

# Contact analysis and overlapping domain decomposition methods for dynamic and nonlinear problems

Von der Fakultät Mathematik und Physik der Universität Stuttgart  
zur Erlangung der Würde eines Doktors der  
Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von  
**Stephan Brunßen**  
aus Varel i.O.

Hauptberichter:	Prof. Dr. B. Wohlmuth
Mitberichter:	Prof. Dr. C. Wieners Prof. Dr. M. Schäfer

Tag der mündlichen Prüfung: 4. April 2008

Institut für Angewandte Analysis und Numerische Simulation  
Universität Stuttgart  
2008



# Acknowledgments

This thesis summarizes the results of my research activities during the years 2003 to 2008 at the chair ‘Numerische Mathematik für Höchstleistungsrechner’ of the Institut für Angewandte Analysis und Numerische Simulation (IANS), Universität Stuttgart.

The research work was funded by the German Research Foundation (DFG) which is gratefully acknowledged.

I would like to express my extraordinary gratitude to my supervisor and academic teacher Prof. Dr. Barbara Wohlmuth for her mentoring, support and enduring encouragement of my work. Her outstanding dedication to the research activities of the whole group has been truly inspiring for me. I would also like to thank her for giving me the opportunity to present my work on international conferences.

My special thanks go to Prof. Dr. Christian Wieners and Prof. Dr. Michael Schäfer for their efforts to provide the referee reports for this thesis. I am very grateful to Prof. Dr. Barbara Kaltenbacher for participating in the oral defense. I also would like to thank the scientists with whom I had the pleasure to co-author several contributions: Dr. Markus Bambach, Prof. Dr. Gerhard Hirt and in particular Prof. Dr. Ekkehard Ramm and again Prof. Dr. Michael Schäfer.

I thank Dr. Florian Schmid and Dr. Stefan Hartmann not only for writing publications with me but also for their close and very pleasant collaboration.

I am much obliged to the supervisors of my Diploma thesis Dr. Henning Behnke and Prof. Dr. Ulrich Mertins for arousing my interest in the Finite Element method.

Thanks to Dr. Paul Lyons and his co-workers for most stimulating discussions about contact problems in an industrial context and also to Dr. Uwe-Jens Görke for his help on finite plasticity. Thanks also to the LASSO GmbH for giving software technical support. During my employment at the IANS, I greatly enjoyed the inspiring and cooperative atmosphere. I thank Dr. Andreas Klimke, Dr. Bishnu Lamichhane, Michael Mair, Brit Steiner and Alexander Weiß for always being available and offering help and advice for accomplishing the various theoretical, technical and administrative tasks connected with this thesis. I would especially like to thank Corinna Hager and Dr. Bernd Flemisch for their cooperation in domain decomposition methods and for proofreading parts of my thesis and Stefan Hieber for his help on contact problems.

I am deeply indebted to my family and especially to my parents for always supporting me with all their strength and goodwill. Finally I would like to express my deepest gratitude to my wife Haixia for her constant moral support and encouragement. Without her love and patience, it would not have been possible to complete this work.

Stuttgart, April 2008

Stephan Brunßen



# Contents

<b>Abstract</b>	<b>ix</b>
<b>Zusammenfassung</b>	<b>xi</b>
<b>Notations</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Deep-rolling . . . . .	1
1.2 Incremental sheet metal forming . . . . .	2
1.3 Aim and outline of this work . . . . .	3
<b>2 Classical 1-grid discretization: Plasticity</b>	<b>7</b>
2.1 Introduction . . . . .	7
2.2 Governing equations and notations . . . . .	7
2.3 Global equilibrium iteration and Finite Element discretization . . . . .	8
2.4 A Gauss point section: Radial Return for linear isotropic hardening . . . . .	10
2.4.1 Consistent elastoplastic tangent module . . . . .	12
2.4.2 Plastic consistency parameter . . . . .	13
2.5 A Gauss point section: The Tangent Cutting Plane algorithm . . . . .	14
2.5.1 Fulfillment of the yield condition . . . . .	16
2.5.2 Discussion . . . . .	17
2.6 A Gauss point section: Sequential quadratic programming . . . . .	17
2.6.1 Definitions . . . . .	17
2.6.2 Derivation of the SQP step . . . . .	18
2.6.3 The SQP step and the algorithm . . . . .	18
2.7 Comparison of the Gauss point algorithms . . . . .	20
2.8 Numerical results . . . . .	21
2.9 Some tensor calculus . . . . .	23
2.10 Summary . . . . .	23
<b>3 Classical 1-grid discretization: Contact</b>	<b>25</b>
3.1 Contact algorithms in the literature . . . . .	25
3.2 The primal dual active set strategy with dual Lagrange multipliers . . . . .	27
3.2.1 Basic idea . . . . .	27
3.2.2 Continuous equations . . . . .	28
3.2.3 Active Set Strategy and Algebraic Multigrid Methods . . . . .	31
3.2.4 Discretized form . . . . .	32

3.2.5	Nonlinear algorithm for 1-grid discretization . . . . .	33
3.2.6	Inexact strategy . . . . .	36
3.3	Numerical results for contact . . . . .	36
3.3.1	Block indentation test . . . . .	36
3.3.2	Comparison of both contact approaches in connection with AMG . . . . .	37
3.3.3	Active set versus adaptive penalty strategies . . . . .	38
3.3.4	Acceleration by AMG-inexact strategy . . . . .	40
3.4	Summary . . . . .	42
<b>4</b>	<b>Dynamic contact</b>	<b>43</b>
4.1	Introduction . . . . .	43
4.2	Problem description . . . . .	44
4.2.1	One-body contact with rigid obstacle . . . . .	44
4.2.2	Boundary value problem of nonlinear elastodynamics . . . . .	46
4.3	Spatial discretization of contact virtual work . . . . .	46
4.3.1	Discrete dual Lagrange multipliers for arbitrary shaped elements . . . . .	47
4.3.2	Semidiscrete initial value problem . . . . .	49
4.4	Time discretization . . . . .	50
4.4.1	Linearization and iterative solution strategy . . . . .	51
4.4.2	Global system with contact manipulation . . . . .	52
4.5	Spatial discretization by a finite shell element . . . . .	53
4.6	Energy conservation for frictionless contact . . . . .	53
4.7	Examples . . . . .	56
4.7.1	Weak non-penetration . . . . .	57
4.7.2	Toss rule . . . . .	57
4.7.3	Ball . . . . .	60
4.7.4	Torus . . . . .	61
4.8	Conclusions . . . . .	62
<b>5</b>	<b>Velocity based contact problems</b>	<b>65</b>
<b>6</b>	<b>Coupling algorithm</b>	<b>71</b>
6.1	Dynamic ODDM, main concepts . . . . .	71
6.2	Review of the literature . . . . .	73
6.3	Coupling scheme . . . . .	74
6.3.1	Fine domain . . . . .	75
6.3.2	Coarse domain . . . . .	75
6.3.3	Solving the nonlinear problem with a coarse global and a fine local mesh . . . . .	77
6.4	Numerical results . . . . .	79
6.4.1	Material data . . . . .	79
6.4.2	Elastic example, non-hierarchical case . . . . .	79
6.4.3	Numerical testing of the mortar operator . . . . .	80
6.4.4	Relative error . . . . .	84

6.4.5	Update strategies . . . . .	86
6.4.6	Deep-rolling like example . . . . .	87
6.5	Conclusions and future work . . . . .	87
<b>7</b>	<b>An inexact Newton block scheme</b>	<b>91</b>
7.1	Motivation . . . . .	91
7.2	Block iterative scheme: Linear elasticity . . . . .	91
7.3	Newton Block iterative scheme: Elasticity with contact . . . . .	93
7.4	Newton Block iterative scheme: Plasticity with contact . . . . .	94
7.5	Block iterative schemes . . . . .	96
7.6	Schur complement method . . . . .	96
7.6.1	Schur complement method, elastic case . . . . .	96
7.6.2	Schur complement method, plastic case . . . . .	98
7.7	Inexact Newton methods . . . . .	99
7.7.1	Avoiding oversolving . . . . .	99
7.7.2	Iterative solver . . . . .	99
7.8	Complete algorithm . . . . .	101
7.9	Numerical results . . . . .	101
7.9.1	Model example . . . . .	101
7.9.2	Schur complement method . . . . .	102
7.9.3	Block Gauss Seidel . . . . .	102
7.10	Summary . . . . .	103
<b>8</b>	<b>Large deformations</b>	<b>105</b>
8.1	Definitions . . . . .	105
8.1.1	Stress and strain tensors . . . . .	106
8.1.2	Multiplicative decomposition . . . . .	106
8.1.3	Applications of pull-back and push-forward . . . . .	107
8.1.4	Additive decomposition of Green's strain . . . . .	107
8.2	Equilibrium, Total Lagrange . . . . .	108
8.3	Constitutive law . . . . .	110
8.4	Consistent linearization of the constitutive law . . . . .	111
8.5	A Gauss point section: Generalized projection method . . . . .	112
8.6	Updated Lagrange . . . . .	113
8.6.1	Some iteration for UL . . . . .	115
8.7	Contact Total Lagrange . . . . .	116
8.7.1	Transformation of the contact stresses . . . . .	116
8.7.2	Mortar coupling fine to coarse in the large deformation regime . .	117
8.8	Conclusions . . . . .	118
<b>9</b>	<b>Conclusions</b>	<b>119</b>
	<b>Bibliography</b>	<b>121</b>





# Abstract

This thesis is concerned with the development of efficient numerical schemes for the Finite Element simulation of elastoplastic incremental metal forming processes. Two examples of this new and promising manufacturing technology are introduced to motivate the research work. Some basic technology is provided to accelerate the implicit Finite Element simulation which is still very costly for this kind of operations due to the small but very mobile forming zone and due to the highly nonlinear field equations and inequalities to be solved. For this purpose, the underlying equations and inequalities are reviewed. The main idea to meet these challenges is to use a ‘divide and conquer’ approach: The work-piece is discretized with a global coarse mesh and the forming zone is meshed with a small fine grid. Unlike in adaptive Finite Elements, no sophisticated remeshing procedures are necessary and the two grids are computationally independent from each other. The interface between coarse and fine computation is small such that a block iterative solution with two different Finite Element programs is possible. To hide the nonlinearities from the global computation, the two meshes interchange information about the plastic deformation and about the contact stresses. Results and algorithms from several disciplines of numerical mathematics and computational mechanics (contact, domain decomposition, iterative solvers, plasticity etc.) are combined to accomplish this task.



# Zusammenfassung

Durch die CNC-gesteuerte inkrementelle Umformtechnik können durch wiederholte Einwirkung einfacher Werkzeuge kostengünstig Bauteile mit komplexen Geometrien hergestellt werden. Wichtige Eigenschaften dieses neuen Fertigungsverfahrens sind die hohe Flexibilität und der geringe Kraftbedarf. Neue Umformmaschinen, Materialien und Steuerungstechniken haben zu einer Technologie geführt, die eine interessante Alternative zu klassischen Umformverfahren, wie z.B. dem Tiefziehen, bieten. Aufgrund der Vielzahl an Größen, die auf Umformprozesse Einfluss nehmen, seitens des Materials, seitens vielerlei geometrischer Nebenbedingungen usw. stellt sich die Frage nach optimalen Umformstrategien. Daher besteht ein großer Bedarf nach einer effizienten Modellierung und Simulation dieser Vorgänge. Um implizite Finite Element Pakete für die Simulation inkrementeller Umformungen auf industrieller Skala nutzbar zu machen, müssen die derzeit noch sehr langen Rechenzeiten dramatisch verkürzt werden.

Im einleitenden Kapitel 1 werden zwei Beispiele für inkrementelle Umformung eingeführt, um die Forschungsarbeit zu motivieren. Die beschriebenen Verfahren sind grundsätzlich sehr schwer zu steuern. Durch das inkrementelle Vorgehen addieren sich Verfahrensfehler sehr schnell auf und führen nach und nach zu immer mehr Abweichung von der Sollkontur des Werkstückes. Ziel dieser Arbeit ist es, die implizite Finite Element Simulation solcher Prozesse zu verschnellern, so dass es in naher Zukunft möglich sein wird, Werkzeugpfade und andere Prozessparameter per Simulation zu testen und gegebenenfalls 'online' abzuändern. Gegenwärtiger Stand der Technik bei den gängigen Softwarepaketen ist es immer noch, mit expliziter Zeitintegration zu arbeiten und die Rechnung durch Techniken wie zum Beispiel Massenskalierung drastisch zu beschleunigen. Es ist allerdings bekannt, dass derartige Methoden insbesondere in den Spannungen zu sehr ungenauen Ergebnissen führen, spätestens sobald dynamische Effekte die Lösung dominieren. Andererseits enthält die implizite Finite Elemente Simulation zahlreiche Herausforderungen. Die Systemmatrizen sind häufig schlecht konditioniert, unter anderem durch die Nutzung eines Penaltyparameters für den Kontakt. Daraus resultieren Schwierigkeiten bei der dringend nötigen Verwendung von iterativen Lösern.

Die Umformzone ist sehr klein aber auch sehr mobil, da das Werkzeug fast jeden Oberflächenpunkt des Werkstückes im Verlaufe des Vorgangs mindestens einmal berührt. Daher müsste bei einer nicht-adaptiven Rechnung die Anzahl der Elemente sehr groß sein. Auch die Anzahl der Zeitschritte ist sehr groß, da allein die Echtzeitdauer mindestens mehrere Minuten beträgt. Die Grundidee, diese Herausforderungen anzugehen, ist im wesentlichen reduzierbar auf das Motto 'Teile und herrsche': Eine dynamische überlappende Gebietszerlegungsmethode wird dergestalt angewendet, dass das Bauteil mit einem globalen,

groben Gitter vernetzt wird und die Umformzone mit einem davon weitestgehend unabhängigen lokalen, feinen Gitter, siehe Kapitel 6. In den Kapiteln 2 und 3 wird ein kurzer Überblick über den klassischen Lösungsprozess mit Finiten Elementen ohne Adaptivität gegeben. Desweiteren werden Definitionen und Schreibweisen eingeführt, die für die Beschreibung des Informationsaustausches zwischen den beiden Gittern gebraucht werden. Die wichtigsten Aspekte dieser Kopplung, nämlich die elastoplastische und die Kontaktkopplung werden in Kapitel 6 behandelt, und ein erster Ansatz für eine blockweise Lösung wird vorgestellt.

In Kapitel 7 wird dieser Ansatz verbessert und mit dem klassischen inexakten Newtonverfahren in Einklang gebracht.

Grundsätzlich besteht der Wunsch, Metallumformsimulationen dynamisch also auch mit Trägheitseffekten durchzuführen. Hierfür ist eine energiekonsistente Zeitintegration des Kontaktproblems eine wichtige Voraussetzung. Dies wird in Kapitel 4 im Rahmen der Behandlung dünnwandiger Strukturen behandelt.

Komplexe Linearisierungen ergeben sich durch verschiebungsbasierte (Kapitel 3) und geschwindigkeitsbasierte (Kapitel 5) Kontaktprobleme, durch ein elastoplastisches Materialgesetz (Kapitel 2) und durch große Deformationen (Kapitel 8).

In Kapitel 9 werden abschließende Bemerkungen gemacht.

# Notations

The following notations and abbreviations will be used in this thesis repeatedly.

## Abbreviations, alphabetical

AMG	Algebraic Multigrid
(pd)ASeS	(primal dual) active set strategy
CAD	computer aided design
CC	current configuration
CNC	computerized numerical control
dof	degree(s) of freedom
FEM	finite element method
GEMM	Generalized energy momentum method
Gen- $\alpha$	Generalized $\alpha$ method
GP	Gauss point
GS	Gauss Seidel
IBF	Institut für Bildsame Formgebung
IC	intermediate configuration
ISF	incremental sheet metal forming
lhs	left hand side
LM	Lagrange multiplier
NCP	nonlinear complementarity function
ODDM	overlapping domain decomposition method
pb	push-back
pf	push-forward
PK	Piola-Kirchhoff
RC	reference configuration
rhs	right hand side
RR	radial return method
SQP	sequential quadratic programming
TCP	tangent cutting plane algorithm
TL	total Lagrange
UL	updated Lagrange

WZL Werkzeugmaschinenlabor

## Small latin letters, alphabetical

$a_0$	dimension dependent scaling factor for the von Mises yield criterion
$c$	contact constant
$d \in \{2, 3\}$	spatial (problem) dimension
$e$	deviatoric strain
$e_r$	$r$ th unit vector in $\mathbb{R}^d$
$f^{int}$	bivariate form of internal forces
$f_n^{int}$	bivariate form of internal forces in the updated Lagrange formulation
$f_n^g$	bivariate form of the geometric stiffness in the updated Lagrange formulation
$f^{ext}$	density of external body forces
$g$	contact gap
$j$	index of Newton iteration
$k$	index of coarse-fine iteration
$l$	either index for linear solver loop or index for UL loop
$l^{ext}$	linear form, defined by $f^{ext}$
$l_n^g$	linear form of geometric residual in the updated Lagrange formulation
$l_c^p$	linear form of plastic load correction
$nc$	number of potential contact nodes
$n$	index of the (last known) time/load step
$n_h$	number of fine grid nodes
$n_H$	number of coarse grid nodes
$\mathbf{n}$	direction of plastic flow
$\tilde{\mathbf{n}}$	scaled direction of plastic flow
$p$	pressure
$\mathbf{s}$	deviatoric stress
$\mathbf{s}^{tr}$	trial deviatoric stress
$t$	time parameter
$t_n$	tool feed in step $n$
$\mathbf{u}$	continuous displacement field
$\mathbf{u}^h$	displacement field, discretized on the FE mesh $\mathbb{T}_h$
$v$	test function
$v^c$	Laursen/Love velocity correction

---

$\mathbf{x}_{ref}$	contact reference point
$\mathbf{x}_t$	coordinates of the material points in configuration $\Omega_t$

