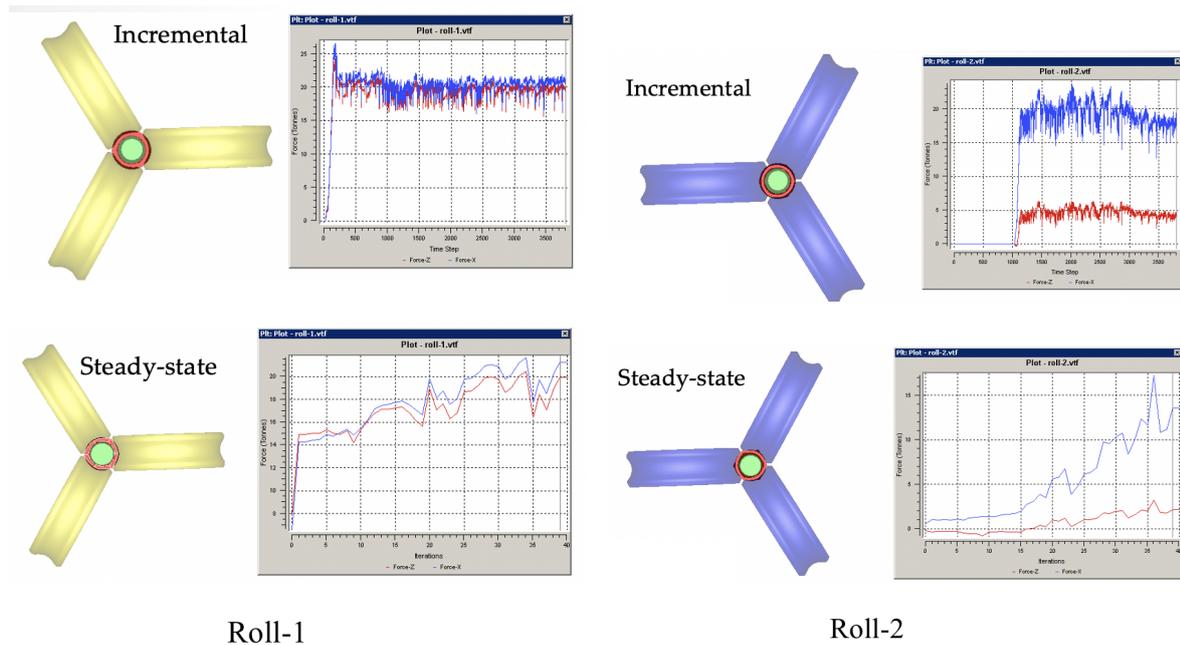


Figure 6.2.2: Tube Rolling test case forces F_x (red) F_z (blue) on roll stand 1 (left) and 2 (right) computed with incremental and steady-state algorithms

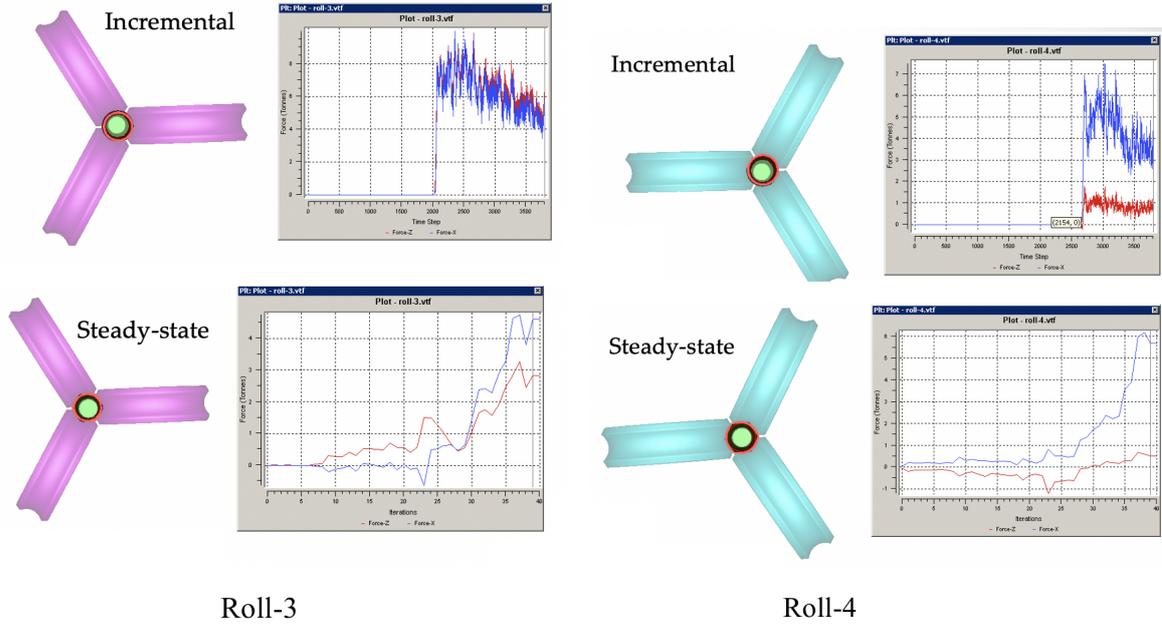


Figure 6.2.3: Tube Rolling test case forces F_x (red) F_z (blue) on roll stand 3 (left) and 4 (right) computed with incremental and steady-state algorithms