## Capital latin letters, alphabetical

$\mathbf{A}_H, \mathbf{A}_h$	coarse and fine linear elastic stiffness
$\mathbf{A}_T$	transformation matrix for the dual Lagrange multipliers
$\mathbb{A}$	set of active contact nodes
$\mathbf{B}$	contact stiffness
$\mathbf{C}$	global damping matrix
$\mathbf{C}^{el}$	Hooke tensor
$\mathbf{C}_n^{ep,(j)}$	elastoplastic tangent module at time $n$ and in Newton step $j$
$\mathbb{D}$	Dirichlet nodes/dof
$\mathbf{D}$	diagonal part of the contact stiffness $\mathbf{B}$
$\mathbb{E}_{h,n}, \mathbb{E}_{\hat{h},n}$	set of plastic history data on $\mathbb{T}_{h,n}$ and on $\mathbb{T}_{\hat{h}}$
$E$	Young's modulus
$E^{tot}$	total energy
$\mathbf{E}(\mathbf{u}; \mathbf{v})$	Gâteaux-derivative of $\mathbf{E}$ in direction of $\mathbf{v}$
$\mathbf{E}(\mathbf{u}; \Delta\Delta\mathbf{u}, \mathbf{v})$	Gâteaux-derivative of $\mathbf{E}(\mathbf{u}; \mathbf{v})$ in direction of $\Delta\Delta\mathbf{u}$
$\mathbf{F}^{con}$	contact residual vector
$\mathbf{F}^{int}$	standard finite element assembly of the internal forces
$\mathbf{F}^{int,\Delta}$	standard finite element assembly of incremental internal forces $t_n \rightarrow t_{n+1}$
$\mathbf{F}^{ext}$	standard finite element assembly of the external forces
$\mathbf{F}_H^{pl}$	standard finite element assembly of $l_c^{pl}$
$\mathbf{F}$	overall Newton residual
$\mathbf{F}_T$	element mapping
$\mathbf{F}_0^t$	deformation gradient for the deformation from $\Omega_0 \rightarrow \Omega_t$
$\mathbb{G}_{h,n}, \mathbb{G}_{\hat{h}}, \mathbb{G}_H$	sets of Gauss points
$G_p$	nodal gap and nodal velocity gap
$\mathbf{G}$	global gap vector
$\mathbf{H}$	history term for dynamic computations
$H_0^1(\Omega)$	space of test functions with zero Dirichlet boundary conditions
$\mathbf{I}$	fourth order identity tensor
$\mathbf{I}_{dev}$	fourth order deviatoric tensor
$\mathbb{I}$	set of inactive contact nodes
$J = J_T$	Jacobian, determinant of the element mapping $\mathbf{F}_T$
$J_0^t$	Jacobian, determinant of the deformation gradient

$\mathbf{J}$	Jacobi matrix of $\mathbf{Z}$
$K$	constant parameter for linear isotropic hardening
$\mathbf{K}$	Jacobi matrix of $\mathbf{F}^{int}$
$\mathbf{K}_n^g$	geometric stiffness UL, $t_n \rightarrow t_{n+1}$
$\mathbf{M}_\rho$	global mass matrix
$\mathbf{M}$	Mortar operator
$\mathbf{M}_T$	mass matrix of element $T$
$\mathbb{M}$	space of (dual) LM
$\mathbb{N}$	set of all structure nodes/dof
$\mathbf{N}_p$	nodal normal vector
$\mathbf{N}$	global contact normal matrices
$N$	number of grid nodes
$\mathbf{P}$	1st Piola-Kirchhoff stress tensor and standard prolongation operator
$\mathbf{P}_\nu$	global vector of normal impulses
$\mathbf{P}^{GPS \rightarrow C}, \mathbf{P}^{GPS \rightarrow F}, \mathbf{P}^{GF \rightarrow S}$	interpolation operators for plastic data
$\mathbb{S}$	set of potential contact nodes
$T$	surface element on the contact boundary, physical space or end of interval of interest in time dependent computations
$\hat{T}$	surface element on the contact boundary, isoparametric space
$\mathbf{T}$	2nd Piola-Kirchhoff stress tensor
$\mathbf{T}_p^\xi, \mathbf{T}_p^\eta$	tangential vectors normal to $\mathbf{N}_p$
$\mathbf{T}^\xi, \mathbf{T}^\eta$	global contact tangent matrices
$\mathbb{T}_H, \mathbb{T}_{h,n}, \mathbb{T}_{\hat{h}}$	coarse, fine and storage mesh at time step $n$
$\mathbf{U}_p$	nodal displacement vector
$\mathbf{U}^h$	global coefficient vector of $\mathbf{u}^h$
$Y$	function of isotropic hardening
$\mathbf{Z}$	effective dynamic structural forces

## Small greek letters, alphabetical

$\alpha$	hardening parameter
$\alpha_m, \alpha_f$	GEMM and Gen- $\alpha$ parameter
$\beta$	Newmark parameter
$\gamma$	Newmark parameter
$\delta_{pq}$	Kronecker symbol
$\boldsymbol{\varepsilon}$	linearized strain tensor
$\boldsymbol{\varepsilon}^e, \boldsymbol{\varepsilon}^p$	elastic and plastic part of the linearized strain tensor
$\kappa$	bulk modulus
$\kappa^{plas}$	volume penalty in the rigid plastic constitutive law
$\lambda$	plastic consistency parameter



---

$\tilde{\lambda}$	scaled plastic consistency parameter
$\bar{\lambda} = \lambda_{n+1}$	current contact stress w.r.t. time $t_{n+1}$
$\lambda_L$	Lamé parameter
$\mu_L$	Lamé parameter
$\mu^{plas}$	viscosity coefficient of the plastic flow
$\nu$	Poisson ratio
$\boldsymbol{\nu}$	contact normal on the tool
$\rho$	either contact penalty or mass density
$\rho_\infty$	dissipation coefficient
$\sigma_0$	(initial) yield stress
$\boldsymbol{\sigma}^{tr}$	trial stress
$\boldsymbol{\sigma}_t = \boldsymbol{\sigma}_n = \boldsymbol{\sigma}(\mathbf{u}_n)$	current (Cauchy) stress at time $t(= t_n)$ and with respect to time $t$
$\phi_i$	scalar FE ansatzfunktion at the $i$ th node, linear if not stated otherwise
$\hat{\phi}_i$	FE ansatzfunktion at the $i$ th node, isoparametric space
$\varphi_t$	placement reference configuration into $\Omega_t$
$\psi_i$	scalar valued dual LM at node $i$
$\hat{\psi}_i$	scalar valued dual LM at node $i$ , isoparametric space

## Capital greek letters, alphabetical

$\Gamma^{con}$	potential contact zone
$\Gamma^{tool}$	tool boundary
$\Gamma_c$	coarse potential (reference) contact domain
$\Gamma_c^d$	coarse Dirichlet boundary
$\Gamma_{f,n}$	fine potential (reference) contact domain at process step $n$
$\Gamma_{f,n}^d$	fine potential (reference) Dirichlet boundary at process step $n$
$\Delta E^{num}$	pure numerical change in energy
$\Delta t$	time step size
$\Delta \mathbf{U}$	Newton increment or time increment (depending on the context)
$\boldsymbol{\Lambda}_p$	nodal vector of contact stresses
$\boldsymbol{\Lambda}_\nu$	global vector of normal contact stresses
$\boldsymbol{\Lambda}_{\nu,p}$	normal part of $\boldsymbol{\Lambda}_p$
$\boldsymbol{\Lambda}_{\tau,p}$	tangential part of $\boldsymbol{\Lambda}_p$
$\Phi$	yield function
$\Psi^e$	elastic potential
$\Omega = \Omega_0$	reference configuration

$\Omega_c$	global reference domain
$\Omega_{f,n}$	fine reference domain at process step $n$
$\Omega_t = \Omega_n$	last known configuration
$\bar{\Omega} = \Omega_{t+\Delta t} = \Omega_{n+1}$	unknown configuration
$\Omega_{p,n}$	zone of plastic deformation

## Mathematical and other symbols

$\nabla_t(\cdot) = \frac{\partial}{\partial \mathbf{x}_t}(\cdot)$	gradient w.r.t. the deformed coordinates of $\Omega_t$
$[\cdot]_{i,j=1,2,\dots}$	$i$ row index, $j$ column index
$\partial\Omega$	boundary of the domain/workpiece
$\mathbf{Q}[i] = Q_i$	$i$ th component of the vector $\mathbf{Q}$
$\mathbf{1}$	second order identity tensor
$\overline{(\cdot)}$	either equivalent strain/stress or current configuration

# 1 Introduction

CNC-controlled incremental metal forming processes have a great innovative potential for the cost-effective manufacturing of workpieces with complex geometries. Important characteristics are the high flexibility and the small power requirements. New processing machines, materials and control techniques have led to a technology, which is supposed to provide a very interesting alternative to conventional forming techniques as for example deep drawing. As these new processes are very difficult to control, there is a need for the modelling and simulation. For extending the usability of implicit Finite Element (FE) codes to large scale incremental metal forming simulations, the computation time has to be decreased dramatically. Two incremental processes are introduced to motivate the research work on the subject, and the main challenges of the implicit FE simulation and solution methods are presented.

In this work, the results of the research activities within the subproject ‘Entwicklung effizienter numerischer Simulationsalgorithmen für CNC-gesteuerte inkrementelle Umformverfahren’ of the DFG priority program SPP 1146 ‘Modellierung inkrementeller Umformverfahren’ are presented. It should be stated that parts of this work have already been published in [17, 18, 19, 20, 50], see Section 1.3.

## 1.1 Deep-rolling

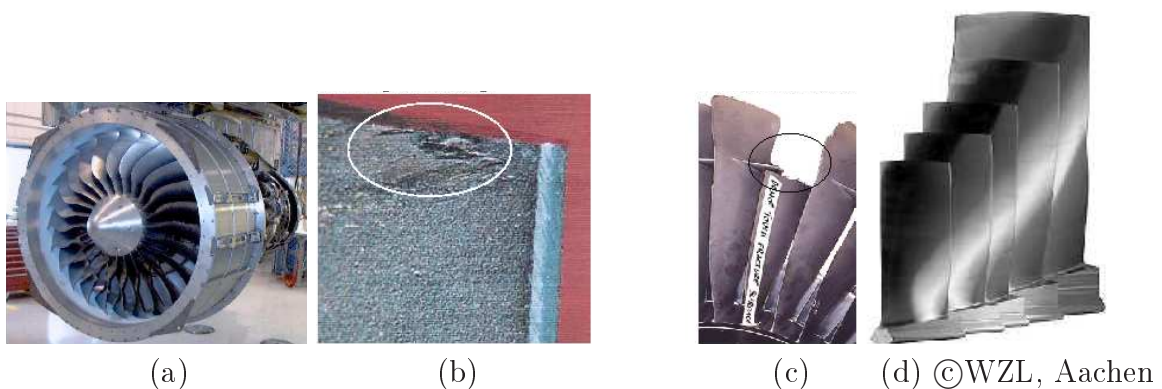


Figure 1.1: Technical application for deep-rolling.

For a more detailed description of the process and the technical applications, see [63] and the references cited therein. The weakening of the fatigue strength of turbine blades

(see Figure 1.1 (d)) due to local impact damage represents a significant safety risk in modern aviation, see Figure 1.1 (a)-(c). Notches, tensile residual stresses and sometimes even micro cracks are induced. It is possible to employ deep rolling, see Figure 1.2, to counteract such component weakening in an effective way. Strain hardening of the deep-rolled rim zone, see Figure 1.2 (d), is one promising way to suppress the crack formation resulting from alternating loads, as well as to stop or slow down crack growth. Deep rolling of thin-walled fan and compressor blades is possible by using special rolling tools, see Figure 1.2 (b),(c) and [55]. These tools use hydrostatically supported balls as rolling elements. The design of such deep-rolling processes is, however, associated with time- and cost-intensive test efforts as no suitable methods exist for deriving the necessary process variables from the desired rim zone properties. In order to derive a possible process control parameter-set in advance, a FE analysis is very helpful, in order to understand the complex deformation mechanisms occurring during the rolling contact.

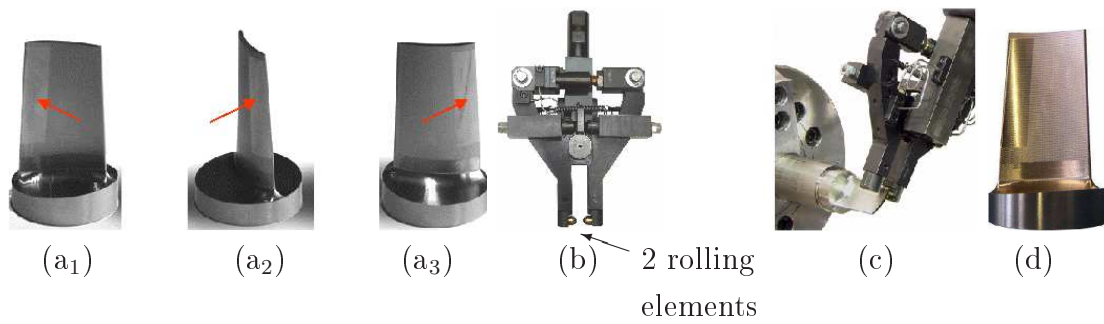


Figure 1.2: Deep-rolling, ©WZL, Aachen.

## 1.2 Incremental sheet metal forming

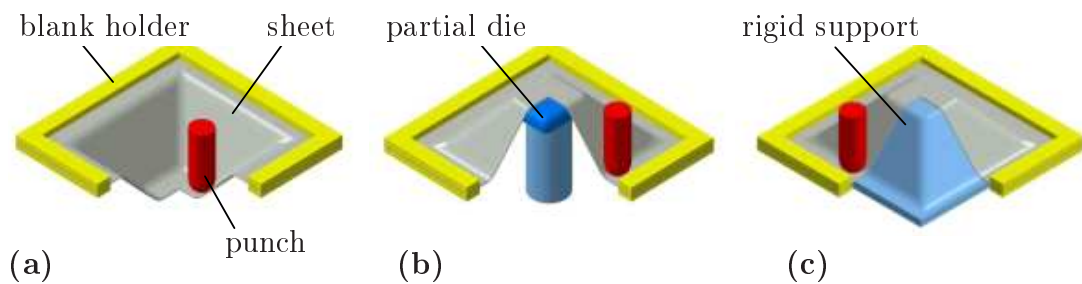


Figure 1.3: Some variants of incremental sheet metal forming processes, see [60].

Incremental sheet metal forming (ISF) is a recently introduced forming technique, using simple shaped, small and flexible tools to form sheet metal workpieces [25, 54, 83]. In Figure 1.3 some variants of this technique are depicted. As one can see from Figure 1.4, the ISF process is very difficult to control. Wrinkles or even cracks occur if the CNC-controlled path is not appropriate. The aim of this work is not to contribute to

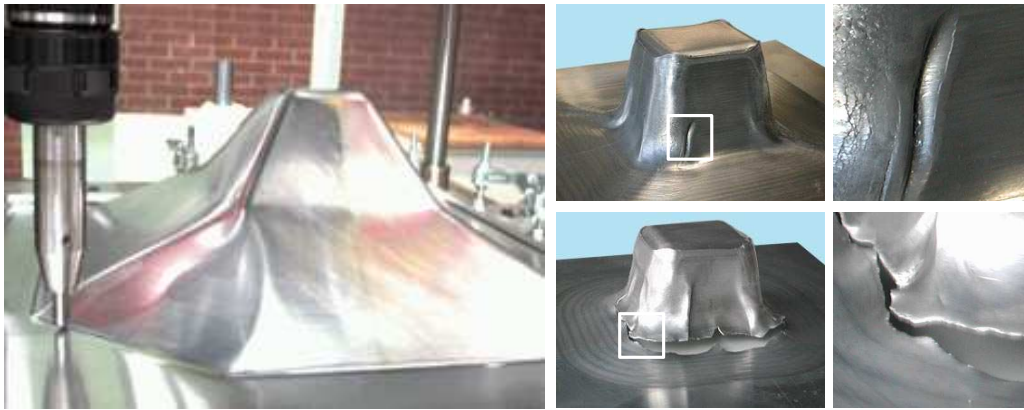


Figure 1.4: ©IBF, Aachen, incremental sheet metal forming (left), wrinkles and cracks (right).

the simulation of crack propagation but to provide some basic technology that will help to accelerate the implicit FE simulation of such processes *before* they degenerate, and to make it feasible in the near future. The goal to be kept in mind is always to sort out the processes which lead to undesired behavior as soon as it is detected. To find the ‘good’ processes by an optimization process would be part of the future work. The competitors of the implicit FEM, namely explicit methods, can be speeded up by means of mass or load scaling to enable the simulation of large scale sheet metal forming problems. However, it is well known that these methods can entail poor results when dynamic effects start to dominate the solution, see [29]. On the other hand, implicit FE simulation is a highly challenging task.

### 1.3 Aim and outline of this work

Only some of the difficulties one has to face are listed here.

- Ill conditioned system matrices due to the use of shell elements.
- Several kinds of locking.

In this work, the focus is on the following points:

- Ill conditioned system matrices due to the use of penalty parameters and therefore difficulties with the strongly desirable use of iterative solvers, see **Chapter 3**. A contact strategy using Lagrange multipliers is introduced there to cope with these difficulties. Chapter 3 is a joint work within the SPP 1146 and is published in [19].
- The interaction between tool and workpiece demands stable and inexpensive coupling schemes.

- The forming zone is small but very mobile. The punch is touching almost every point of the workpiece surface at some time in the process. Therefore the number of degrees of freedom (dof) has to be very large in a non-adaptive computation. Also the number of time steps is very large since the total duration of an incremental shaping procedure is at least several minutes. The main idea to tackle these problems is to use a ‘divide and conquer’ approach: A dynamic overlapping domain decomposition (ODDM) is employed by discretizing the global workpiece with a coarse mesh and the small forming zone with a relatively fine mesh, see **Chapter 6**. In **Chapter 2** and **Chapter 3**, a short review of the classical FE solution process without any adaptive computation is recalled. Parts of Chapter 2 have been published in [17]. The purpose of these two chapters is also to provide definitions and notations which are needed in the description of the coupling algorithm. The two main aspects of this algorithm, the treatment of the material nonlinearity and the treatment of the contact are explained in **Chapter 6** and first considerations are made about the blockwise (first block: coarse system, second block: fine system) solution process. The results of this chapter are published in [20]. In **Chapter 7**, the solution process is improved and put into the perspective of classical inexact Newton schemes.
- Sometimes ISF simulations are performed dynamically and with an implicit time integration for two reasons:
  1. ISF can actually be regarded as a dynamic process since elastic waves are travelling through the material and dynamic springback (movement of the workpiece due to a sudden load change or after it is released from the clamping) occurs.
  2. It seems to be a reasonable hope that a computation with a lumped mass matrix will improve the condition number of the overall effective stiffness matrix [84] and will help to overcome difficulties in the pre-buckling phase.

These two points are not completely within the scope of this work, however energy consistent time integration of the contact problem is a prerequisite for these points and is treated in **Chapter 4**. For the energy consistent treatment of plasticity in the regime of large deformations, pioneering work has been done, see for example [86] and the references therein.

- First steps were taken for the dynamic contact analysis of thin walled structures, see **Chapter 4** which is a result of a cooperation with Ramm/Hartmann [50].
- Complex linearization problems arise due to
  1. displacement based (**Chapter 3**) and velocity based (**Chapter 5**, published in [18]) contact problems (boundary nonlinearity)
  2. the elastoplastic constitutive law, **Chapter 2** (material nonlinearity)
  3. large deformations, **Chapter 8** (geometric nonlinearity)

- Finally, the acceleration of the computation by using iterative solvers efficiently especially in the framework of inexact Newton methods is a topic of **Chapter 3** and **Chapter 7**.

In **Chapter 9**, concluding remarks are made. One last word on the citations: Works from several disciplines of computational mechanics and numerical mathematics are cited here. Although all quoted works are important contributions to these fields, the list is rather fragmentary, and the author apologizes to those who had deserved to be quoted here and are not.





## 2 Classical 1-grid discretization: Plasticity

A short review of the classical FEM solution process of elastoplastic problems without any mesh-adaptivity is recalled in this chapter. Some basic concepts of metal plasticity will be considered and the need for consistent linearization of the constitutive relations. In order to keep the description of the coupling technique in Chapter 6 clear, a quite simple material model was used, namely elastoplasticity for small deformations, see classical textbooks on computational mechanics and plasticity [10, 49, 97, 109] and the references cited therein.

### 2.1 Introduction

In many large scale forming simulation FE packages there is the problem that the modeling of elastoplasticity is only validated well with examples like deep-drawing or other examples where the plastic strain, here in this work taken as the driving variable, is increasing in every time step. Incremental forming operations, make greater demands on the elastoplastic simulation because of the permanently changing loading situation. In this chapter, the linearization of the constitutive equations emphasized and not the large deformations. In Section 2.2 and Section 2.3, a simple plasticity model the finite element discretization are detailed. In Section 2.4, a classical return mapping algorithm is discussed and the so called consistently linearized material tangent is derived. In Section 2.5, the tangent cutting plane algorithm is discussed which is implemented in LARSTRAN/SHAPE [33], a FE package dedicated to metal forming simulations. In a quite different direction goes the sequential quadratic programming (SQP) approach. This is presented in Section 2.6. In Section 2.7 the different approaches for the algorithmic treatment of the constitutive law are briefly compared. In Section 2.8 and in Section 2.9, numerical results and some rules for tensor calculus are given.

### 2.2 Governing equations and notations

It is assumed here, that an additive decomposition of the linearized strain tensor  $\boldsymbol{\varepsilon} \in \mathbb{R}^{d \times d}$  ( $d \in \{2, 3\}$  is the problem dimension) into an elastic part  $\boldsymbol{\varepsilon}^e$  and a plastic part  $\boldsymbol{\varepsilon}^p$  is valid.

The governing equations for the constitutive law are the following:

$$\boldsymbol{\varepsilon}^e := \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p \quad (2.1)$$

$$\boldsymbol{\sigma} = \mathbf{C}^{el} \boldsymbol{\varepsilon}^e \quad (2.2)$$

$$\dot{\boldsymbol{\varepsilon}}^p = \lambda \partial_{\boldsymbol{\sigma}} \Phi(\boldsymbol{\sigma}, \alpha) \quad (2.3)$$

$$\dot{\alpha} = \lambda \quad (2.4)$$

$$\lambda \geq 0, \quad \Phi(\boldsymbol{\sigma}, \alpha) \leq 0 \quad \text{and} \quad \lambda \Phi(\boldsymbol{\sigma}, \alpha) = 0$$

with  $\boldsymbol{\sigma} \in \mathbb{R}^{d \times d}$  the actual Cauchy stress,  $\lambda \in \mathbb{R}$  the plastic consistency parameter. The 4th order tensor  $\mathbf{C}^{el}(\mu_L, \lambda_L)$  is the isotropic elastic material tensor with the Lamé parameters  $(\mu_L, \lambda_L)$  and  $\Phi : \mathbb{R}^{d \times d} \times \mathbb{R} \rightarrow \mathbb{R}$  is the yield function defined by

$$\Phi(\boldsymbol{\sigma}, \alpha) = \bar{\sigma} - Y(\alpha) \quad (2.5)$$

By  $Y : \mathbb{R} \rightarrow \mathbb{R}$ , the function of isotropic hardening with the hardening parameter  $\alpha \in \mathbb{R}$  is denoted. The equivalent stress  $\bar{(\cdot)} : \mathbb{R}^{d \times d} \rightarrow \mathbb{R}$  can be defined by

$$\bar{\sigma} := \begin{cases} a_0 \sqrt{\text{dev } \boldsymbol{\sigma} : \text{dev } \boldsymbol{\sigma}} & \text{von Mises} \\ \left( \frac{1}{2} [a|K_1 + K_2|^M + a|K_1 - K_2|^M + (2-a)|2K_2|^M] \right)^{1/M} & \text{Barlat/Lian, [7]} \\ \vdots & \vdots \end{cases} \quad (2.6)$$

with

$$K_1 := \frac{1}{2}(\sigma_{xx} + h\sigma_{yy}), \quad K_2 := \sqrt{\frac{1}{4}(\sigma_{xx} - h\sigma_{yy})^2 + p^2\sigma_{xy}^2}$$

and  $a, h, p, M$  material parameters for the Barlat/Lian yield function and with the dimension dependent scaling factor

$$a_0 := \sqrt{\frac{d}{d-1}}$$

for the von Mises yield function. The function of isotropic hardening can be defined by

$$Y(\alpha) := \begin{cases} Y(0) + K\alpha & \text{linear isotropic} \\ \vdots & \vdots \end{cases} \quad (2.7)$$

with  $K \in \mathbb{R}_+$  the slope of the linear isotropic hardening curve. There is a vast literature about other possible yield functions, see [3].

## 2.3 Global equilibrium iteration and Finite Element discretization

Given a grid with  $N$  nodes and with FE ansatz functions  $\phi_1, \dots, \phi_N$ , it is recalled briefly how the global equilibrium iteration is set up: Since small deformations are assumed, all

integrations can be carried out with respect to the reference configuration  $\Omega$ . By  $H_0^1(\Omega)$ , the space of test functions with zero-Dirichlet boundary conditions <sup>1</sup> is denoted. The relation  $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{u})$  is nonlinear and not even differentiable in the strong sense. A closed form of the relation is not needed here. Usually, before spatial discretization a time discretization is performed. In addition to commonly used Euler first-order methods, which are used here, higher order methods with step size control have gained increasing attention in the last years, see for example [36] and the references therein. Throughout this chapter, the general trapezoidal rule ( $\gamma \in [0, 1]$ )

$$\mathbf{z}_{n+1} = \mathbf{z}_n + \Delta t(\gamma\Phi(\mathbf{z}_{n+1}) + (1 - \gamma)\Phi(\mathbf{z}_n))$$

is used with  $\gamma = 1$  (backward, implicit Euler). The time step  $t_n \rightarrow t_{n+1}$  is considered and the global equilibrium at the current time step  $t_{n+1}$  with the external forces  $\mathbf{f}_{n+1}^{ext}$  acting on the body is enforced. Strong emphasis should be placed on the fact that the stress  $\boldsymbol{\sigma}_n$  does not only depend on the current displacements  $\mathbf{u}_n$  but also on the complete path. So there is also a dependence on  $\mathbf{u}_{n-1}, \mathbf{u}_{n-2}, \dots, \mathbf{u}_1$ . The global equilibrium equation reads in weak form

$$f^{int}(\mathbf{u}_{n+1}, \mathbf{v}) := \int_{\Omega} \boldsymbol{\sigma}_{n+1}(\mathbf{u}_{n+1}) : \boldsymbol{\varepsilon}(\mathbf{v}) = \int_{\Omega} \mathbf{f}_{n+1}^{ext} \cdot \mathbf{v} =: l_{n+1}^{ext}(\mathbf{v}), \quad \mathbf{v} \in [H_0^1(\Omega)]^d \quad (2.8)$$

Discretize  $\mathbf{u}_{n+1}$  by  $\mathbf{u}_{n+1}^h$  and define the vectors  $\mathbf{F}^{int}(\mathbf{U}_{n+1})$  and  $\mathbf{F}^{ext}$  as standard FE assemblies of  $f^{int}$  and  $l^{ext}$  and the vector  $\mathbf{F}(\mathbf{U}_{n+1})$  as

$$\mathbf{F}(\mathbf{U}_{n+1}) := \mathbf{F}^{int}(\mathbf{U}_{n+1}) - \mathbf{F}^{ext}$$

A few remarks on the notation: The diction  $\mathbf{F}[i]$  and  $F_i$  mean the same, namely the  $i$ th component of the vector  $\mathbf{F}$ . Often FE functions, for example  $\mathbf{u}^h$ , are identified with their coefficient vector  $\mathbf{U}$ , and sometimes the time index  $n$  is omitted. With this, the equilibrium equation (2.8) in the discrete form is formulated now:

$$\mathbf{F}(\mathbf{U}_{n+1}) = \mathbf{0} \quad (2.9)$$

After consistent linearization [97, 99], one arrives at a global system of linear equations with the tangential stiffness  $\mathbf{K}_{n+1}^{(j)}$  in the  $j$ th Newton step defined as

$$\mathbf{K}_{n+1}^{(j)}[ik] := \frac{\partial \mathbf{F}(\mathbf{U}_{n+1}^{(j)})[i]}{\partial \mathbf{U}^{(j)}[k]} = \frac{\partial \mathbf{F}^{int}(\mathbf{U}_{n+1}^{(j)})[i]}{\partial \mathbf{U}^{(j)}[k]} \quad (2.10)$$

---

<sup>1</sup>convenient abuse of notation in order to keep the notation simple in the case when more than one Dirichlet boundary is involved

The corresponding derivative is determined as follows (sometimes the relation  $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{u})$  is not written explicitly):

$$\begin{aligned}
 K_{ik}^{(j)} &= \frac{\partial F_i^{int}(\mathbf{U}_{n+1}^{(j)})}{\partial U_k^{(j)}} \\
 &= \int_{\Omega} \frac{\partial}{\partial U_k^{(j)}} \boldsymbol{\sigma}_{n+1}^{(j)} : \boldsymbol{\varepsilon}(\phi_i) \\
 &= \int_{\Omega} \boldsymbol{\varepsilon}(\phi_k) : \mathbf{C}_{n+1}^{ep,(j)} : \boldsymbol{\varepsilon}(\phi_i)
 \end{aligned} \tag{2.11}$$

In the last equality, the chain rule was used with the inner derivative  $\frac{\partial}{\partial U_k} \boldsymbol{\varepsilon}(\mathbf{U}_{n+1}) = \boldsymbol{\varepsilon}(\phi_k)$  and the outer derivative, which shall be defined as the 4th order tensor

$$\mathbf{C}_{n+1}^{ep,(j)} := \frac{\partial \boldsymbol{\sigma}_{n+1}^{(j)}}{\partial \boldsymbol{\varepsilon}(\mathbf{U}_{n+1}^{(j)})} \tag{2.12}$$

which is denoted as elastoplastic tangent. Finally, the zone of plastic deformation  $\Omega_{p,n} \subset \Omega$  at the time step  $t_n$  is defined as the zone where new plastic deformation takes place:

$$\Omega_{p,n} := \{\mathbf{x} \in \Omega : \boldsymbol{\varepsilon}^p(\mathbf{x}, t_n) \neq \boldsymbol{\varepsilon}^p(\mathbf{x}, t_{n-1})\} \tag{2.13}$$

This definition will be needed in the coupling algorithm.

The following sections are basically concerned with the elastoplastic tangent in Equation (2.12) and the fulfillment of the yield condition.

*Remark:* The notation in (2.12) is a little bit sloppy: It turns out that  $\mathbf{F}$  is only a Lipschitz continuous function. Thus,  $\mathbf{C}_{n+1}^{ep,(j)}$  cannot be viewed as a unique total derivative but it is one element of the generalized Jacobian as it is introduced by Clarke, see for example [62].

## 2.4 A Gauss point section: Radial Return for linear isotropic hardening

Now the radial return (RR) algorithm for linear isotropic hardening is described. The purpose of this algorithm is to fulfill the constitutive Equations (2.1) - (2.4) at the end of each load step. One sees that within the framework of this stress integration algorithm, a consistent material tangent can be readily obtained. Except for minor technical details, the resulting algorithm is the same as documented in, e.g., [97].

**Definitions** The yield function is now specified to a von-Mises type of yield function with linear isotropic hardening (the first case in (2.6) and in (2.7)). Some abbreviations

have to be made.

$$\begin{aligned}
 \tilde{\lambda} &:= a_0^2 \lambda \quad \text{scaled plastic consistency parameter} & (2.14) \\
 \mathbf{s} &:= \text{dev}(\boldsymbol{\sigma}) \quad \text{deviatoric stress} \\
 \|\boldsymbol{\sigma}\| &:= \sqrt{\boldsymbol{\sigma} : \boldsymbol{\sigma}} \quad \text{Frobenius norm}
 \end{aligned}$$

Because of the structure of  $\Phi$ , the ordinary differential equation for the evolution of the plastic strain can be written explicitly in terms of the stress:

$$\dot{\boldsymbol{\epsilon}}^p = \lambda \frac{a_0 \mathbf{s}}{\|\mathbf{s}\|} = \tilde{\lambda} \frac{\mathbf{s}}{\bar{\mathbf{s}}} \quad (2.15)$$

Here is  $\bar{\mathbf{s}} = a_0 \sqrt{\text{dev } \mathbf{s} : \text{dev } \mathbf{s}} = a_0 \|\mathbf{s}\|$  according to Equation (2.6)<sub>1</sub>. In the following algorithm, the notations  $\mathbf{1}$ ,  $\mathbf{I}$  and

$$\mathbf{I}_{dev} := \mathbf{I} - \frac{1}{d} \mathbf{1} \otimes \mathbf{1}$$

are used for the second order identity tensor, the fourth order symmetric identity tensor and the fourth order deviatoric tensor.

**Algorithm**  $[\boldsymbol{\epsilon}_{n+1}^{p,(j)}, \boldsymbol{\sigma}_{n+1}^{(j)}, \mathbf{C}_{n+1}^{ep,(j)}, \alpha_{n+1}^{(j)}] = \text{RADIAL\_RETURN}(\boldsymbol{\epsilon}_{n+1}^{(j-1)}, \boldsymbol{\epsilon}_n^p, \alpha_n, \Phi)$

1. Compute trial stress

$$\begin{aligned}
 \kappa &:= \lambda_L - \frac{2}{d} \mu_L & \text{bulk modulus} \\
 \mathbf{e}_{n+1} &:= \mathbf{I}_{dev} \boldsymbol{\epsilon}_{n+1}^{(j-1)} & \text{deviatoric strain} \\
 \mathbf{s}_{n+1}^{tr} &:= 2\mu_L (\mathbf{e}_{n+1} - \boldsymbol{\epsilon}_n^p) & \text{trial deviatoric stress}
 \end{aligned}$$

2. Check yield condition

$$\Phi_{n+1}^{tr} := \bar{\mathbf{s}}_{n+1}^{tr} - Y(\alpha_n) \quad \text{evaluation of the yield function}$$

IF  $\Phi_{n+1}^{tr} \leq 0$  THEN:

$$\begin{aligned}
 \boldsymbol{\sigma}_{n+1}^{(j)} &:= \kappa \text{tr}(\boldsymbol{\epsilon}_{n+1}^{(j-1)}) + \mathbf{s}_{n+1}^{tr} \\
 \boldsymbol{\epsilon}_{n+1}^{p,(j)} &:= \boldsymbol{\epsilon}_n^p \\
 \mathbf{C}_{n+1}^{ep,(j)} &:= \mathbf{C}^{el} \\
 \alpha_{n+1}^{(j)} &:= \alpha_n
 \end{aligned}$$

AND EXIT

3. Compute flow direction  $\mathbf{n}_{n+1}$  and plastic consistency parameter  $\lambda$

$$\begin{aligned}
 \lambda &:= \text{root}[g(\lambda) = 0] && \text{plastic consistency parameter, see Section 2.4.2} \\
 \mathbf{n}_{n+1} &:= \frac{1}{\|\mathbf{s}_{n+1}^{tr}\|} \mathbf{s}_{n+1}^{tr} && \text{flow direction} \\
 \tilde{\mathbf{n}}_{n+1} &:= a_0^{-1} \mathbf{n}_{n+1} && \text{scaled flow direction} \\
 \alpha_{n+1}^{(j)} &:= \alpha_n + \lambda && \text{backward Euler for (2.4)}
 \end{aligned} \tag{2.16}$$

4. Update plastic strain and stress according to (2.15)

$$\begin{aligned}
 \boldsymbol{\varepsilon}_{n+1}^{p,(j)} &:= \boldsymbol{\varepsilon}_n^p + \tilde{\lambda} \tilde{\mathbf{n}}_{n+1} \\
 \boldsymbol{\sigma}_{n+1}^{(j)} &:= \kappa \text{tr}(\boldsymbol{\varepsilon}_{n+1}^{(j-1)}) \mathbf{1} + \mathbf{s}_{n+1}^{tr} - 2\mu_L \tilde{\lambda} \tilde{\mathbf{n}}_{n+1} \\
 &= \kappa \text{tr}(\boldsymbol{\varepsilon}_{n+1}^{(j-1)}) \mathbf{1} + \mathbf{s}_{n+1}^{tr} - 2\mu_L a_0 \lambda \mathbf{n}_{n+1}
 \end{aligned} \tag{2.17}$$

5. Compute the *consistent* elastoplastic tangent module  $\mathbf{C}_{n+1}^{ep,(j)}$  according to Equation (2.18)

### 2.4.1 Consistent elastoplastic tangent module

For the sake of brevity, the index for the Newton iteration ( $j$ ) is omitted in the following. With (2.21) and some equations from tensor calculus, see Section 2.9, the following derivatives can be computed:

$$\begin{aligned}
 \partial_{\boldsymbol{\varepsilon}_{n+1}}(\text{tr}(\boldsymbol{\varepsilon}_{n+1})\mathbf{1}) &= \mathbf{1} \otimes \mathbf{1} \\
 \partial_{\boldsymbol{\varepsilon}_{n+1}}(\lambda \mathbf{n}_{n+1}) &= \mathbf{n}_{n+1} \otimes \frac{\partial \lambda}{\partial \boldsymbol{\varepsilon}_{n+1}} - \lambda \frac{\partial \mathbf{n}_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} \\
 \frac{\partial \lambda}{\partial \boldsymbol{\varepsilon}_{n+1}} &= \frac{a_0^{-1}}{a_0^{-2}K + 2\mu_L} \frac{\mathbf{s}_{n+1}^{tr}}{\|\mathbf{s}_{n+1}^{tr}\|} : \frac{\partial \mathbf{s}_{n+1}^{tr}}{\partial \boldsymbol{\varepsilon}_{n+1}} = \frac{a_0^{-1}}{a_0^{-2}K + 2\mu_L} 2\mu_L \frac{\mathbf{s}_{n+1}^{tr}}{\|\mathbf{s}_{n+1}^{tr}\|} \\
 &= 2\mu_L \frac{a_0^{-1}}{a_0^{-2}K + 2\mu_L} \mathbf{n}_{n+1}
 \end{aligned}$$