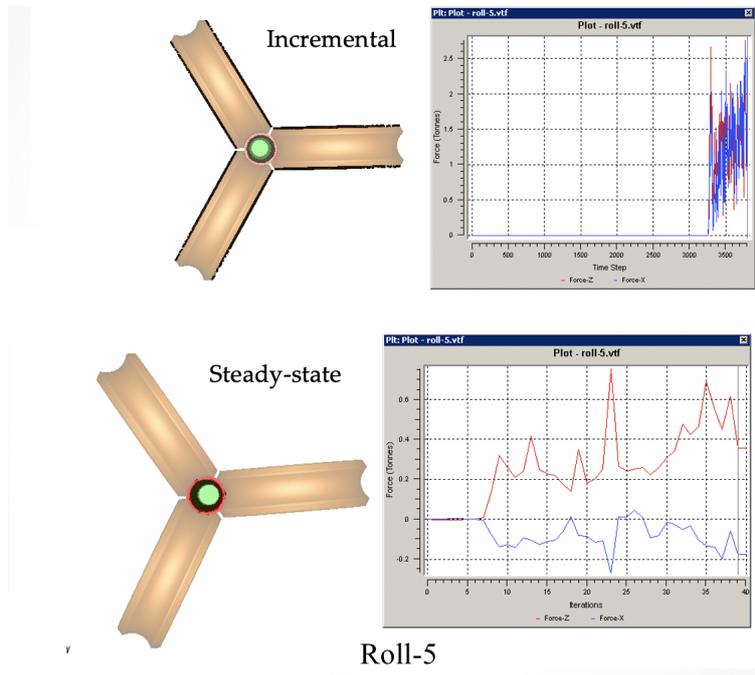


Figure 6.2.4: Tube Rolling test case forces F_x (red) F_z (blue) on roll stand 5 computed with incremental and steady-state algorithms

The steady-state algorithm fails to converge on this problem because the contact is asymmetric. Large oscillations were observed in the tool force in the incremental solution itself, which indicate that the process may not have an inherent steady-state associated with it. In fact, we can attain steady-state conditions in a stand N only if sufficient length of the workpiece has been rolled under stand $N+1$ (in the downwind). In other words, the force in the stand N is stabilized when the stand $N+1$ is full. An additional factor important for stability is the inter-stand tensions, which must be stable for reaching a steady-state. These necessary conditions to achieve steady-state may not be true in reality, which is also indicated in the incremental solution results. Hence, the

steady-state algorithm which principally works on the stabilized forces, geometry and state variables for convergence may not find a stable solution for such problems even playing around with fixed point relaxation. Nonetheless, the problem with asymmetric contact is still an important factor inducing instability and must be specifically treated.

6.2.2 Resolution approach for long interstand distances

Another bottleneck with the existing steady-state formulation is the resolution of the problem with long interstands. In such scenarios, it is really not possible to reduce the size of the model. The interstand length with the associated mesh must be modeled to simulate the different interstand phenomena like creep, thermal transfer, recrystallization. In such scenarios, the user loses the advantage of steady-state formulation. Hence, special consideration is necessary for these scenarios such that the interstand effects are modeled without actually modelling the complete interstand length of the workpiece.



Figure 6.2.5: Steady-state model necessary for modeling the interstand phenomena

6.2.3 Boundary condition at output plane for steady-state elastoplasticity resolution

A special consideration is also needed for the boundary condition at the output plane, which is not treated in the present work. The free surface condition at Γ_{out} is not realistic for the elastoplastic problems. The traction force at this surface is unknown which comes as an effect of internal stress acting on the boundary from the adjacent particles. However, these elements are not modeled and hence the effect is taken into account by adding internal force contributions from the fictitious neighboring elements in ω^* to the elements in ω [1] as shown in Figure 6.2.6.

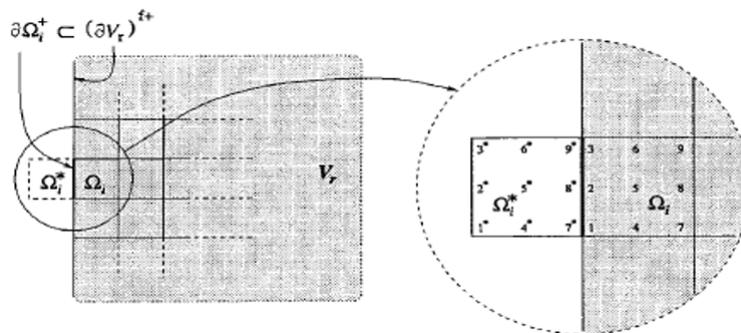


Figure 6.2.6: Accounting for the boundary condition at output plane Γ_{out} [1]