Using the fact that  $\mathbf{n}_{n+1}$  is trace-free:

$$\begin{aligned}
 \partial_{\boldsymbol{\varepsilon}_{n+1}} \mathbf{n}_{n+1} &= \frac{\partial \mathbf{n}_{n+1}}{\partial \mathbf{s}_{n+1}^{tr}} \frac{\partial \mathbf{s}_{n+1}^{tr}}{\partial \boldsymbol{\varepsilon}_{n+1}} \\
 &= \frac{1}{\|\mathbf{s}_{n+1}^{tr}\|} (\mathbf{I} - \mathbf{n}_{n+1} \otimes \mathbf{n}_{n+1}) \frac{\partial \mathbf{s}_{n+1}^{tr}}{\partial \boldsymbol{\varepsilon}_{n+1}} \\
 &= \frac{1}{\|\mathbf{s}_{n+1}^{tr}\|} (\mathbf{I} - \mathbf{n}_{n+1} \otimes \mathbf{n}_{n+1}) 2\mu_L \mathbf{I}_{dev} \\
 &= \frac{2\mu_L}{\|\mathbf{s}_{n+1}^{tr}\|} (\mathbf{I}_{dev} - \mathbf{n}_{n+1} \otimes \mathbf{n}_{n+1})
 \end{aligned}$$

With this, one arrives at the desired tangent module:

$$\begin{aligned}
 \mathbf{C}_{n+1}^{ep} &= \frac{\partial \boldsymbol{\sigma}_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} \\
 &= \kappa \mathbf{1} \otimes \mathbf{1} + 2\mu_L \mathbf{I}_{dev} - 2\mu_L a_0 \left[ \mathbf{n}_{n+1} \otimes \frac{\partial \lambda}{\partial \boldsymbol{\varepsilon}_{n+1}} + \lambda \frac{\partial \mathbf{n}_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} \right] \\
 &= \mathbf{C}^{el} - 2\mu_L a_0 \mathbf{n}_{n+1} \otimes \frac{2\mu_L a_0^{-1}}{a_0^{-2}K + 2\mu_L} \mathbf{n}_{n+1} - 2\mu_L a_0 \frac{2\mu_L \lambda}{\|\mathbf{s}_{n+1}^{tr}\|} (\mathbf{I}_{dev} - \mathbf{n}_{n+1} \otimes \mathbf{n}_{n+1}) \\
 &= \mathbf{C}^{el} - \mathbf{C}_{n+1}^p
 \end{aligned} \tag{2.18}$$

with the abbreviations

$$\begin{aligned}
 \tilde{\theta} &:= \frac{2\mu_L}{a_0^{-2}K + 2\mu_L} - \frac{2\mu_L \tilde{\lambda}}{\bar{\mathbf{s}}_{n+1}^{tr}} \\
 \mathbf{C}_{n+1}^p &:= 2\mu_L \frac{2\mu_L \tilde{\lambda}}{\bar{\mathbf{s}}_{n+1}^{tr}} \mathbf{I}_{dev} - 2\mu_L a_0^2 \tilde{\theta} \tilde{\mathbf{n}}_{n+1} \otimes \tilde{\mathbf{n}}_{n+1}
 \end{aligned} \tag{2.19}$$

## 2.4.2 Plastic consistency parameter

The aim of this section is to compute an expression for the plastic consistency parameter  $\lambda$ . The deviatoric part of Equation (2.17) is:

$$\mathbf{s}_{n+1} - \mathbf{s}_{n+1}^{tr} + 2\mu_L a_0 \lambda \mathbf{n}_{n+1} = \mathbf{0}$$

Multiplying this with  $\mathbf{n}_{n+1}^\top$  and using that

$$\mathbf{n}_{n+1} = \frac{\mathbf{s}_{n+1}^{tr}}{\|\mathbf{s}_{n+1}^{tr}\|} = \frac{\mathbf{s}_{n+1}}{\|\mathbf{s}_{n+1}\|}$$

one gets:

$$g(\lambda) := \|\mathbf{s}_{n+1}\| - \|\mathbf{s}_{n+1}^{tr}\| + 2\mu_L a_0 \lambda = 0 \tag{2.20}$$

Using that the yield condition is fulfilled at  $t_{n+1}$  and with the aid of (2.16), one can compute:

$$\begin{aligned}
 g(\lambda) &= a_0^{-1} Y(\alpha_{n+1}) - \|\mathbf{s}_{n+1}^{tr}\| + 2\mu_L a_0 \lambda \\
 &= a_0^{-1} Y(0) + a_0^{-1} K \alpha_{n+1} - \|\mathbf{s}_{n+1}^{tr}\| + 2\mu_L a_0 \lambda \\
 &= a_0^{-1} Y(0) + a_0^{-1} K (\alpha_n + \lambda) - \|\mathbf{s}_{n+1}^{tr}\| + 2\mu_L a_0 \lambda
 \end{aligned}$$

The consistency parameter  $\lambda$  can be written explicitly here as root of  $g$ .

$$\lambda = \frac{a_0^{-1} Y(0) + \|\mathbf{s}_{n+1}^{tr}\| - a_0^{-1} K \alpha_n}{a_0^{-1} K + 2a_0 \mu_L} \tag{2.21}$$

$$\Leftrightarrow \tilde{\lambda} = \frac{-Y(0) - K \alpha_n + \bar{\mathbf{s}}_{n+1}^{tr}}{a_0^{-2} K + 2\mu_L} \tag{2.22}$$

Note that consistent linearization of the RR algorithm is also straightforward in the case of nonlinear isotropic and kinematic hardening, see [97], as long as the context of J2-plasticity is not left.

## 2.5 A Gauss point section: The Tangent Cutting Plane algorithm

The following algorithm is given in [3]. It is based on works on tangent cutting plane (TCP) algorithms like for example [98]. A need for the tangent cutting plane algorithm is given when

1. anisotropic plasticity occurs, e.g. the second case of (2.6) applies. See the situation in Figure 2.1 (observe that the yield surface does not only change the location and size due to kinematic and isotropic hardening but also the shape due to the anisotropy), where classical return mapping algorithms do not apply any more
2. **and** when the user cannot or is not willing to provide twice differentiable yield functions in order to make general return mapping algorithms ([97], Chapter 3) applicable.

Unlike in Section 2.4, this Section is not restricted to the case of a von-Mises type yield function (2.6)<sub>1</sub>. Thus the deviatoric parts of the stress and the strain tensors are not computed in the TCP algorithm. Another difference to RR is that in this method a closed expression for  $\lambda$  like in Equation (2.21) cannot be derived since the yield function is supposed to be more general. Hence, a local Newton iteration, see Section 2.5.1, must be performed to guarantee the fulfillment of the yield condition and the starting iterate  $\boldsymbol{\sigma}_{n+1}^{(j,0)}$  takes the role of the trial stress.

**Algorithm**  $[\boldsymbol{\sigma}_{n+1}^{(j)}, \boldsymbol{\varepsilon}_{n+1}^{p,(j)}, \mathbf{C}_{n+1}^{ep,(j)}, \alpha_{n+1}^{(j)}] = \text{CUTTING\_PLANE}(\boldsymbol{\varepsilon}_{n+1}^{(j-1)}, \boldsymbol{\varepsilon}_n^p, \alpha_n, \Phi)$

1. Trial values

$$\begin{aligned}\boldsymbol{\sigma}_{n+1}^{(j,0)} &:= \mathbf{C}^{el} : (\boldsymbol{\varepsilon}_{n+1}^{(j-1)} - \boldsymbol{\varepsilon}_n^p) \\ \boldsymbol{\varepsilon}_{n+1}^{p,(j,0)} &:= \boldsymbol{\varepsilon}_n^p \\ \alpha_{n+1}^{(j,0)} &:= \alpha_n\end{aligned}$$

2. Check yield condition

IF  $\bar{\boldsymbol{\sigma}}_{n+1}^{(j,0)} - Y(\alpha_n) \leq 0$  THEN

$$\text{SET } (\bullet)_{n+1}^{(j)} := (\bullet)_{n+1}^{(j,0)} \text{ AND } \mathbf{C}_{n+1}^{ep,(j)} := \mathbf{C}^{el}$$

AND EXIT

3. Initialization

- a)  $\mathbf{n}_{n+1}^{(0)} := \partial_{\boldsymbol{\sigma}} \Phi|_{\boldsymbol{\sigma}_{n+1}^{(j,0)}} \in \mathbb{R}^{d \times d}$
- b)  $A_{n+1}^{(0)} := \mathbf{n}_{n+1}^{(0)} : \mathbf{C}^{el} : \mathbf{n}_{n+1}^{(0)} + \partial_{\alpha} \Phi|_{\alpha_{n+1}^{(j,0)}}$



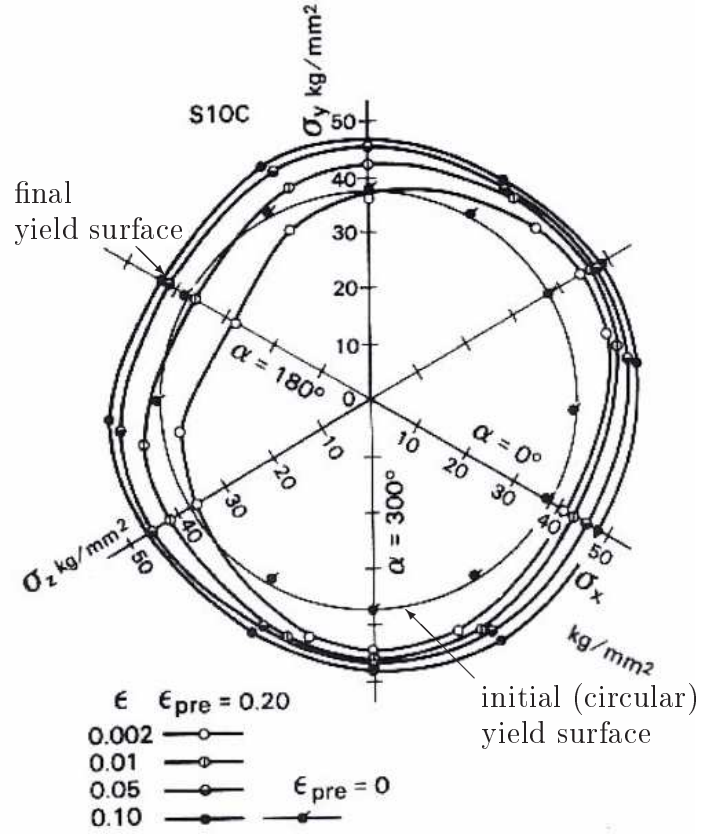


Figure 2.1: Anisotropic yield surfaces for the steel S10C, for different plastic strains, taken from [65], page 94.

c)  $\lambda^{(0)} := 0$

d)  $Y_{n+1}^{(0)} := Y(\alpha_n)$

4. Iteration  $i = 0, 1, 2, \dots$ , see Section 2.5.1

a)  $\Delta\lambda^{(i+1)} := \frac{\bar{\sigma}_{n+1}^{(j,i)} - Y_{n+1}^{(i)}}{A_{n+1}^{(i)}}$

b)  $\sigma_{n+1}^{(j,i+1)} := \sigma_{n+1}^{(j,i)} - \Delta\lambda^{(i+1)} \mathbf{C}^{el} : \mathbf{n}_{n+1}^{(i)}$

c)  $\bar{\sigma}_{n+1}^{(j,i+1)} := \bar{\sigma}(\sigma_{n+1}^{(j,i+1)})$

d)  $\mathbf{n}_{n+1}^{(i+1)} := \partial \sigma \Phi |_{\sigma_{n+1}^{(j,i+1)}}$

e)  $A_{n+1}^{(i+1)} := \mathbf{n}_{n+1}^{(i+1)} : \mathbf{C}^{el} : \mathbf{n}_{n+1}^{(i+1)} + \partial_{\alpha} Y |_{\alpha_{n+1}^{(j,i)}}$

f)  $\lambda^{(i+1)} := \lambda^{(i)} + \Delta\lambda^{(i+1)}$

- g)  $\alpha_{n+1}^{(j,i+1)} := \alpha_{n+1}^{(j,i)} + \Delta\lambda^{(i+1)}$
- h)  $\boldsymbol{\varepsilon}_{n+1}^{p,(j,i+1)} := \boldsymbol{\varepsilon}_{n+1}^{p,(j,i)} + \Delta\lambda^{(i+1)} \mathbf{n}_{n+1}^{(i+1)}$
- i)  $Y_{n+1}^{(i+1)} := Y(\alpha_{n+1}^{(j,i+1)})$
- j) IF  $\bar{\boldsymbol{\sigma}}_{n+1}^{(j,i+1)} - Y_{n+1}^{(i+1)} < \text{TOL}_\Phi$  THEN  
 INNER ITERATION FINISHED, GOTO 5  
 ELSE  $i = i + 1$  GOTO 4  
 ENDIF

5. Elastoplastic module  $\mathbf{C}_{n+1}^{ep,(j)} := \mathbf{C}^{el} - \frac{1}{A_{n+1}}(\mathbf{C}^{el} : \mathbf{n}_{n+1}) \otimes (\mathbf{C}^{el} : \mathbf{n}_{n+1})$

Details about the background can be found in [3]. Only a few points are discussed here, especially the fulfillment of the yield condition and the elastoplastic tangent which are the two most important differences to the RR algorithm in Section (2.4).

### 2.5.1 Fulfillment of the yield condition

An equation for the plastic consistency parameter has to be developed, such that the yield condition is fulfilled in the current Newton step.

$$\Phi_{n+1}^{(j)}(\lambda) = \Phi(\boldsymbol{\sigma}_{n+1}^{(j)}(\lambda), \alpha_{n+1}^{(j)}(\lambda)) \stackrel{!}{=} 0 \quad (2.23)$$

From this and Equations (2.4), (2.5) follows:

$$\begin{aligned} \Phi_{n+1}^{(j)} &= \bar{\boldsymbol{\sigma}}_{n+1}^{(j)}(\lambda) - Y(\alpha_{n+1}^{(j)}(\lambda)) \\ \partial_\lambda \Phi_{n+1}^{(j)} &= \partial \boldsymbol{\sigma} \Phi_{n+1}^{(j)} : \partial_\lambda \boldsymbol{\sigma}_{n+1}^{(j)} - \partial_\alpha Y_{n+1}^{(j)} \partial_\lambda \alpha_{n+1}^{(j)} \\ &= -\partial \boldsymbol{\sigma} \Phi_{n+1}^{(j)} : \mathbf{C}_{n+1}^{el} : \partial \boldsymbol{\sigma} \Phi_{n+1}^{(j)} - \partial_\alpha Y_{n+1}^{(j)} \end{aligned} \quad (2.24)$$

The last equation is obtained with the aid of

$$\begin{aligned} \boldsymbol{\varepsilon}_{n+1}^{p,(j)} - \boldsymbol{\varepsilon}_n^p &= \lambda \partial \boldsymbol{\sigma} \Phi_{n+1}^{(j)} \quad (\text{backward Euler scheme for Equation (2.1)}) \\ \boldsymbol{\sigma}_{n+1}^{(j)} - \boldsymbol{\sigma}_n &= \mathbf{C}^{el} : [\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_{n+1}^{p,(j)} - \boldsymbol{\varepsilon}_n + \boldsymbol{\varepsilon}_n^p] \\ &= \mathbf{C}^{el} : [\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n - \lambda \partial \boldsymbol{\sigma} \Phi_{n+1}^{(j)}] \\ \partial_\lambda \boldsymbol{\sigma}_{n+1}^{(j)} &= -\mathbf{C}^{el} : \partial \boldsymbol{\sigma} \Phi_{n+1}^{(j)} \end{aligned}$$

Note that the total strain  $\boldsymbol{\varepsilon}_{n+1}$  does not depend on  $\lambda$ . So, with the aid of (2.24), the Newton scheme for finding the root of  $\Phi$  reads like this:

$$\partial_\lambda \Phi_{n+1}^{(j)}|_{\lambda^{(i)}} \Delta\lambda^{(i+1)} = -\Phi_{n+1}^{(j)}(\lambda^{(i)})$$

with

$$\Delta\lambda^{(i+1)} = \frac{\Phi_{n+1}^{(j)}(\lambda^{(i)})}{\left[ \partial \boldsymbol{\sigma} \Phi_{n+1}^{(j)} : \mathbf{C}^{el} : \partial \boldsymbol{\sigma} \Phi_{n+1}^{(j)} + \partial_\alpha Y_{n+1}^{(j)} \right]_{\lambda^{(i)}}}$$

## 2.5.2 Discussion

One of the problems arising here is that the yield condition is fulfilled but  $\mathbf{C}_{n+1}^{ep,(j)}$  in Point 5 of Algorithm CUTTING\_PLANE is not the consistent material tangent. Indeed one can see that (2.12) is not fulfilled by the tangent cutting plane algorithm presented here. In fact it is derived from the time-continuous tangent

$$\mathbf{C}^{ep} = \frac{\partial \dot{\boldsymbol{\sigma}}}{\partial \dot{\boldsymbol{\varepsilon}}}$$

see for example [3]. This is done because it is very difficult to obtain a closed expression for the final  $\boldsymbol{\sigma}_{n+1}^{(j)}$ , see line 4b, since there is no explicit expression for a root  $\lambda$  of the nonlinear Equation (2.23). As it is shown in the results in Section 2.8, no good convergence is achievable using this tangent, even in the case of the simple von Mises yield function.

## 2.6 A Gauss point section: Sequential quadratic programming

In this section, another method for solving the constrained optimization problem is given, namely sequential quadratic programming (SQP). In [105] the method is described and applied to perfect plasticity (no hardening,  $\Phi = \Phi(\boldsymbol{\sigma})$ ). The basic idea is to use the fact that it is not necessary to fulfill the yield condition in each semismooth Newton step. The Newton iteration is replaced by a sequence of quadratic minimization problems with linearized constraints ('linearized projection step'). The advantage of the SQP method is the large flexibility which allows to enhance it to more general plasticity models, where the generalization of the Radial Return algorithm is not straightforward. Another important point is the mesh independent convergence of the SQP loop.

### 2.6.1 Definitions

Some definitions have to be made: Let  $\mathbb{K}$  be the set of admissible stresses

$$\mathbb{K} := \{\boldsymbol{\tau} \in \mathbb{S}^{d \times d} : \Phi(\boldsymbol{\tau}) \leq 0\}$$

Let  $\mathbb{S}^{d \times d}$  denote the subset of symmetric matrices in  $\mathbb{R}^{d \times d}$ . Some variables for the linearized projection step  $j$ , namely the linearized flow rule  $\Phi_{n+1}^{(j)}$ , the quadratic dual energy functional  $\mathcal{E}_{n+1}^{(j)}$ , the elastoplastic module  $\mathbf{C}_{n+1}^{sqp,(j)}$ , the half-space  $\mathbb{K}_{n+1}^{(j)}$  of admissible stresses with respect to  $\Phi_{n+1}^{(j)}$ , the inner product  $e_{n+1}^{(j)}(\cdot, \cdot)$  and the associated energy  $E_{n+1}^{(j)}$

are defined by

$$\Phi_{n+1}^{(j)}(\boldsymbol{\sigma}) := \Phi(\boldsymbol{\sigma}_{n+1}^{(j-1)}) + \partial_{\boldsymbol{\sigma}}\Phi|_{\boldsymbol{\sigma}_{n+1}^{(j-1)}} : (\boldsymbol{\sigma} - \boldsymbol{\sigma}_{n+1}^{(j-1)}) \quad (2.25)$$

$$\mathcal{E}_{n+1}^{(j)}(\boldsymbol{\sigma}) := \int_{\Omega} E_{n+1}^{(j)} \left( \boldsymbol{\sigma} - \mathbf{C}_{n+1}^{sqp,(j)} : (\boldsymbol{\varepsilon}_n^p + \lambda^{(j-1)} \partial_{\boldsymbol{\sigma}}\boldsymbol{\sigma}\Phi|_{\boldsymbol{\sigma}_{n+1}^{(j-1)}} : \boldsymbol{\sigma}_{n+1}^{(j-1)}) \right) \quad (2.26)$$

$$\mathbf{C}_{n+1}^{sqp,(j)} := (\mathbf{I} + \lambda^{(j-1)} \mathbf{C}^{el} \partial_{\boldsymbol{\sigma}}\boldsymbol{\sigma}\Phi|_{\boldsymbol{\sigma}_{n+1}^{(j-1)}})^{-1} \mathbf{C}^{el} \quad (2.27)$$

$$\mathbb{K}_{n+1}^{(j)} := \{\boldsymbol{\tau} \in \mathbb{S}^{d \times d} : \Phi_{n+1}^{(j)}(\boldsymbol{\tau}) \leq 0\} \quad (2.28)$$

$$e_{n+1}^{(j)}(\boldsymbol{\sigma}, \boldsymbol{\tau}) := \boldsymbol{\sigma} : (\mathbf{C}_{n+1}^{sqp,(j)})^{-1} : \boldsymbol{\tau} \quad (2.29)$$

$$E_{n+1}^{(j)}(\boldsymbol{\sigma}) := \frac{1}{2} \boldsymbol{\sigma} : (\mathbf{C}_{n+1}^{sqp,(j)})^{-1} : \boldsymbol{\sigma} \quad (2.30)$$

The trial stress  $\boldsymbol{\theta}_{n+1}^{(j)}$  is defined by:

$$\boldsymbol{\theta}_{n+1}^{(j)}(\mathbf{u}) := \mathbf{C}_{n+1}^{sqp,(j)} : \left( \boldsymbol{\varepsilon}(\mathbf{u}) - \boldsymbol{\varepsilon}_n^p + \lambda^{(j-1)} \partial_{\boldsymbol{\sigma}}\boldsymbol{\sigma}\Phi|_{\boldsymbol{\sigma}_{n+1}^{(j-1)}} : \boldsymbol{\sigma}_{n+1}^{(j-1)} \right) \quad (2.31)$$

## 2.6.2 Derivation of the SQP step

Starting point of the SQP step is the original system containing flow rule, complementary conditions and equilibrium which is to be solved at the end of the load step:

$$\boldsymbol{\sigma}_{n+1} - \mathbf{C}^{el} : (\boldsymbol{\varepsilon}(\mathbf{u}_{n+1}) - \boldsymbol{\varepsilon}_n^p + \lambda \partial_{\boldsymbol{\sigma}}\boldsymbol{\sigma}\Phi|_{\boldsymbol{\sigma}_{n+1}}) = \mathbf{0} \quad (2.32)$$

$$\Phi(\boldsymbol{\sigma}_{n+1}) \leq 0, \quad \lambda \Phi(\boldsymbol{\sigma}_{n+1}) = 0, \quad \lambda \geq 0 \quad (2.33)$$

$$\int_{\Omega} \boldsymbol{\sigma}_{n+1} : \boldsymbol{\varepsilon}(\mathbf{v}) = l_{n+1}^{ext}(\mathbf{v}), \quad \mathbf{v} \in [H_0^1(\Omega)]^d \quad (2.34)$$

Linearizing Equation (2.32) with respect to  $(\boldsymbol{\sigma}_{n+1}, \lambda, \mathbf{u}_{n+1})$ , using the definitions (2.25) and (2.31) and using the equality

$$\partial_{\boldsymbol{\sigma}}\Phi|_{\boldsymbol{\sigma}_{n+1}^{(j-1)}} = \partial_{\boldsymbol{\sigma}}\Phi_{n+1}^{(j)}|_{\boldsymbol{\sigma}_{n+1}^{(j)}}$$

( $\Phi_{n+1}^{(j)}$  is linear), one arrives after some algebraic manipulations at the result which will be discussed in Section 2.6.3. Equation (2.33) and (2.34) are reformulated with the linearized flow function and the linearized material response, respectively.

## 2.6.3 The SQP step and the algorithm

So in each SQP step, the linearized flow rule and the linearized complementary conditions

$$\boldsymbol{\sigma}_{n+1}^{(j)} - \boldsymbol{\theta}_{n+1}^{(j)} - \lambda^{(j)} \mathbf{C}_{n+1}^{sqp,(j)} : \partial_{\boldsymbol{\sigma}}\Phi_{n+1}^{(j)}|_{\boldsymbol{\sigma}_{n+1}^{(j)}} = \mathbf{0} \quad (2.35)$$

$$\Phi_{n+1}^{(j)} \leq 0, \quad \lambda^{(j)} \Phi_{n+1}^{(j)} = 0, \quad \lambda^{(j)} \geq 0 \quad (2.36)$$

are solved at each integration point for  $(\boldsymbol{\sigma}_{n+1}^{(j)}, \lambda^{(j)})$  and this linearized material response is inserted into the equilibrium equation:

$$\int_{\Omega} \boldsymbol{\sigma}_{n+1}^{(j)} : \boldsymbol{\varepsilon}(\mathbf{v}) = l_{n+1}^{ext}(\mathbf{v}), \quad \mathbf{v} \in [H_0^1(\Omega)]^d \quad (2.37)$$

The System (2.35)-(2.37) represents indeed a SQP step as it is equivalent to:

Find a minimizer  $\boldsymbol{\sigma}_{n+1}^{(j)}$  of  $\mathcal{E}_{n+1}^{(j)}(\cdot)$  subject to the linear constraints  $\boldsymbol{\sigma} \in \mathbb{K}_{n+1}^{(j)}$  and the equilibrium equation.

In contrary to  $\boldsymbol{\sigma} \in \mathbb{K}$ , the constraint  $\boldsymbol{\sigma} \in \mathbb{K}_{n+1}^{(j)}$  is indeed linear since  $\mathbb{K}_{n+1}^{(j)}$  is a half-space. Lemma 7 in [105] states that a unique solution  $\boldsymbol{\sigma}_{n+1}^{(j)}$  of this problem exists since  $\mathcal{E}_{n+1}^{(j)}(\cdot)$  is uniformly convex. So (without giving a formal proof) mesh-independent convergence can be expected.

The nonlinear problem (2.37) is to be solved with an inner Newton iteration. Fortunately, the application of an inexact method is possible: It is not necessary to iterate the inner Newton loop until convergence in each SQP step. Further inexact methods will be presented in the following chapters. SQP would insofar fit very well to the divide-and-conquer approach of this work: The difficulties of fulfilling the yield condition and the equilibrium condition are separated in such a way that computational cost can be reduced.

A consistent tangent module has to be computed. For this purpose, problem (2.37) is slightly reformulated with the aid of Lemma 1 in [105], which states that the solution  $\boldsymbol{\sigma}_{n+1}^{(j)}$  of [(2.35),(2.36)] is uniquely determined as

$$\boldsymbol{\sigma}_{n+1}^{(j)} = \Pi_{n+1}^{(j)}(\boldsymbol{\theta}_{n+1}^{(j)}) \quad (2.38)$$

where  $\Pi_{n+1}^{(j)}$  is the  $e_{n+1}^{(j)}(\cdot, \cdot)$ -orthogonal projection onto the half-space  $\mathbb{K}_{n+1}^{(j)}$ . Then

$$f_{n+1}^{int,(j)}(\mathbf{u}, \mathbf{v}) := \int_{\Omega} \Pi_{n+1}^{(j)}(\boldsymbol{\theta}_{n+1}^{(j)}(\mathbf{u})) : \boldsymbol{\varepsilon}(\mathbf{v}) - l_{n+1}^{ext}(\mathbf{v}) \stackrel{!}{=} 0 \quad (2.39)$$

is equivalent to (2.37). By deriving  $\partial \mathbf{u} f_{n+1}^{int,(j)}|_{\mathbf{u}_{n+1}^{(j,i)}}$ , one gets the consistent tangent module

$$\mathbf{C}_{n+1}^{ep,(j,i)} := \frac{\partial \Pi_{n+1}^{(j)}}{\partial \boldsymbol{\theta}} \Big|_{\boldsymbol{\theta}_{n+1}^{(j)}(\mathbf{u}_{n+1}^{(j,i)})} \mathbf{C}_{n+1}^{sqp,(j)} \quad (2.40)$$

in the inner Newton step  $i$ . Now, the solution algorithm is given; note that  $j$  is in this context the index of the SQP iteration and  $i$  index of the inner Newton iteration.

**Algorithm**  $[\boldsymbol{\sigma}_{n+1}, \boldsymbol{\varepsilon}_{n+1}^p] = \text{SQP}(\boldsymbol{\varepsilon}_n^p, \Phi, \lambda_n, \boldsymbol{\sigma}_n, \mathbf{u}_n)$

1. Init.  $\mathbf{u}_{n+1}^{(0)} := \mathbf{u}_n, \boldsymbol{\sigma}_{n+1}^{(0)} := \boldsymbol{\sigma}_n, \lambda_{n+1}^{(0)} := \lambda_n$  and  $j := 1$

2. Init.  $\mathbf{u}_{n+1}^{(j,0)} := \mathbf{u}_{n+1}^{(j-1)}$  and  $i := 1$

3. Compute the trial stress

$$\boldsymbol{\theta}_{n+1}^{(j)}(\mathbf{u}_{n+1}^{(j,i-1)})$$

according to Equation (2.31) and the linearized projection

$$\boldsymbol{\sigma}_{n+1}^{(j,i-1)} = \Pi_{n+1}^{(j)}(\boldsymbol{\theta}_{n+1}^{(j)}(\mathbf{u}_{n+1}^{(j,i-1)}))$$

according to Equation (2.38) for the Newton residual of the nonlinear problem (2.39).

4. If  $|f_{n+1}^{int,(j,i-1)}|$  is not small enough then goto step 5 else goto step 6.

5. Compute  $\mathbf{C}_{n+1}^{ep,(j,i)}$  according to (2.40) and compute  $\Delta \mathbf{u}_{n+1}^{(j,i)}$  as Newton step in the solution of the nonlinear problem (2.39) with the residual, computed in step (3). Update

$$\mathbf{u}_{n+1}^{(j,i)} := \mathbf{u}_{n+1}^{(j,i-1)} + \Delta \mathbf{u}_{n+1}^{(j,i)}$$

set  $i := i + 1$  and goto step 3.

6. Set  $\mathbf{u}_{n+1}^{(j)} := \mathbf{u}_{n+1}^{(j,i)}$ ,  $\boldsymbol{\sigma}_{n+1}^{(j)} := \boldsymbol{\sigma}_{n+1}^{(j,i)}$  and  $\lambda^{(j)} := \lambda^{(j,i)}$

7. If

$$\|\boldsymbol{\sigma}_{n+1}^{(j)} - \mathbf{C}^{el} : (\boldsymbol{\varepsilon}(\mathbf{u}_{n+1}^{(j)}) - \boldsymbol{\varepsilon}_n^p + \lambda^{(j)} \partial \boldsymbol{\sigma} \Phi |_{\boldsymbol{\sigma}_{n+1}^{(j)}})\|$$

which is the residual of Equation (2.32), is not small enough, set  $j := j + 1$  and goto step 3 else goto step 8

8. Set  $\mathbf{u}_{n+1} := \mathbf{u}_{n+1}^{(j)}$ ,  $\boldsymbol{\sigma}_{n+1} := \boldsymbol{\sigma}_{n+1}^{(j)}$ ,  $\lambda := \lambda^{(j)}$  and  $\boldsymbol{\varepsilon}_{n+1}^p := \boldsymbol{\varepsilon}_n^p + \lambda \partial \boldsymbol{\sigma} \Phi |_{\boldsymbol{\sigma}_{n+1}}$  and exit.

## 2.7 Comparison of the Gauss point algorithms

The different algorithms for stress integration are compared in some headwords. The generalized projection method will be explained in the context of finite plasticity. There are more possibilities to solve the constrained optimization problem which are not listed here, for example the interior point method which has been applied to plasticity in [69].

### Consistently linearized RR, see Section 2.4

- For certain types of yield functions easy to linearize consistently.
- Difficult to adapt to general yield functions. In the general case, consistent linearization difficult.

**TCP, see Section 2.5**

- Applicable to many kinds of flow laws, including anisotropic plasticity.
- Consistent linearization very difficult.

**SQP, see Section 2.6**

- The fact that the yield condition does not need to be fulfilled in each Newton step can be used to save computation time.
- Outer SQP loop and inner Newton loop.
- Applicable to many kinds of flow laws, including anisotropic plasticity since in each SQP step a back projection onto a simple half-space is performed.
- Mesh independent and good convergence of the outer SQP loop.

**Generalized projection, see Section 8.5**

- Applicable to many kinds of flow laws, including anisotropic plasticity.
- Consistent linearization easy.
- Fulfillment of the yield condition more costly since a Newton iteration has to be performed to compute the stresses and the plastic data.

## 2.8 Numerical results

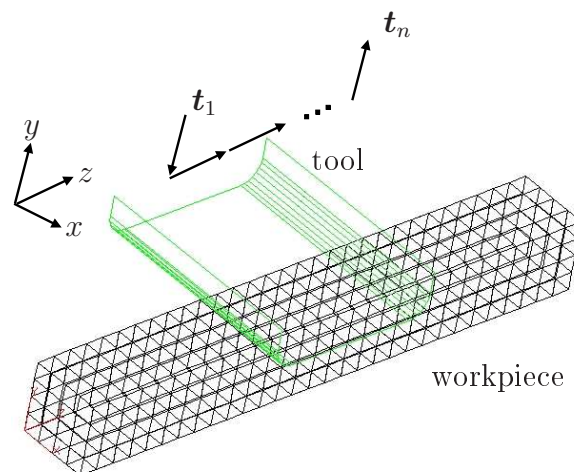


Figure 2.2: Problem setting of the forging simulation.

In this section, TCP and RR are compared by means of an elastoplastic forging-like simulation, depicted in Figure 2.2. The assumptions are J2 plasticity, linear isotropic hardening and small deformations. So, each FE element with TCP, which is formulated for the more general case of anisotropic plasticity, is expected to behave like an element with the consistently formulated RR algorithm. Results for ( $\mathbf{t}_n =$  tool feed in step  $n$ )

$$\mathbf{t}_{1,\dots,5} = \begin{bmatrix} 0 \\ -0.05 \\ 0 \end{bmatrix}, \mathbf{t}_{6,\dots,50} = \begin{bmatrix} 0 \\ 0 \\ 0.1 \end{bmatrix}$$

are shown in Figure 2.3. To compare the convergence of the Newton scheme between RR and TCP and to see how the element behavior is in the case of springback, a slightly modified example is considered. The results are given in Figure 2.4.

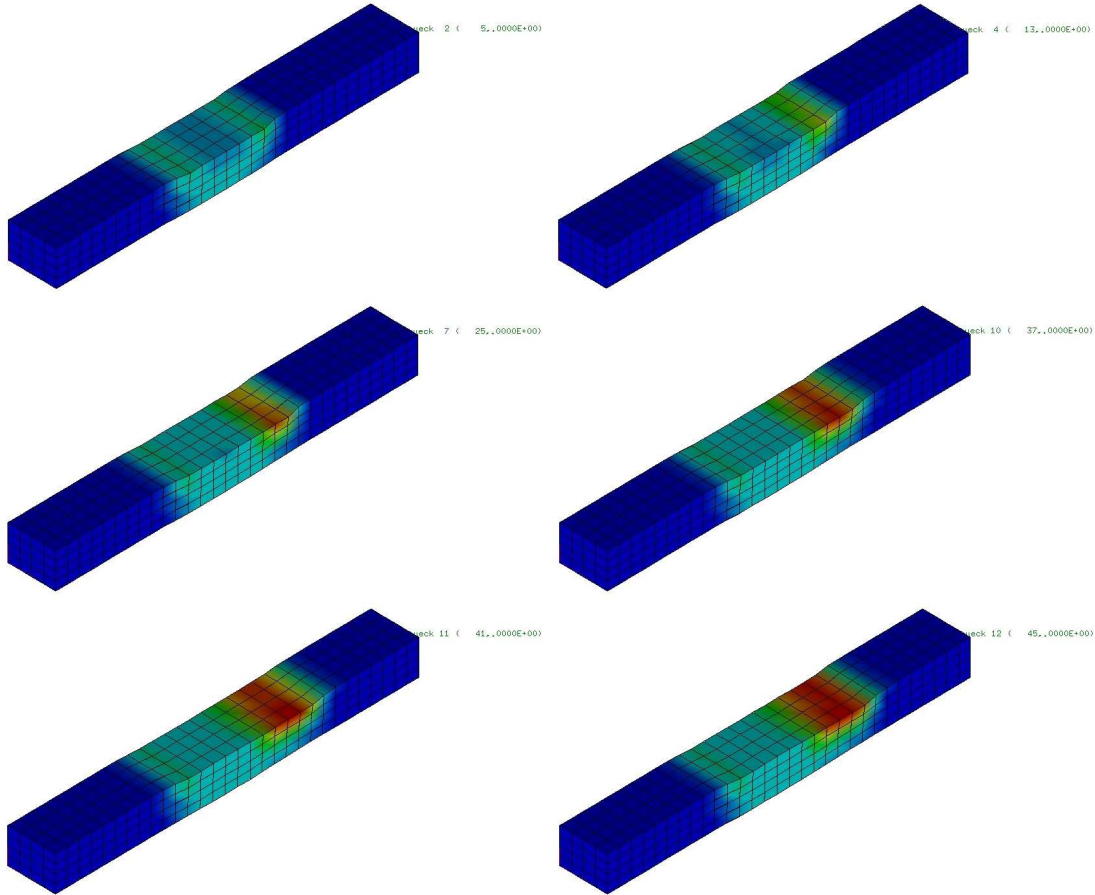


Figure 2.3: Steps 5,13,25,37,41 and 45 of the forging simulation, the evolution of the hardening parameter  $\alpha$  is shown.

$$\mathbf{t}_{1,\dots,5} = \begin{bmatrix} 0 \\ -0.05 \\ 0 \end{bmatrix}, \mathbf{t}_{6,\dots,20} = \begin{bmatrix} 0 \\ 0 \\ 0.1 \end{bmatrix}, \mathbf{t}_{23} = \begin{bmatrix} 0 \\ 0.05 \\ 0 \end{bmatrix}$$

Obviously, the RR element formulation performs much better, especially in the springback situation.