6.2.4 Speed-up of the steady-state elastoplasticity resolution

The main objective of the present PhD thesis was to construct and implement a functional algorithm for steady-state elastoplasticity resolution which is accurate. The presented algorithm meets both these objectives and also demonstrates a typical speedup of 10 times with the problems tested. However, the speed-up can further be improved with the following strategies:

Parallel implementation of streamline building and integration

The parallel implementation of the steady-state algorithm with viscoplasticity was possible as a finite element based framework was chosen for the resolution of the state variables and free surface. On the other hand, for elastoplasticity, the streamline based framework makes the parallel implementation challenging especially in the distributed memory parallel setting when the predecessor point is located in another partition. At present, the parallel implementation for streamline building, which is out of the scope of the PhD project, has not been completed. This restricts the simulations to be launched on a single processor, and upon completion, the computation time can be further reduced.

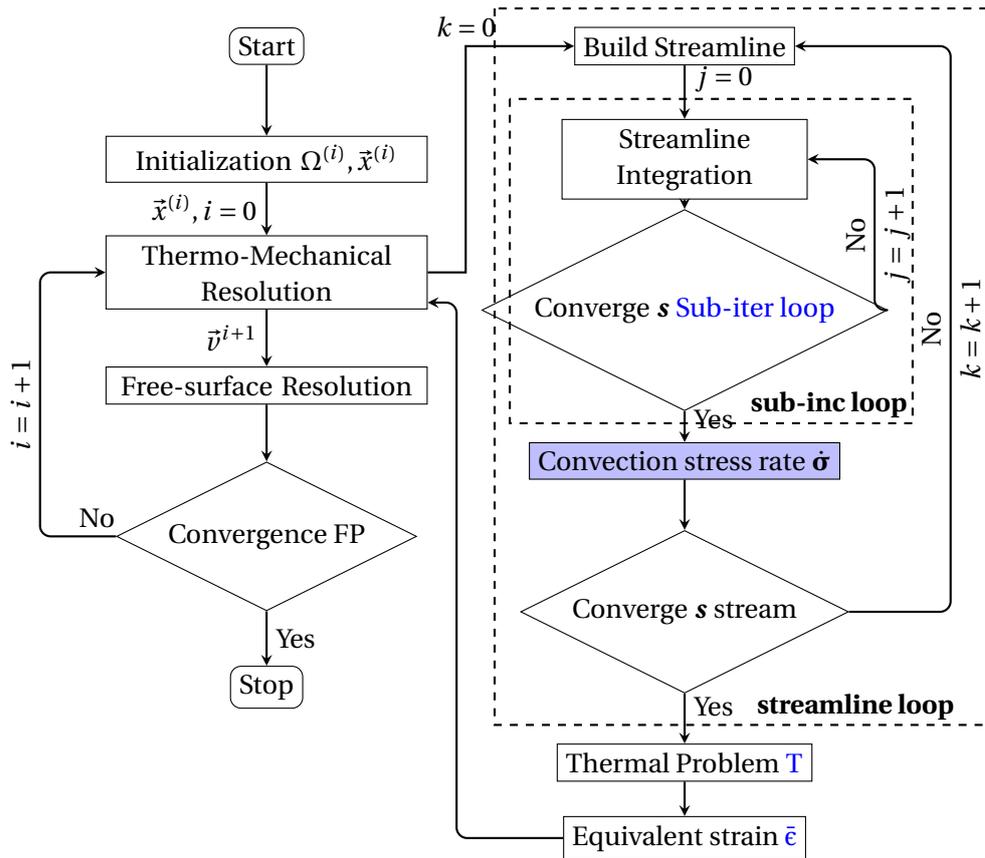


Figure 6.2.7: Algorithm-3 for EVP steady-state resolution with streamline loop in ForgeNxt® 2016

Speed-up with convection step

As seen in the algorithm proposed in Section 5.4.1, the streamline method provides the computation of the stress field and naturally takes care of the convection of the history-dependent variables in the downwind direction. This natural convection is however slow as the stability of the solution requires as many iterations as the number of integration points on the streamline. We had seen in the Section 2.3.5 that the global resolution (convection) method strategy is much faster in comparison to the streamline method for the integration of state-variables. However, in the case of elastoviscoplasticity, the global resolution requires the stress-rate (source) term, which is strongly coupled with velocity. Hence, the strongly coupled equation in stress and velocity must be computed until convergence in a fixed point iterative loop. This strategy is however not quite robust as the initial viscoplastic solution results in strong perturbation in the stress field which leads to perturbations in velocity. A converged solution is not always ensured. An alternative approach is to predict the stress-rate from the presented pseudo time-step approach with streamlines and use this estimate to speed up the convection with global resolution. In other words, as soon as a good estimation of the stress rate $\dot{\sigma}$ is available, the stress field $\bar{\sigma}$ can be computed from the reso-

lution of the following convection equation. This step figures in the resolution algorithm after the convergence of the sub-increment loop as shown in the Figure 6.2.7.