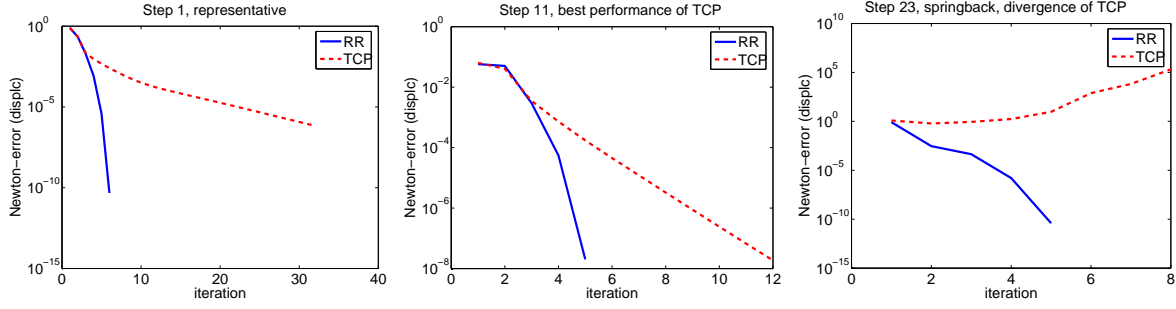


Figure 2.4:  $\frac{\|\Delta \mathbf{U}^{(j)}\|}{\|\Delta \mathbf{U}^{(1)}\|}$  plotted over the Newton iteration ( $j$ ).

## 2.9 Some tensor calculus

Some rules from tensor calculus are given in this section. Let  $\mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{S}, \mathbf{T}$  and  $\boldsymbol{\xi}$  be arbitrary second order tensors, let  $\mathbf{1}$  be the second order identity tensor and let  $\mathbf{I}$  be the fourth order symmetric identity tensor.

$$\begin{aligned}
 \text{tr}(\mathbf{B}) &:= \mathbf{1} : \mathbf{B} \\
 (\mathbf{B} \otimes \mathbf{C})\mathbf{D} &:= (\mathbf{C} : \mathbf{D})\mathbf{B} \\
 \partial_{\boldsymbol{\varepsilon}} \text{tr}(\boldsymbol{\varepsilon}) &= \mathbf{1} \\
 \frac{\partial(\alpha \mathbf{S})}{\partial \mathbf{T}} &= \mathbf{S} \otimes \frac{\partial \alpha}{\partial \mathbf{T}} + \alpha \frac{\partial \mathbf{S}}{\partial \mathbf{T}} \\
 \mathbf{n}(\boldsymbol{\xi}) &:= \frac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|} \\
 \partial_{\boldsymbol{\varepsilon}} \|\boldsymbol{\xi}\| &= \mathbf{n} : \partial_{\boldsymbol{\varepsilon}} \boldsymbol{\xi} \\
 \Rightarrow \partial_{\boldsymbol{\xi}} \mathbf{n} &= \frac{1}{\|\boldsymbol{\xi}\|} (\mathbf{I} - \mathbf{n} \otimes \mathbf{n})
 \end{aligned}$$

## 2.10 Summary

It was shown in this chapter that the importance of correct derivatives in the material stiffness matrix cannot be overestimated in the context of metal forming. In principle similar good convergence is not only achievable in the very simple case of J2 plasticity but also in the case of a general nonlinear isotropic and kinematic hardening law, see [97] page 146. Several possibilities to solve the constrained optimization problem of plasticity were shown and compared.



# 3 Classical 1-grid discretization: Contact

The interaction between tool and workpiece is modelled as one-body contact problem with a rigid obstacle. Friction is neglected throughout this work but Tresca and Coulomb friction can be incorporated in any of the algorithms presented here [56]. For extending the usability of implicit FE codes for large scale forming simulations, the computation time has to be decreased dramatically. In principle, this can be achieved by using iterative solvers. In order to facilitate the use of this kind of solvers, one needs a contact algorithm which does not deteriorate the condition number of the system matrix and therefore does not slow down the convergence of iterative solvers like penalty formulations do. Additionally, an algorithm is desirable which does not blow up the size of the system matrix like methods using standard Lagrange multipliers. The work detailed in [19] shows that a contact algorithm based on a primal-dual active set strategy (pdASeS) provides these advantages and is therefore highly efficient with respect to computation time in combination with fast iterative solvers, especially algebraic multigrid methods.

## 3.1 Contact algorithms in the literature

Efficient contact algorithms have been developed in recent years, see [75, 92, 110, 111, 112] and the references therein. For more complex contact problems where the contact area is not known in advance, active set strategies are well established nowadays. Four important variants of them are listed with their advantages (+) and disadvantages (−), see for example Willner [106] and the references therein.

### Penalty formulation

- + Pure displacement based formulation, no change in the system size.
- − For small values of the penalty parameter, the penetration is quite large.
- − For large values of the penalty parameter, the condition number of the system matrix is deteriorated.

### Standard Lagrange multipliers

- + Exact fulfillment of the corresponding discrete non-penetration condition.
- + No deterioration of the condition number of the stiffness matrix.

- The multipliers are additional unknowns, the size of the system matrix increases.
- Only global elimination of the LM is possible leading to dense matrices.

#### **Perturbed Lagrangian formulation**

- + Very stiff constitutive laws pose no numerical problems.
- + No deterioration of the condition number of the stiffness matrix.
- Mixed method, the size of the system matrix increases.
- Only global elimination of the LM is possible leading to dense matrices.

#### **Augmented Lagrangian formulation**

- + Nearly exact fulfillment of the non-penetration condition with low penalties.
- + No change in the system size.
- + No deterioration of the condition number of the stiffness matrix.
- Higher numerical costs due to extra augmentation loop.

The algorithm, proposed here, which goes back to [2, 53], unifies the advantages of Lagrange multipliers and the penalty method:

#### **Dual Lagrange multipliers**

- + No change in the system size due to the use of dual Lagrange multipliers, which makes the method very useful even for large commercial FE packages.
- + Exact fulfillment of the geometric constraints of the tool in a weak integral sense.
- + No deterioration of the condition number of the stiffness matrix.

There exists in the literature a large number of alternative methods such as e.g. FETI type algorithms [34] and monotone methods [67, 70]. In this work, the comparison between the primal-dual active set strategy and the penalty method is emphasized, since one can see from the survey given above that the problems of the penalty formulation are somehow preserved in the other algorithms and just assume a different shape.

## 3.2 The primal dual active set strategy with dual Lagrange multipliers

In the following, the primal-dual active set strategy will be sometimes referred to as active set strategy, keeping in mind that there are different kinds of active set strategies given in the literature. It turns out that this strategy is very attractive for the use in metal forming, especially in combination with iterative solvers, as their convergence behavior strongly depends on the condition number of the system matrix. Due to steady increase in the problem size and the need for a robust, highly accurate and scalable solution method, algebraic multigrid methods [31, 102] are the main focus of this investigation. In [19], the active set strategy was adapted for nonlinear material behavior and some details were given about the implementation in LARSTRAN. It will be shown that in combination with the active set strategy, the well-known good complexity of multigrid methods can be achieved. By using algebraic multigrid, problem dependency can be overcome such that the used solver provides a black-box character. It will be shown that the Newton and the solver loop of the algebraic multigrid method can be merged. This inexact strategy provides a further decrease in computation time.

To handle the nonlinearity of the Karush–Kuhn–Tucker contact conditions, a primal-dual active set strategy based on dual Lagrange multipliers [57, 108] is applied. This can be interpreted as a Newton method [2, 53]. In contrast to the commonly used penalty method [61], the active set strategy allows to adjust the geometric constraints of the tool exactly in a weak integral sense. This point will be explained in more detail in Chapter 4. The contact stress can be easily recovered from the displacements in a variationally consistent way and does not depend on a tuning parameter.

### 3.2.1 Basic idea

Just to give a first impression, a simple rectangular workpiece is considered which is indented by a rigid, hemispherical punch, see Figure 3.1a. By  $\mathbb{S}$  the set of potential contact nodes is denoted, see Figure 3.1a. Here,  $j$  is the iteration index for the active set loop. Let  $\mathbb{A}_j \subset \mathbb{S}$  be the active set, for which the body is in contact with the obstacle. The complementary set  $\mathbb{I}_j := \mathbb{S} - \mathbb{A}_j$  is called the inactive set. The following simple test example shows how the sets  $\mathbb{I}_{j+1}$  and  $\mathbb{A}_{j+1}$  are correctly determined. The following situation is assumed, see Figure 3.1b, in the  $j$ -th step: Nodes  $\{1,2\}$  are active, nodes  $\{3,4\}$  are inactive. The tool geometry is then adjusted to the active nodes, no matter if the active set is already correct or not, see Figure 3.1c. Now the nodal forces and the detected penetration are used to update the active and inactive node sets:

$$\begin{aligned}\mathbb{A}_{j+1} &:= \{p \in \mathbb{I}_j : p \text{ penetrates}\} \cup \{p \in \mathbb{A}_j : p \text{ under compression}\} \\ \mathbb{I}_{j+1} &:= \{p \in \mathbb{I}_j : \text{no penetration}\} \cup \{p \in \mathbb{A}_j : p \text{ under tension}\}\end{aligned}$$

Then the restrictions from the tool geometry are again adjusted to the correct active nodes and the inactive nodes are released, see Figure 3.1d. The step from (c) to (d) is



In order to avoid a case distinction in the algebraic part, only  $d = 3$  is considered from now on. It is assumed that a workpiece  $\Omega \subset \mathbb{R}^3$  comes into contact with a rigid tool with the boundary  $\Gamma^{tool}$ . The following notations are introduced: The part of  $\partial\Omega$  which comes potentially into contact with  $\Gamma^{tool}$  is called  $\Gamma^{con}$ . By  $\mathbf{x}_{ref}$  the coordinates of an arbitrary point on the potential contact zone in the so called contact reference configuration are denoted. With respect to this special configuration, all important contact variables like the gap and the contact normal are computed. Several choices are possible:

- $\mathbf{x}_{ref} = \mathbf{x}_0$ : The contact reference configuration is the reference configuration. Obviously this will lead to good results only in the regime of small deformations.
- $\mathbf{x}_{ref} = \mathbf{x}_{n+1}$ : The most accurate and also most difficult way to choose  $\mathbf{x}_{ref}$  is to choose the unknown, solution dependent, current position of the material point. Yang et al. pursue this way in [115]. Since all contact variables are solution dependent, complicated linearizations have to be performed there. It turns out that this formulation is extremely stable even if large load steps are performed. Since there are technical restrictions on the load step size anyway (the movement of the tool has to be resolved), this method is not pursued in this work.

To avoid difficulties arising from large deformations, the contact reference configuration must be chosen in such a way that the deformation from there to the current configuration is small. So in this work, the two following options are chosen:

- $\mathbf{x}_{ref} = \mathbf{x}_n$ : Like in an updated Lagrange manner, it is a good choice to take the last time step as contact reference configuration. This choice is assumed in each of the following chapters except Chapter 4. This performs well under the small-time-step assumption, as the numerical examples show. Moreover, it is always possible to check again the newly computed final contact normal and the final gap with the reference normal and the reference gap, see Figure 3.2. If the forbidden halfspace shows a major discrepancy with the tool geometry, then one should redo the computation with updated normal and gap.
- $\mathbf{x}_{ref} = \mathbf{x}_{n+1}^{(j)}$ : A more up-to-date choice is the last Newton iteration. This is done in Chapter 4. It is important to note that this procedure is different from [115], because a full linearization is not performed. This method is more in accordance with a fixpoint iteration.

The point on the tool boundary  $\Gamma^{tool}$  which corresponds to  $\mathbf{x}_{ref}$ , is called  $\mathbf{x}_{tool}(\mathbf{x}_{ref})$ . If  $\Gamma^{tool}$  can be described by a sufficiently smooth function, the inward-looking normal  $\boldsymbol{\nu}$  can be defined on every point on  $\Gamma^{tool}$  such that the following closest point relation holds:

$$\begin{aligned} \|\mathbf{x}_{tool}(\mathbf{x}_{ref}) - \mathbf{x}_{ref}\|_2 &\rightarrow \min, \\ \mathbf{x}_{tool}(\mathbf{x}_{ref}) - \mathbf{x}_{ref} &= \hat{g}(\mathbf{x}_{ref}, t_{n+1})\boldsymbol{\nu}(\mathbf{x}_{ref}, t_{n+1}) \end{aligned} \quad (3.1)$$

where  $\hat{g}(\mathbf{x}_{ref}, t_{n+1})$  denotes the gap between  $\mathbf{x}_{tool}(\mathbf{x}_{ref})$  and  $\mathbf{x}_{ref}$  with the tool position at time  $t_{n+1}$ . The gap has the meaning of the distance between the tool and the structure

in the contact reference configuration. The sign of this gap function will define whether a material point on the potential contact zone will come into contact with the rigid obstacle or not. A positive value of  $\hat{g}(\mathbf{x}, t)$  indicates that there is no contact, whereas a negative value describes an interpenetration, which is prohibited. In the following, it will be always written  $\boldsymbol{\nu}$  instead of  $\boldsymbol{\nu}(\mathbf{x}_{\text{ref}}, t_{n+1})$  and  $\hat{g}$  instead of  $\hat{g}(\mathbf{x}_{\text{ref}}, t_{n+1})$ , and it will be assumed that the relation (3.1) holds. By  $\mathbf{u}$ , the unknown displacements  $\mathbf{x}_{n+1} - \mathbf{x}_0$  are denoted, and by  $\mathbf{u}(\mathbf{x}_{\text{ref}})$  the known displacements  $\mathbf{x}_{\text{ref}} - \mathbf{x}_0$ . Finally, the scalar valued normal part  $\sigma_\nu(\mathbf{u})$  of the stresses on  $\Gamma^{\text{con}}$  and the tangential stress vector  $\boldsymbol{\sigma}_\tau(\mathbf{u}) \in \mathbb{R}^d$  are defined by

$$\sigma_\nu(\mathbf{u}) := (\boldsymbol{\sigma}(\mathbf{u}) \boldsymbol{\nu}) \cdot \boldsymbol{\nu}, \quad \boldsymbol{\sigma}_\tau(\mathbf{u}) := \boldsymbol{\sigma}(\mathbf{u}) \boldsymbol{\nu} - \sigma_\nu(\mathbf{u}) \boldsymbol{\nu} \quad (3.2)$$

The well known contact conditions for all  $\mathbf{x}_{\text{ref}}$  in the potential contact zone  $\Gamma^{\text{con}}$  are given by (the time index  $n$  is omitted in this subsection):

$$\begin{aligned} (\mathbf{u} - \mathbf{u}(\mathbf{x}_{\text{ref}})) \cdot \boldsymbol{\nu} - \hat{g} &\leq 0 \\ \sigma_\nu(\mathbf{u}) &\leq 0 \\ \sigma_\nu(\mathbf{u}) ((\mathbf{u} - \mathbf{u}(\mathbf{x}_{\text{ref}})) \cdot \boldsymbol{\nu} - \hat{g}) &= 0 \\ \boldsymbol{\sigma}_\tau(\mathbf{u}) &= 0 \end{aligned}$$

Hiding all the known information in

$$g := \hat{g} + \mathbf{u}(\mathbf{x}_{\text{ref}}) \cdot \boldsymbol{\nu}$$

the contact conditions can be written as

$$\mathbf{u} \cdot \boldsymbol{\nu} - g \leq 0 \quad (3.3)$$

$$\sigma_\nu(\mathbf{u}) \leq 0 \quad (3.4)$$

$$\sigma_\nu(\mathbf{u}) (\mathbf{u} \cdot \boldsymbol{\nu} - g) = 0 \quad (3.5)$$

$$\boldsymbol{\sigma}_\tau(\mathbf{u}) = 0 \quad (3.6)$$

The conditions (3.3)-(3.6) can be interpreted in the following way:

(3.3) means that no penetration is allowed.

(3.4) allows only compression but no tension, since there is no adhesion.

(3.5) is the complementary condition that means that either non-zero contact stresses are present and the gap is zero or the gap is non-zero and there are no contact stresses.

(3.6) finally means that the stress in tangential direction is zero, since there is no friction.

The equilibrium is given by (for simplicity volume forces and other external forces are assumed to be zero except from the contact forces): Find  $\mathbf{u} \in [H_0^1(\Omega)]^3$  such that

$$f^{\text{int}}(\mathbf{u}, \mathbf{v}) + \int_{\Gamma^{\text{con}}} \boldsymbol{\lambda} \cdot \mathbf{v} \stackrel{!}{=} 0 \quad (3.7)$$



for arbitrary test functions  $\mathbf{v} \in [H_0^1(\Omega)]^3$  and with the internal forces  $f^{int}$  depending nonlinearly on  $\mathbf{u}$ . The Lagrange multiplier  $\boldsymbol{\lambda} := -\boldsymbol{\sigma}(\mathbf{u})\boldsymbol{\nu}$  is exactly the contact stress, which is necessary to adjust the contact displacements on the contact boundary  $\Gamma^{con}$ . It is introduced as additional unknown to fulfill the contact conditions.

### 3.2.3 Active Set Strategy and Algebraic Multigrid Methods

Later in this chapter, the task will be to solve the linearized equation system (3.24). Multigrid methods have proven to be highly efficient for problems arising from finite element discretizations of partial differential equations, see e.g. [48] and especially for contact problems, see Kornhuber and Krause [68, 70]. Due to the fact that locally refined meshes are widely used throughout engineering practice for such kind of problems, a hierarchical grid structure might be very difficult to achieve on large unstructured grids. For that reason the focus is on algebraic multigrid methods. They preserve the main multigrid properties but require only the system matrix of the (fine) computational grid as input and thus appear to be very attractive.

Algebraic multigrid, first introduced by Brandt, McCormick, Ruge and Stüben in the 1980's, has received steady progress in its development, to become a robust and efficient tool in numerics (see e.g. [31] and the references therein). It has been applied to a wide field of applications, including structural mechanics. Although the initial attempts were restricted to M-matrices, it has been shown that the method works well also for other matrices in many cases. For the according theory, the reader is referred to [16, 95, 102].

A vital property of a method for its usage in engineering is a generalized applicability to different problems. Therefore, the reduction of parameters which have to be chosen in advance is inevitable. Thus, a compromise has to be found to fulfill that demand and still maintain good performance. In order to do so, for the numerical examples in Section 3.3, the classical Ruge-Stüben AMG method is used in a scalar approach to preserve the property of a total black-box solver and ensure a generalized applicability.

In the same way as geometric multigrid, the AMG uses a number of grid levels to achieve an effective relaxation of all error components iteratively in the computed solution. In contrast to its geometric counterpart, algebraic multigrid does not need any geometric mesh information to establish a grid hierarchy. Instead the system matrix entries serve as vertices and the matrix topology determines their connectivity according to ideas from graph theory. All multigrid components including the coarse levels are then computed algebraically. For further acceleration, the AMG will be used as a preconditioner for a conjugate gradient method, which has proven to be very efficient [13, 102]. In connection with the active set strategy, the emerging system matrices are slightly unsymmetric. For that reason a BiCGStab method is used when necessary. For simplicity, the according AMG-preconditioned CG method is denoted simply by AMG method in the rest of this work. In the following, it will be examined how those methods perform in combination with the described active set strategy, and the developed algorithms are discussed.

### 3.2.4 Discretized form

The Lagrange multiplier  $\boldsymbol{\lambda}$  is approximated by

$$\boldsymbol{\lambda}_h = \sum_{i \in \mathbb{S}} \Lambda_i \psi_i \in \mathbb{R}^3 \quad (3.8)$$

with dual Lagrange multipliers  $\psi_i \in \mathbb{M}_h$  and  $\mathbb{S}$  the set of all potential contact nodes on  $\Gamma^{con}$ . For literature about dual Lagrange multipliers, the reader is referred to [41, 108]. Furthermore  $\phi_p$ , restricted to  $\Gamma^{con}$ , and  $\psi_q$  are the scalar-valued basis functions, associated with the node  $p$  respectively node  $q$ . See especially [50] and the references therein for the situation when  $\Gamma^{con}$  is two-dimensional and meshed with quadrilaterals. The algebraic representation of (3.7) has the form

$$\mathbf{F}^{int}(\mathbf{U}) + \mathbf{B}\boldsymbol{\Lambda} = \mathbf{0} \quad (3.9)$$

with

$$\mathbf{B}[p, q] := \int_{\Gamma^{con}} \phi_p \psi_q \mathbf{1}_3, \quad p \in \mathbb{N}, q \in \mathbb{S}$$

where  $\mathbb{N} \supset \mathbb{S}$  is the set of all structure nodes. The biorthogonality of the basis functions yields

$$\int_{\Gamma^{con}} \phi_p \psi_q = \delta_{pq} \int_{\Gamma^{con}} \phi_p \quad (3.10)$$

with the Kronecker symbol  $\delta_{pq}$  understood in the way that

$$\delta_{pq} = \begin{cases} 1 & \text{structure node } p \text{ coincides with potential contact node } q \\ 0 & \text{otherwise} \end{cases}$$

Now, using an appropriate node numbering,  $\mathbf{B}$  has the form

$$\mathbf{B} = (\mathbf{0} \ \mathbf{D})^\top \quad (3.11)$$

Due to (3.10) the entries of the diagonal matrix  $\mathbf{D}$  are given by

$$\mathbf{D}[p, q] = \delta_{pq} \mathbf{1}_3 \cdot \int_{\Gamma^{con}} \phi_p \quad (3.12)$$

Then the contact conditions have to be discretized. The strong pointwise non-penetration condition (3.3) is replaced by the weaker integral condition

$$\int_{\Gamma^{con}} (\mathbf{u} \cdot \boldsymbol{\nu}) \psi_p \leq \int_{\Gamma^{con}} g \psi_p, \quad p \in \mathbb{S} \quad (3.13)$$

If the right hand side is defined by  $G_p$  and using (3.12), the algebraic representation of the weak non-penetration condition can be written as

$$U_{\nu, p} := \mathbf{N}_p^\top \mathbf{D}[p, p] \mathbf{U}_p \leq G_p, \quad p \in \mathbb{S} \quad (3.14)$$

where  $\mathbf{U}_p \in \mathbb{R}^3$  denotes the coefficient vector of  $\mathbf{u}_h$  associated with the vertex  $p$ . The normal vector at the vertex  $p$  is denoted by  $\mathbf{N}_p$ . Condition (3.4) is discretized by  $\Lambda_{\nu,p} \geq 0$  at each vertex  $p \in \mathbb{S}$ , where  $\Lambda_{\nu,p}$  is defined according to (3.14) by

$$\Lambda_{\nu,p} := \mathbf{N}_p^\top \mathbf{D}[p,p] \Lambda_p, \quad \Lambda_p \in \mathbb{R}^3$$

Introducing the tangential part of the Lagrange multiplier  $\boldsymbol{\lambda}$  at the vertex  $p \in \mathbb{S}$  by  $\boldsymbol{\Lambda}_{\tau,p} := \Lambda_p - (\Lambda_p \cdot \mathbf{N}_p) \mathbf{N}_p$ , the discrete algebraic version of (3.3) - (3.6) is:

$$\begin{aligned} U_{\nu,p} &\leq G_p, \quad \Lambda_{\nu,p} \geq 0, \quad \Lambda_{\nu,p}(U_{\nu,p} - G_p) = 0, \quad p \in \mathbb{S} \\ \boldsymbol{\Lambda}_{\tau,p} &= \mathbf{0} \end{aligned} \quad (3.15)$$

### 3.2.5 Nonlinear algorithm for 1-grid discretization

The tangential vectors are given by  $\mathbf{T}_p^\xi \perp \mathbf{N}_p$  and  $\mathbf{T}_p^\eta := \mathbf{T}_p^\xi \times \mathbf{N}_p$  with  $\|\mathbf{N}_p\| = \|\mathbf{T}_p^\xi\| = \|\mathbf{T}_p^\eta\| = 1$ . After introducing

$$\mathbf{T}_\mathbb{S}^{(m)} = \text{diag} \left( \mathbf{T}_p^{(m)} \right) \in \mathbb{R}^{|\mathbb{S}| \times 3|\mathbb{S}|}, \quad m = \xi, \eta, \quad p \in \mathbb{S} \quad (3.16)$$

it can be readily shown, that the system (3.15) is equivalent to the following nonlinear problem:

$$\mathbf{F}^{con} := \begin{bmatrix} \mathbf{F}_\nu^{con} \\ \mathbf{T}_\mathbb{S} \boldsymbol{\Lambda}_\mathbb{S} \end{bmatrix} \stackrel{!}{=} \mathbf{0}$$

with

$$\mathbf{F}_\nu^{con}(\mathbf{U}, \boldsymbol{\Lambda})[p] := \Lambda_{\nu,p} - (\Lambda_{\nu,p} + c(U_{\nu,p} - G_p))_+ = 0, \quad p \in \mathbb{S} \quad (3.17)$$

where  $(x)_+ := \frac{1}{2}(|x| + x)$ . The meaning of the constant  $c$  will be explained in Section 3.2.6. Now the nonlinear equation system (3.9) is expanded to the following one:

$$\begin{aligned} \mathbf{F}^{int}(\mathbf{U}) + \mathbf{B}\boldsymbol{\Lambda} &= \mathbf{0} \\ \mathbf{F}^{con}(\mathbf{U}, \boldsymbol{\Lambda}) &= \mathbf{0} \end{aligned} \quad (3.18)$$

The set of potential contact nodes  $\mathbb{S}$  can be split up into two parts:

$$\mathbb{I}^{(j)} := \{p \in \mathbb{S} : \Lambda_{\nu,p}^{(j)} + c(U_{\nu,p}^{(j)} - G_p) \leq 0\} \quad (3.19)$$

$$\mathbb{A}^{(j)} := \{p \in \mathbb{S} : \Lambda_{\nu,p}^{(j)} + c(U_{\nu,p}^{(j)} - G_p) > 0\} \quad (3.20)$$

where, as before,  $\mathbb{A}^{(j)}$  and  $\mathbb{I}^{(j)}$  refer to the active and inactive set, respectively. The next task is to linearize (3.18) consistently. To do so, the diagonal matrix  $\mathbf{D}$  is decomposed into

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_\mathbb{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_\mathbb{A} \end{bmatrix}$$

Moreover, the global normal matrix  $\mathbf{N}_\mathbb{A} \in \mathbb{R}^{|\mathbb{A}| \times 3|\mathbb{A}|}$ , where  $|\mathbb{A}|$  denotes the number of vertices in  $\mathbb{A}$ , is defined by

$$\mathbf{N}_\mathbb{A} := \text{diag} (w_{pp} \mathbf{N}_p), \quad p \in \mathbb{A} \quad (3.21)$$

The weighting factor  $w_{pp}$  is an abbreviation for  $\mathbf{D}[p, p]_{1,1} = \dots = \mathbf{D}[p, p]_{3,3} = \int_{\Gamma^{\text{con}}} \phi_p$ . Finally, consistent linearization gives:

$$\begin{bmatrix} \mathbf{K}_{\text{NN}} & \mathbf{K}_{\text{NI}} & \mathbf{K}_{\text{NA}} & \mathbf{0} & \mathbf{0} \\ \mathbf{K}_{\text{IN}} & \mathbf{K}_{\text{II}} & \mathbf{K}_{\text{IA}} & \mathbf{D}_{\text{I}} & \mathbf{0} \\ \mathbf{K}_{\text{AN}} & \mathbf{K}_{\text{AI}} & \mathbf{K}_{\text{AA}} & \mathbf{0} & \mathbf{D}_{\text{A}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}_{\text{I}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{N}_{\text{A}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{T}_{\text{A}} \end{bmatrix}^{(j-1)} \begin{bmatrix} \Delta \mathbf{U}_{\text{N}}^{(j)} \\ \Delta \mathbf{U}_{\text{I}}^{(j-1)} \\ \Delta \mathbf{U}_{\text{A}}^{(j-1)} \\ \Delta \boldsymbol{\Lambda}_{\text{I}}^{(j)} \\ \Delta \boldsymbol{\Lambda}_{\text{A}}^{(j-1)} \end{bmatrix} = - \begin{bmatrix} \mathbf{F}_{\text{N}}^{\text{int}}(\mathbf{U}) \\ \mathbf{F}_{\text{I}}^{\text{int}}(\mathbf{U}) + \mathbf{D}_{\text{I}} \boldsymbol{\Lambda}_{\text{I}} \\ \mathbf{F}_{\text{A}}^{\text{int}}(\mathbf{U}) + \mathbf{D}_{\text{A}} \boldsymbol{\Lambda}_{\text{A}} \\ \boldsymbol{\Lambda}_{\text{I}} \\ -(\mathbf{N}_{\text{A}} \mathbf{U}_{\text{A}} - \mathbf{G}_{\text{A}}) \\ \mathbf{T}_{\text{A}} \boldsymbol{\Lambda}_{\text{A}} \end{bmatrix}^{(j-1)} \quad (3.22)$$

with

$$\mathbf{T}_{\text{A}} := \begin{bmatrix} \mathbf{T}_{\text{A}}^{(\varepsilon)} \\ \mathbf{T}_{\text{A}}^{(\eta)} \end{bmatrix} \text{ and with } \mathbf{K} := \text{Jacobian of } \mathbf{F}^{\text{int}} \text{ in terms of (2.11).}$$

Here,  $\mathbf{G}_{\text{A}}$  denotes the vector containing the entries  $G_p$  associated with the active vertex  $p \in \text{A}$ . One remark has to be made to the linearization of (3.18)<sub>2</sub>. From (3.17), one can see that

$$\begin{bmatrix} \mathbf{F}_{\nu}^{\text{con}}[\text{I}] \\ \mathbf{F}_{\nu}^{\text{con}}[\text{A}] \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Lambda}_{\text{I}} \\ -c(\mathbf{N}_{\text{A}} \mathbf{U}_{\text{A}} - \mathbf{G}_{\text{A}}) \end{bmatrix}$$

Finally, one gets

$$\frac{\partial \mathbf{F}_{\nu}^{\text{con}}[\text{I}]}{\partial (\mathbf{U}, \boldsymbol{\Lambda})} \Rightarrow \text{4th line of (3.22)}$$

and

$$\frac{\partial \mathbf{F}_{\nu}^{\text{con}}[\text{A}]}{\partial (\mathbf{U}, \boldsymbol{\Lambda})} \Rightarrow \text{5th line of (3.22)}$$

System (3.22) is not solved directly, since this would mean in the context of a FE package that the size of the system to be solved always changes if the size of  $\mathbb{S}$  is variable from time step to time step. This would be inefficient, but due to the diagonal structure of  $\mathbf{D}$ ,  $\boldsymbol{\Lambda}$  can be eliminated:

$$\boldsymbol{\Lambda}^{(j)} = \mathbf{D}^{-1} \left( [\mathbf{K}\mathbf{U}^{(j-1)}]_{\mathbb{S}} - [\mathbf{K}\mathbf{U}^{(j)}]_{\mathbb{S}} - \mathbf{F}_{\mathbb{S}}^{\text{int}}(\mathbf{U}^{(j-1)}) \right) \quad (3.23)$$

With this, a local static condensation of the Lagrange multipliers can be performed to get the reduced system for the displacements  $\Delta \mathbf{U}$ . The resulting system, written as an incremental Newton scheme, is then:

$$\begin{bmatrix} \mathbf{K}_{\text{NN}} & \mathbf{K}_{\text{NI}} & \mathbf{K}_{\text{NA}} \\ \mathbf{K}_{\text{IN}} & \mathbf{K}_{\text{II}} & \mathbf{K}_{\text{IA}} \\ \mathbf{0} & \mathbf{0} & -\mathbf{N}_{\text{A}} \\ \mathbf{T}_{\text{A}} \mathbf{K}_{\text{AN}} & \mathbf{T}_{\text{A}} \mathbf{K}_{\text{AI}} & \mathbf{T}_{\text{A}} \mathbf{K}_{\text{AA}} \end{bmatrix}^{(j-1)} \begin{bmatrix} \Delta \mathbf{U}_{\text{N}}^{(j)} \\ \Delta \mathbf{U}_{\text{I}}^{(j-1)} \\ \Delta \mathbf{U}_{\text{A}}^{(j-1)} \end{bmatrix} = - \begin{bmatrix} \mathbf{F}_{\text{N}}^{\text{int}}(\mathbf{U}) \\ \mathbf{F}_{\text{I}}^{\text{int}}(\mathbf{U}) \\ \mathbf{G}_{\text{A}} - \mathbf{N}_{\text{A}} \mathbf{U}_{\text{A}} \\ \mathbf{T}_{\text{A}} \mathbf{F}_{\text{A}}^{\text{int}}(\mathbf{U}) \end{bmatrix}^{(j-1)} \quad (3.24)$$

with

$$\Delta \mathbf{U}^{(j)} := \mathbf{U}^{(j)} - \mathbf{U}^{(j-1)}$$

Due to the consistent linearization, superlinear convergence can be expected. An important conclusion of this section is that the two nonlinearities of material behavior and

contact can be handled in only one Newton loop. This is possible because the search for the correct active set can also be interpreted as a Newton iteration, see also [2, 27, 53]. Now, Algorithm 1 can be applied for solving the nonlinear problem (3.18). Although, it is listed here for the sake of completeness, note that the integration of the SQP method into the overall solution process is different from the others, since it contains an outer SQP loop and an inner Newton loop for the mechanical equilibrium. To combine this efficiently with the active set loop will be part of the future work.

---

**Algorithm 1** Plasticity and contact.

---

- 1: **input:**  $[\mathbb{A}^{(0)}, j_{max}, \text{TOL}_{newt}, \mathbf{U}_{n+1}^{(0)}, [\boldsymbol{\varepsilon}_n^p, \alpha_n], \text{DIRECT}]$  with  $j_{max}$  a maximum number of Newton iterations, which can be set from outside, the starting iterate is  $\mathbf{U}_{n+1}^{(0)}$
- 2: init.  $j := 1$
- 3: assemble stiffness matrix and right hand side according to (2.11), (3.16), (3.21) and (3.24)
- 4: compute  $m_j$  according to Section 3.3.4
- 5: solve Equation (3.24) by

$$\left\{ \begin{array}{ll} \text{using Algorithm 2 with the input } [m_j, (3.24)_{lhs}, (3.24)_{rhs}] & \text{DIRECT} = 0 \\ \text{solving directly} & \text{DIRECT} = 1 \end{array} \right.$$

to compute a new Newton iterate  $\Delta \mathbf{U}_{n+1}^{(j)}$

- 6: update  $\mathbf{U}_{n+1}^{(j)} := \mathbf{U}_{n+1}^{(j-1)} + \Delta \mathbf{U}_{n+1}^{(j)}$
- 7: compute  $\boldsymbol{\Lambda}_{n+1}^{(j)}$  using Equation (3.23)
- 8: update  $\mathbb{I}^{(j)}$  and  $\mathbb{A}^{(j)}$  using (3.19) and (3.20)
- 9: on the GP level: map the stress state back to the yield surface and update the plastic strain:

$$\begin{aligned} & \text{call } [\boldsymbol{\varepsilon}_{n+1}^{p,(j)}, \mathbf{C}_{n+1}^{ep,(j)}, \alpha_{n+1}^{(j)}] \\ & \left\{ \begin{array}{ll} \text{TCP}(\boldsymbol{\varepsilon}(\mathbf{U}_{n+1}^{(j)}), \boldsymbol{\sigma}_{n+1}^{(j)}, \boldsymbol{\varepsilon}_n^p, \alpha_n, \Phi) & \text{see Section 2.5} \\ \text{RADIAL\_RETURN}(\boldsymbol{\varepsilon}(\mathbf{U}_{n+1}^{(j)}), \boldsymbol{\varepsilon}_n^p, \alpha_n, \Phi) & \text{see Section 2.4} \\ \text{SQP METHOD}(\dots) & \text{see Section 2.6} \\ \text{generalized projection method} & \text{see Section 8.5} \\ \vdots & \text{other options, see e.g. [26]} \end{array} \right. \\ & = \end{aligned}$$

- 10: **if** (a suitable Newton error criterion is met accurate to  $\text{TOL}_{newt}$  **and**  $\mathbb{A}^{(j)} = \mathbb{A}^{(j-1)}$ )  
**or**  $j = j_{max}$  **exit else set**  $j := j + 1$  **and goto** 3
  - 11: **output:**  $[\boldsymbol{\Lambda}_{n+1}, \mathbf{U}_{n+1}, [\boldsymbol{\varepsilon}_{n+1}^p, \alpha_{n+1}], \mathbb{A}]$
-

---

**Algorithm 2** AMG solver algorithm.

---

```

1: input:  $[m, \mathbf{A}, \mathbf{F}]$ 
2: for  $l = 1, \dots, m$  do
3:    $\Delta \mathbf{U}^{(j,l)} := \text{AMG\_CYCLE}(\Delta \mathbf{U}^{(j,l-1)}, \mathbf{A}, \mathbf{F})$ 
4: end for
5:  $\Delta \mathbf{U}^{(j)} := \Delta \mathbf{U}^{(j,l)}$ 
6: output:  $[\Delta \mathbf{U}^{(j)}]$ 

```

---

### 3.2.6 Inexact strategy

The next step to improve the performance even further is to use an inexact Newton strategy. The idea to reduce the total number of AMG cycles (see Algorithm 2) per time step is to perform only a few AMG steps ( $m_j$  small) in the first Newton iterations. In particular, it is not efficient to iterate the AMG until convergence, since the first iterates will be very far from the solution anyway. This method is called AMG-inexact strategy here and is used in Algorithm 1, line 4 and 5. In Section 3.3.4, some ad hoc strategies for choosing  $m_j$  are presented. By  $\mathbf{U}^{(j,l)}$ , the solution after the Newton iteration  $j$  and the AMG iteration  $l$  is denoted. In the inexact case, the constant  $c > 0$  in (3.17) has the meaning of a weighting between the non-penetration condition (3.3) and the contact stress condition (3.4) in the inactive-active decision. In the exact case,  $c$  does not play a role since it cannot happen that both of the summands  $\Lambda$  and  $U_\nu - G$  are not equal to zero. The ad hoc choice of  $\mathbb{A}$  and  $\mathbb{I}$  in Section 3.2.1 corresponds to the exact case, where the choice of  $c$  does not play a role.