Find $\tilde{\boldsymbol{\sigma}} \in \mathbb{V}^d$ such that $\forall \tilde{\boldsymbol{\epsilon}}^* \in \mathbb{V}_0^d$

$$\int_{\Omega} (\vec{v} \cdot \vec{\nabla} \tilde{\boldsymbol{\sigma}}) \tilde{\boldsymbol{\sigma}}^* d\omega = \int_{\Omega} \dot{\boldsymbol{\sigma}} \tilde{\boldsymbol{\sigma}}^* d\omega \quad (6.2.1a)$$

$$\left\{ \begin{array}{l} \mathbb{V}^d = \{ \tilde{\boldsymbol{\sigma}} \in H^1(\Omega), \tilde{\boldsymbol{\sigma}} = \tilde{\boldsymbol{\sigma}}_{imp} \text{ on } \Gamma_e \} \\ \mathbb{V}_0^d = \{ \tilde{\boldsymbol{\sigma}} \in H^1(\Omega), \tilde{\boldsymbol{\sigma}} = 0 \text{ on } \Gamma_e \} \end{array} \right. \quad (6.2.1b)$$

It can be seen that the converged deviatoric stress and pressure from the is used to compute the stress-rate $\dot{\boldsymbol{\sigma}}$ after the [Sub-iter loop](#) which ensures the stability of stresses below the tool. The convection step ensures faster transport of state-variables and can further reduce the computation time.

From the industrial viewpoint, the present algorithm is robust enough and provides accurate solution. Among the presented perspectives the most important ones are the description of an accurate boundary condition at the output plane and the parallel implementation of the streamline building and integration of the state-variables.

Bibliography

- [1] D. Balagangadhar and D. A. Tortorelli. Design of large-deformation steady elastoplastic manufacturing processes. part i: a displacement-based reference frame formulation. *International Journal for Numerical Methods in Engineering*, 49(7):899–932, 2000.
- [2] A. Hacquin. *Modélisation Thermomécanique Tridimensionnelle du Laminage - Couplage Bande/ Cylindres*. PhD thesis, ENSMP, 1996.
- [3] A. Hacquin, P. Montmitonnet, and J.P. Guillerault. A steady state thermo-elastoviscoplastic finite element model of rolling with coupled thermo-elastic roll deformation. *Journal of Materials Processing Technology*, 60(1):109 – 116, 1996. ISSN 0924-0136.

Appendix A

Appendix

A.1 Finding the bubble shape function N_b

In the above-mentioned description, the shape function N_b is unknown. In order to find the shape function, we divide the reference 3D tetrahedral element with reference coordinates (ξ, η, ζ) into four sub-elements with reference coordinates (ξ', η', ζ') . The sub-elements are constructed from the three of the four corner nodes (1-2-3-4) of the parent element and its centroid (5) as shown in the Figure A.1.1. It can be seen that we can easily move from the coordinates of the centroid in the reference element to that of the sub-element in consideration. For doing so, we consider the reference element (1-2-3-4) below. For describing the piecewise centroid shape function, we find the coordinates of the centroid (5) in this element (1-2-3-4) in the sub-element (5-1-2-3), (5-1-2-4), (5-1-3-4), (5-2-3-4) (see Figure A.1.1).

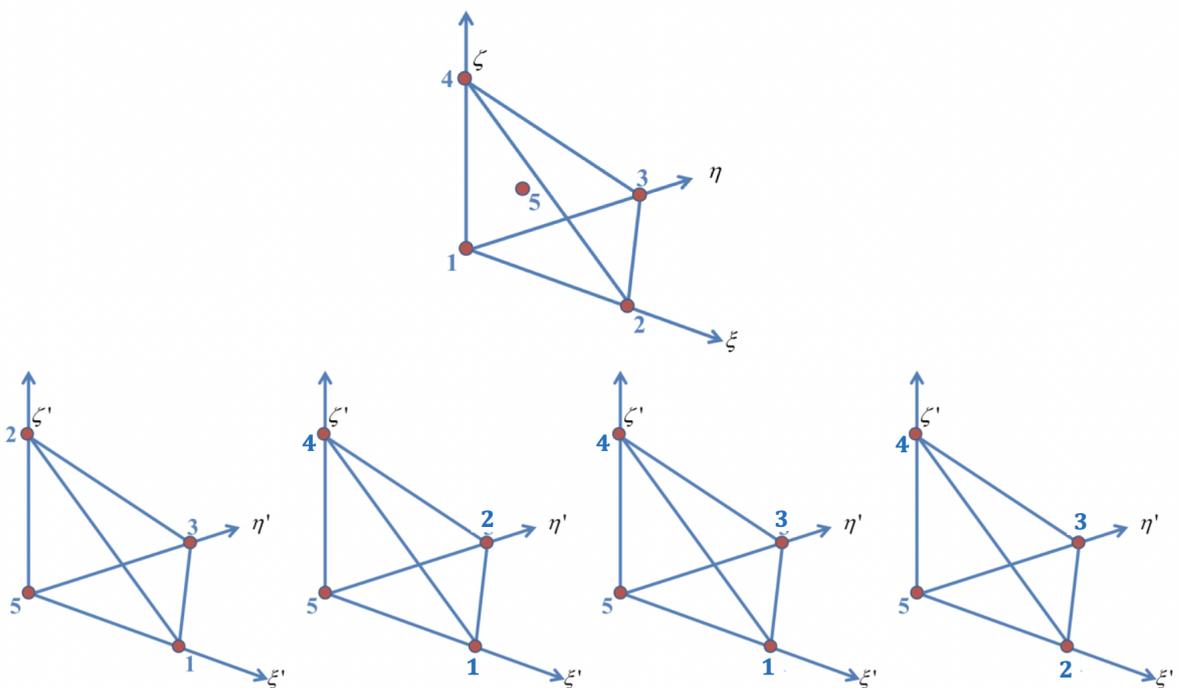


Figure A.1.1: Sub-elements inside the tetrahedral element

map (1-2-3-4) → (5-1-2-3) . Knowing that the vectors $\hat{u} = X_5\vec{X}_1 = \begin{pmatrix} -1/4 \\ -1/4 \\ -1/4 \end{pmatrix}$ $\hat{v} = X_5\vec{X}_2 = \begin{pmatrix} -1/4 \\ 3/4 \\ -1/4 \end{pmatrix}$

and $\hat{w} = X_5\vec{X}_3 = \begin{pmatrix} 3/4 \\ -1/4 \\ -1/4 \end{pmatrix}$ are the coordinates of the new reference frame (ξ', η', ζ') in the reference

element (ξ, η, ζ) , the rotation from reference frame $(X_1\vec{X}_2, X_1\vec{X}_3, X_1\vec{X}_4) = (\hat{i}, \hat{j}, \hat{k}) = \left(\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right)$

of reference element to the sub element $(X_5\vec{X}_1, X_5\vec{X}_2, X_5\vec{X}_3)$ is given as following. The system (of vectors) describe the matrix M^{-1} to define the map from $(\hat{i}, \hat{j}, \hat{k}) \rightarrow (\hat{u}, \hat{v}, \hat{w})$.

$$M^{-1} = (\hat{u}, \hat{v}, \hat{w}) = \left(\begin{pmatrix} -1/4 \\ -1/4 \\ -1/4 \end{pmatrix}, \begin{pmatrix} -1/4 \\ 3/4 \\ -1/4 \end{pmatrix}, \begin{pmatrix} 3/4 \\ -1/4 \\ -1/4 \end{pmatrix} \right) \quad (A.1.1)$$

However, we are interested in the inverse map matrix M from $(\hat{u}, \hat{v}, \hat{w}) \rightarrow (\hat{i}, \hat{j}, \hat{k})$

$$M = \begin{bmatrix} -1 & -1 & -2 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad (A.1.2)$$

So the transformation from (ξ, η, ζ) to (ξ', η', ζ') is written as:

$$\begin{pmatrix} \xi' \\ \eta' \\ \zeta' \end{pmatrix} = M \begin{pmatrix} \xi - 1/4 \\ \eta - 1/4 \\ \zeta - 1/4 \end{pmatrix} = \begin{bmatrix} -1 & -1 & -2 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad (A.1.3)$$

From this map, we can verify that $\varsigma(X_5) = \varsigma \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$, $\varsigma(X_1) = \varsigma \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, $\varsigma(X_2) = \varsigma \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

and $\varsigma(X_3) = \varsigma \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$. In a similar manner, we can find the map for the other sub elements as following:

map (1-2-3-4) → (5-1-2-4)

$$\begin{pmatrix} \xi' \\ \eta' \\ \zeta' \end{pmatrix} = M \begin{pmatrix} \xi - 1/4 \\ \eta - 1/4 \\ \zeta - 1/4 \end{pmatrix} = \begin{bmatrix} -1 & -2 & -1 \\ 1 & -1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad (A.1.4)$$

map (1-2-3-4) → (5-1-3-4)

$$\begin{pmatrix} \xi' \\ \eta' \\ \zeta' \end{pmatrix} = M \begin{pmatrix} \xi - 1/4 \\ \eta - 1/4 \\ \zeta - 1/4 \end{pmatrix} = \begin{bmatrix} -2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad (A.1.5)$$

map (1-2-3-4) → (5-2-3-4)

$$\begin{pmatrix} \xi' \\ \eta' \\ \zeta' \end{pmatrix} = M \begin{pmatrix} \xi - 1/4 \\ \eta - 1/4 \\ \zeta - 1/4 \end{pmatrix} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad (A.1.6)$$

For knowing if a point $P(\xi, \eta, \zeta)$ (or else $P(\xi', \eta', \zeta')$) belongs to the sub-tetra element we use the distance function $\bar{\delta}$ in 3D:

$$\bar{\delta} = |\xi'| + |\eta'| + |\zeta'| + |1 - \xi' - \eta' - \zeta'| - 1 \quad (A.1.7)$$

List of acronyms

ALE Arbitrarily Lagrangian Eulerian.

DoF Degrees of freedom.

EP Elastoplastic.

EVP Elasto-viscoplastic.

EVP-Test-Case-01 Cold rolling of a thick flat rectangular section 2D plate.

EVP-Test-Case-02 Cold rolling of flat rectangular section 3D plate.

EVP-Test-Case-03 Cold rolling of a long product.

Fg3[®] Transvalor S.A. ForgeNxt[®].

ForgeNxt[®]2016 ForgeNxt[®] steady-state solver, released version.

ForgeNxt[®]inc ForgeNxt[®] incremental solver.

HPTS Heterogeneous pseudo-time-step.

HSM Hertz–Signorini–Moreau.

KKT Karush–Kuhn–Tucker.

LS-SUPG Least-square Streamline Upwind Petrov-Galerkin.

PG Petrov-Galerkin.

SPR Superconvergent Patch Recovery.

Steady-State Explicit ForgeNxt[®] steady-state solver with explicit contact constraint.

Steady-State Implicit ForgeNxt[®] steady-state solver with implicit contact constraint.

Step-1 Thermo-Mechanical Resolution.

Step-2 Free-surface Resolution.

Stream-loop Streamline loop for **EVP** resolution.

Sub-iter loop Sub-iterative loop for **EVP** resolution.

SUPG Streamline Upwind Petrov Galerkin.

SUPG-dif Free-surface correction with 2-**DoF** formulation.

SUPG-NC Nodal form of free-surface equation.

SUPG-NC-dif Nodal form of free-surface equation with 2-DoF formulation.

SUPG-reg Free-surface correction with surface regularization.

TL Total Lagrangian formulation.

UL Updated Lagrangian formulation.

VP Viscoplastic.

VP-Test-Case-01 Hot rolling of a thick flat rectangular section plate.

VP-Test-Case-02 Hot rolling of a long product from oval to square section.

VP-Test-Case-03 Hot rolling of a long product from square to rectangular section.

List of symbols

- $\mathbf{1}$ Identity matrix .
- α SUPG stabilization function.
- α_f Viscoplastic friction coefficient.
- β Thermal hardening sensitivity.
- \mathbb{C} Elasticity compliance fourth-order tensor.
- δ Gap function.
- \mathbf{e} Deviatoric strain tensor.
- e Numerical pseudo-adhesion for the contact surface.
- ϵ_0 Regularisation term.
- $\bar{\epsilon}$ Equivalent strain.
- $\dot{\bar{\epsilon}}$ Equivalent strain-rate/rate of deformation.
- $\bar{\epsilon}_{imp}$ Imposed equivalent strain at input plane.
- \mathbf{e} Strain tensor.
- $\dot{\mathbf{e}}$ strain-rate/rate of deformation tensor.
- $\dot{\mathbf{e}}^e$ Elastic strain-rate/rate of deformation tensor.
- $\dot{\mathbf{e}}^p$ Plastic strain-rate/rate of deformation tensor.
- $\dot{\mathbf{e}}^{vp}$ viscoplastic strain-rate/rate of deformation tensor.
- \mathbf{e}^e Elastic strain tensor.
- \mathbf{e}^p Plastic strain tensor.
- ϵ_r Emissivity.
- Γ Free surface boundary.
- Γ_c Contact surface.
- Γ_{edge} Free surface edge.
- Γ_{in} Input plane.
- Γ_{out} Output plane.
- Γ_{sym} Symmetry plane.

- ϕ Material-Spatial configuration map.
- ι constitutive model internal variable.
- λ Lagrangian multiplier (Mechanical).
- λ^e Lamé's coefficient 1.
- $\tilde{\lambda}$ New Lagrangian multiplier (Mechanical).
- λ^{pl} Plasticity multiplier.
- μ Material viscosity.
- μ^e Lamé's coefficient 2.
- μ_f Coulomb friction coefficient.
- μ_k Lagrangian multiplier (Free surface correction).
- \vec{V} Displacement field.
- ν Poisson's ratio.
- Ω_x Current configuration.
- ω Computation domain differential.
- Φ Mechanical problem functional.
- Φ_{LS} Free surface correction functional.
- Φ_p Extended mechanical potential with penalty constraint.
- Φ^{vp} Viscoplastic potential.
- ϕ_{imp} Imposed thermal flux at the input plane Γ_{in} .
- ρ Material Density.
- $\rho_c^{(1)}$ Penalty coefficient for contact term in Mechanical Problem.
- $\rho_c^{(2)}$ Penalty coefficient for contact term in Free Surface Correction Problem.
- ρ_c Penalty coefficient for contact term in Mechanical Problem.
- $\bar{\sigma}$ von Mises stress.
- $\boldsymbol{\sigma}$ Cauchy's stress tensor.
- $\dot{\boldsymbol{\sigma}}$ Stress-rate tensor.
- σ_n Contact normal stress.
- σ_r Stefan–Boltzmann constant.
- σ_y Yield stress.
- $\vec{\tau}_f$ Tangential friction stress.
- tr Trace of the matrix.
- b_m Workpiece thermal effusivities.

b Averaged thermal effusivities.

b_{tool} Tool thermal effusivities.

\vec{b} Body force.

\mathbb{C}^e Elasticity compliance fourth-order tensor.

C_k SUPG cosinus.

c_p specific heat capacity.

\vec{d} Shape correction direction.

E Elasticity Modulus.

$\dot{\mathbf{e}}$ Deviatoric strain-rate/rate of deformation tensor.

$\dot{\mathbf{e}}^e$ Elastic deviatoric strain-rate/rate of deformation tensor.

$\dot{\mathbf{e}}^p$ Plastic deviatoric strain-rate/rate of deformation tensor.

\mathbf{e}^e Deviatoric elastic strain tensor.

\mathbf{e}^p Deviatoric plastic strain tensor.

\mathbf{F} Deformation gradient tensor.

f Surface element.

\mathbf{f} History-dependent field variable.

g_f Tresca scalar.

\tilde{h} Free surface correction impenetrability constraint for steady-state formulation.

h_{cond} Conductivity transfer coefficient.

h_{conv} Convection transfer coefficient.

h_e characteristic element length.

h Mechanical contact constraint for steady-state formulation.

\mathbf{i} Previous iteration (fixed-point algorithm).

$\mathbf{i+1}$ Current iteration (fixed-point algorithm).

\mathbf{int} Integration point.

$\mathbf{int-1}$ Upwind integration point.

\mathbf{J} Jacobian.

\mathbf{K} Material consistency.

k Reference node index.

\mathbf{K}_0 Reference consistency.

- K_b Bulk Modulus.
- K_{cond} Thermal conductivity.
- K_f Viscoplastic friction consistency.
- $L_p(\vec{U}, \Lambda)$ Extended mechanical potential with penalty constraint.
- $L(\vec{u}, \lambda)$ Lagrangian functional.
- m Material sensitivity to strain-rate/rate of deformation.
- n Strain hardening sensitivity.
- nbe** Number of elements in the mesh.
- nbn** Number of nodes in the mesh.
- N_e Linear bubble interpolation shape function corresponding to node e .
- N_k Linear Galerkin interpolation shape function at node k .
- $N_k^{LS-SUPG}$ LS-SUPG shape function at node k .
- N_k^{SUPG} SUPG shape function at node k .
- \vec{n}^{tool} Unit tool normal.
- \vec{n} Surface mesh unit normal.
- \vec{n}^f Surface mesh unit normal at facet f .
- \vec{n}^{sym} Unit normal at symmetry plane.
- \mathbf{P}** First Piola-Kirchoff's Stress Tensor.
- p Pressure field.
- \dot{p} Pressure-rate field.
- \mathcal{P}_k Patch of elements about node k .
- P_{pl} Plastic power dissipated into heat.
- pre** Upwind projection of the integration point int .
- q Slipping sensitivity index.
- \vec{q} Heat flux.
- R_X Material frame.
- \mathbf{S}** Second Piola-Kirchoff's Stress Tensor.
- \mathbf{s}** Deviatoric stress tensor.
- $\dot{\mathbf{s}}$** Deviatoric stress-rate tensor.
- S_k Free surface correction residual.

- s_k Nodal surface area.
- \tilde{S}_k Free surface correction residual (with [Nodal form of free-surface equation \(SUPG-NC\)](#)).
- t_0 Initial time.
- t Previous time.
- Δt Time increment.
- $t + \Delta t$ Current time.
- T_{ext} Ambient Temperature.
- T_{imp} Imposed temperature field at input plane Γ_{in} .
- \tilde{T}_{IS} Interstand tension.
- t Shape correction (scalar).
- T_{tool} Tool temperature field.
- \vec{t} Shape correction vector.
- \vec{u}^{tool} Tool Displacement.
- \vec{U}_{km} Surface normal matrix.
- \vec{u}_k Upwind normal at node k .
- $\delta\vec{u}_k$ Change of the upwind normal at node k .
- \hat{u}_k Unit upwind normal at node k .
- \vec{u} Displacement field.
- $\Delta\vec{u}$ Differential slip between two contact surfaces.
- \vec{u}^f Surface mesh normal at facet f .
- \vec{u}_n Normal displacement field.
- $\Delta\vec{u}_t$ Differential tangential slip between two contact surfaces.
- \hat{w} Mesh Velocity.
- $\mathbb{V}_0^{\vec{v}}$ Functional space for admissible kinematic velocities .
- \vec{v}_e Characteristic element velocity.
- \mathbb{V}^P Functional space for admissible pressure .
- \vec{v} Velocity field.
- $\mathbb{V}^{\vec{v},b}$ Bubble functional space for admissible kinematic velocities.
- \vec{v}^b Bubble velocity field.
- $\Delta\vec{v}$ Relative velocity wrt tool.
- $\mathbb{V}^{\vec{v},l}$ Linear functional space for admissible kinematic velocities.
- \vec{v}^l Linear velocity field.

\vec{v}_n Normal velocity.

\vec{v}_t Tangential velocity.

$\Delta\vec{v}_t$ Relative tangential velocity wrt tool.

\vec{v}^{tool} Tool Velocity.

$\mathbb{V}^{\vec{v}}$ Functional space for admissible kinematic velocities.

\mathbf{X} Material coordinates.

\mathbf{X}' Hardening back stress.

$\vec{x}^{(i)}$ Position vector at previous configuration.

$\vec{x}^{(i+1)}$ Position vector at previous configuration.

RÉSUMÉ

La formulation [ForgeNxt®2016](#) existante a été proposée pour la résolution des procédés de mise en forme à chaud en régime permanent avec des matériaux viscoplastiques. La formulation comprend un problème multi-champs fortement couplé et résolu avec une méthode itérative à point-fixé. L'algorithme itératif comporte deux étapes principales: (i) un solveur thermomécanique calcule d'abord les champs inconnus *vitesse / pression / température* sur un volume de contrôle (ii) un solveur à surface libre calcule ensuite l'inconnu *forme stationnaire*. Le [ForgeNxt®2016](#), avec des maillages non structurés basés sur des éléments tétraédriques et une parallélisation avec partitionnement de domaine, s'avère au moins 50 fois plus rapide que l'approche incrémentale pour résoudre ces procédés. Cependant, la formulation n'est pas suffisamment robuste pour trouver une solution stable, en particulier avec des géométries complexes. Le travail présenté dans ce manuscrit est axé sur l'amélioration de la formulation en régime permanent avec deux objectifs principaux. Premièrement, un couplage de contact cohérent est prévu pour une solution stable, et est obtenu avec une approche *condensation nodale*. Les conditions de contact cohérentes *Explicit* et *Implicit* sont dérivées et le nouvel couplage de contact est testé avec des cas tests de laminage à chaud industriels complexes. Deuxièmement, la formulation originale [ForgeNxt®2016](#) était principalement axée uniquement sur les problèmes de mise en forme à chaud avec des modèles de matériaux viscoplastiques qui ne prennent pas en compte les effets d'élasticité. Ces effets deviennent importants en froid et ne peuvent être ignorés. Une approche *pseudo-time-step* permet de modéliser le temps dans la formulation indépendante du temps et est facilement adaptable à l'algorithme itératif [ForgeNxt®2016](#) existant. Bien qu'à l'origine, cette approche utilisait des maillages structurés pour suivre les lignes de courant, un nouveau cadre est développé pour tracer les lignes de courant dans les maillages non structurés et pour intégrer les variables d'état pour résoudre le problème dépendant de l'historique. À chaque itération, les variables d'état doivent être transportées des lignes de courant vers le maillage et vice-versa, qui est généralement diffusif. Pour limiter la diffusion, des outils comme la méthode [SPR](#) pour le lissage de champ et l'interpolation P1+ sont invoqués. L'algorithme [ForgeNxt®2016](#) mis à jour est testé avec différents problèmes de laminage à froid. Les résultats des simulations de laminage à chaud et à froid avec l'algorithme proposé sont validés avec la solution incrémentale (dépendante du temps) du même problème dans [ForgeNxt®inc.](#)

MOTS CLÉS

modélisation stationnaire, formulation de contact, formulation multi-champs, laminage à froid

ABSTRACT

The existing [ForgeNxt®2016](#) formulation was proposed for the resolution of the steady-state hot forming processes with viscoplastic materials. The formulation comprises of a strongly-coupled multi-field problem and solved with a fixed-point iterative method. There are two main steps in the iterative algorithm: (i) a thermo-mechanical solver firstly computes the unknown *velocity/pressure/Temperature* field on a control volume (ii) a free-surface solver then computes the unknown *steady-state shape*. The [ForgeNxt®2016](#), with unstructured meshes based on tetrahedral elements and parallelization with domain partitioning, is found to be at least 50 times faster than the traditional incremental approach for solving these processes. However, the formulation is not robust enough to find a stable solution especially with complex geometries. The work presented in this manuscript is focused on the improvement of the steady-state formulation with two main objectives. Firstly, a consistent contact-coupling is anticipated for a stable solution, and is achieved with a *nodal condensation* approach. The *Explicit* and *Implicit* consistent contact conditions are derived and the updated contact-coupling is tested with complex industrial hot-rolling test cases. Secondly, the original [ForgeNxt®2016](#) formulation was mainly focused only on hot forming problems with viscoplastic material models which do not consider elasticity effects. These effects become prominent in cold conditions and cannot be ignored. A *pseudo-time-step approach* makes possible to model the time in the time-independent formulation and is easily adaptable to the existing [ForgeNxt®2016](#) iterative algorithm. Though originally this approach used structured meshes for tracking streamlines, a new framework is developed to trace the streamlines in the unstructured meshes and to integrate the state-variables for solving the history-dependent problem. In each iteration, the state-variables must be transported from the streamlines to the mesh and vice-versa, which is generally diffusive. To restrict the diffusion, tools like [SPR](#) method for field smoothening and P1+ interpolation are invoked. The updated [ForgeNxt®2016](#) algorithm is tested with different cold rolling problems. The results from the hot and cold rolling simulations with the proposed algorithm are validated with the incremental (time-dependent) solution of the same problem in [ForgeNxt®inc.](#)

KEYWORDS

steady-state modeling, contact formulation, multi-field formulation, cold rolling