## 3.3 Numerical results for contact

In this section, some numerical results are given to show the performance of the proposed algorithms. For that reason, the active set strategy is compared to a standard contact formulation, namely the penalty formulation, which is widely used throughout engineering practice. Only the penalty formulation is considered here, being aware that there are alternative solution strategies already mentioned in Section 3.1. Moreover, a comparison is shown between the algebraic multigrid solver and other classical iterative schemes. Finally, further examples are given by using the inexact strategy suggested above and an elastoplastic constitutive law.

### 3.3.1 Block indentation test

In the following, the example depicted in Figure 3.3 is considered, where a block of dimension  $100 \times 100 \times 50$  is indented by a hemispherical rigid punch with radius 30 by the depth  $d = 5$  within one time step. The block is discretized by tri-linear hexahedral elements and a constitutive law of nonlinear elasticity is used. For further information, the reader is referred to [10, 33].

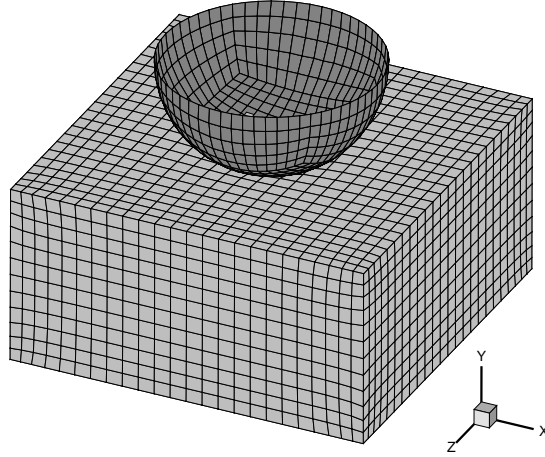


Figure 3.3: Setting of the contact problem with rigid tool.

Table 3.1: Comparison of results with decreasing penalty factor and active set strategy (ASeS).

method	displacement $d_{y,min}$	Mises equiv. stress $\bar{\sigma}_{max}$
$\rho = 10^5$	-2.79	5665
$\rho = 10^7$	-4.86	8370
$\rho = 10^9$	-4.89	8399
ASeS	-4.99	8398

### 3.3.2 Comparison of both contact approaches in connection with AMG

In this section, the active set strategy is compared with a standard penalty approach. The penalty method can be formulated for the potential contact nodes  $p \in \mathbb{S}$ :

$$[\mathbf{F}^{int}(\mathbf{U})]_p^\top \cdot \mathbf{N}_p + \rho(\mathbf{U}_p \cdot \mathbf{N}_p - G_p)_+ = 0, \quad p \in \mathbb{S}$$

Thus, the penetration in the normal direction is penalized by the artificial spring force with the spring rate  $\rho$ . This constant must be of significant size ( $10^8 - 10^{10}$  are common values in metal forming simulation) to avoid too large penetration of the bodies and ensure high accuracy. Figure 3.4 and Table 3.1 illustrate the mentioned behavior.

Unfortunately, the need for a large penalty value  $\rho$  is contradictory to the requirements of iterative solvers. The penalty factor deteriorates the condition numbers of the emerging equation systems, and thus leads to bad convergence rates of the iterative solvers. This problem can be significantly improved by using the primal-dual active set strategy as illustrated below. Figure 3.5 shows the convergence histories for the solution of the

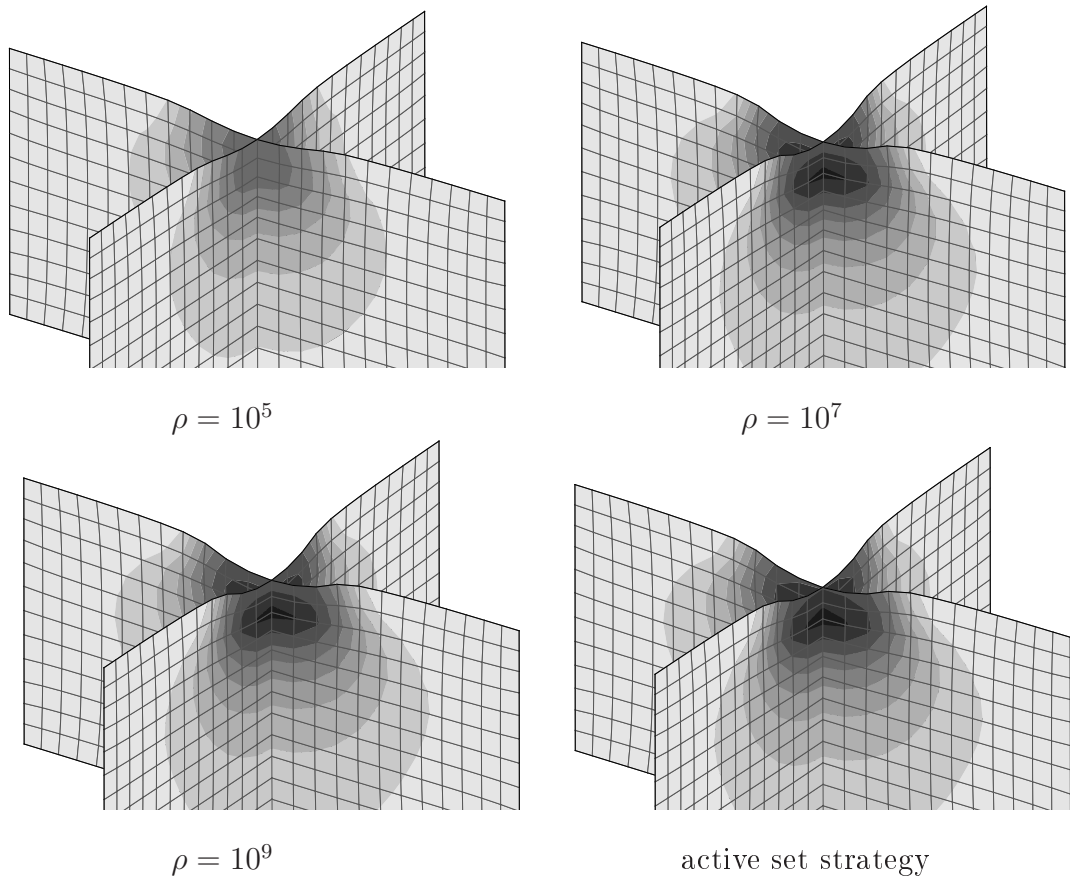


Figure 3.4: Deformed shape, loss in accuracy with decreasing penalty factor and active set strategy, von Mises equivalent stress  $\bar{\sigma}$ .

arising equation system in a representative Newton step, on the one hand using a penalty formulation ( $\rho = 10^9$ ) and on the other hand the active set strategy (ASeS). In both cases the algebraic multigrid solver is applied.

### 3.3.3 Active set versus adaptive penalty strategies

In Figure 3.6(A), the average computation time per Newton step for the active set strategy (a) is compared with the following different penalty strategies p1-p5 with the penalty parameter  $\rho = 10^{i_j}$  in the  $j$ th Newton step, see Table 3.2. Here for example  $i = 6, 8, 9$  means: The penalty parameter  $\rho$  is  $10^6$  in the first,  $10^8$  in the second and  $10^9$  in all the Newton steps during the rest of the time step. One sees that the computation time for the active set is significantly smaller in comparison to e.g. the penalty strategy p5, where a high penalty value is chosen right from the start. The explanation is that the contact penalty deteriorates the condition of the stiffness matrix as mentioned before and therefore slows down the convergence of the iterative solver [16, 48]. But one could think



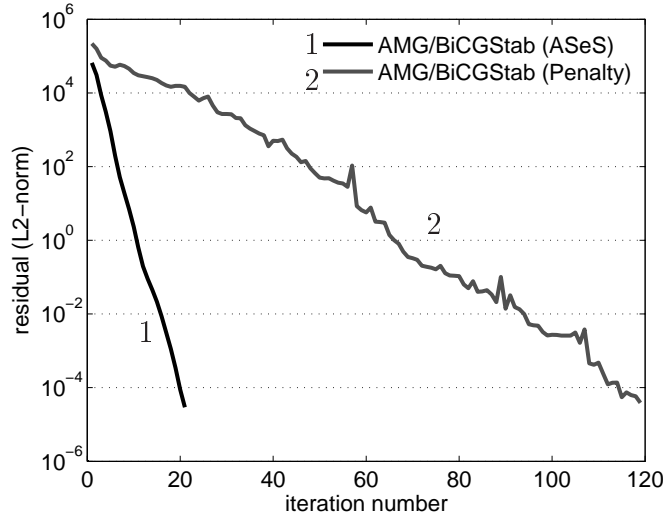


Figure 3.5: Residual reduction per iteration. ASeS respectively penalty approach and AMG-solver.

Table 3.2: Different penalty strategies.

p1:	$i = 2,4,6,8,9$
p2:	$i = 4,9$
p3:	$i = 6,8,9$
p4:	$i = 4,7,9$
p5:	$i = 9$
a:	-

that there are clever strategies which work adaptively in the sense that they start with a small penalty value in the first Newton steps and choose the expensive high penalty value only in the last Newton steps. This is done in the strategies p1-p4. Figure 3.6(B) shows that this does not work out because low penalty values in the beginning compels to take more Newton steps, since there is a large penetration in the beginning, and therefore the convergence of the Newton method is undermined. Only the strategy p3 performs quite well but this strategy as well as the others need much more total computation time over the whole time step than the active set strategy, see Figure 3.6(C). It can be concluded that the convergence speed of iterative solvers is significantly lower when using a penalty formulation, no matter if a high value is used right from the start or if an adaptive strategy is used. Again, it should be remarked, that the other contact formulations, mentioned in Section 3.1 do not suffer from this problem, but have other drawbacks. For example the Augmented Lagrange technique can work with low penalty values, but has to pay the price of higher computation times [106].

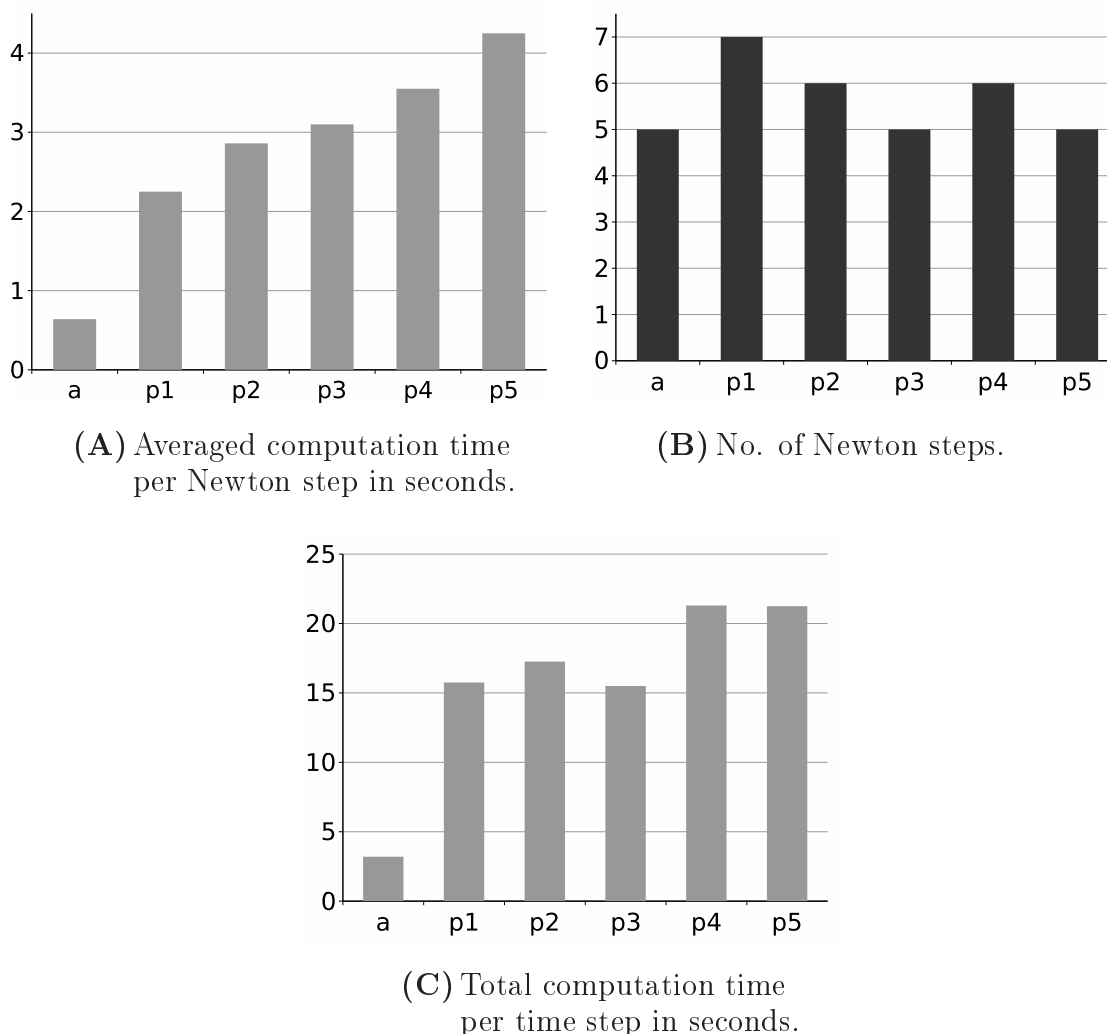


Figure 3.6: Comparison between active set strategy (a) and different adaptive penalty strategies (p1-p5).

### 3.3.4 Acceleration by AMG-inexact strategy

As mentioned above, further acceleration can be achieved by an inexact strategy according to Algorithm 2. Figure 3.7 and Table 3.3 describe the parameter settings and show the numerical results for the simulations performed.

In the strategy i4, for example (see Figure 3.7, Table 3.3), the number of AMG cycles  $m$  is fixed to 1, 1, 2 in the first three Newton-Raphson steps (NR steps) and during the rest of the time step the solver is let to iterate until convergence. In contrast the exact strategy i1, in which always AMG convergence is awaited, altogether has much more solver cycles and therefore a larger computation time. Other ‘semi-inexact’ examples with a variable number of solver cycles are given in the strategies i2 and i3. It is even

Table 3.3: Inexact strategy, convergence history of respective loops.

run	NR steps with prescribed values ( $m_j$ )	NR step of active set convergence	NR step with NR loop convergence	total AMG cycles	total time [s]
i1	-	3	6	67	10.2
i2	2 (2,5)	3	6	35	8.1
i3	4 (2,2,3,4)	3	6	18	5.7
i4	3 (1,1,2)	3	6	15	5.1
i5	all (1,1,...)	3	15	15	8.9

possible to perform the totally inexact strategy given in i5, where only one AMG cycle per Newton step is performed, which means that the Newton loop and the solver loop coincide now. Although the total number of AMG cycles is 15 as in case i4, this totally inexact strategy needs more computation time than the semi-inexact one. This can be explained by the fact that in strategy i5 the setup of the AMG solver must be initialized more often, since more Newton steps are performed, and thus there are more different equation systems. This again could be overcome by preserving the coarse grid levels from one Newton step to the next, but it would have to be observed that the active set might change and has to be represented correctly on the coarse grids. By applying the whole setup phase in each Newton step, this problem is avoided. Using an adaptive strategy would be appealing, but then again the black-box character of the described method is lost to some extent.

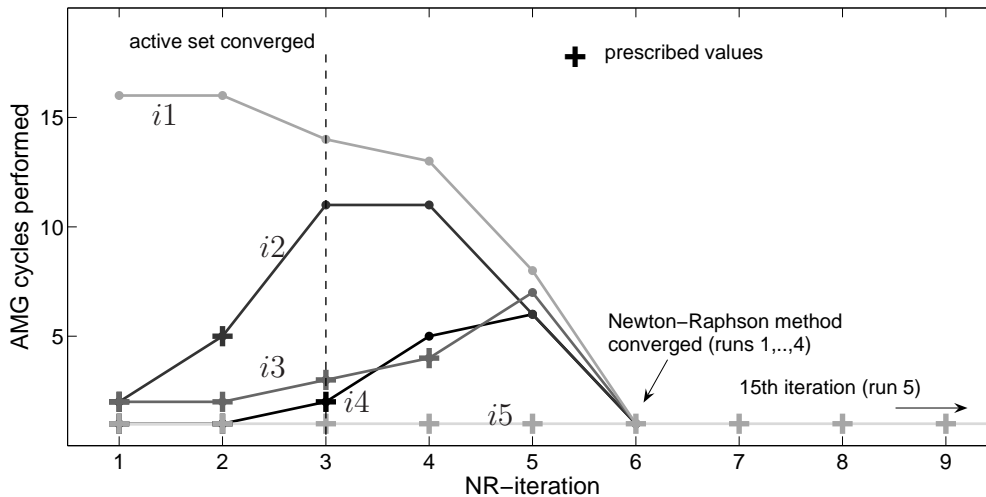


Figure 3.7: Active set strategy and AMG with loop unrolling.

## 3.4 Summary

It can be concluded that the proposed contact algorithm allows to apply fast iterative solvers on nonlinear problems in a very efficient way. It was shown, that the discussed algebraic multigrid methods significantly gain in efficiency in comparison to standard penalty approaches when used in combination with a primal-dual active set strategy. It will be part of the future work to compare this method with the other contact formulations mentioned in Section 3.1. Furthermore it was shown that computation time can be further decreased by merging the otherwise nested loops of each method into a single solver loop by the described inexact strategies. The theory explained exhibits the character of the method and a way to implement the method in a practical manner into a FE package. The study is based on a very simple approach for the algebraic multigrid in order to preserve the black-box applicability of the method to a maximum extent.

The remark has to be made, that there are many reasons for a bad condition number of the system matrix, not only the contact penalty. For example, also buckling tends to deteriorate the condition number but it is not addressed in this work. The reason is that here, it is assumed, that a metal forming process which is near to buckling or any kind of bifurcation behavior <sup>1</sup> belongs already to the ill-controlled processes which are not to be simulated here. Nevertheless, it stays part of the future work to

1. validate this assumption carefully and, if it proves to be true,
2. to incorporate the detection of undesired bifurcation behavior into the algorithms of this work.

---

<sup>1</sup>to such an extent that the condition number gets significantly deteriorated (at the bifurcation point, the Jacobian is singular)

# 4 Dynamic contact

The efficient modeling of dynamic, thin-walled, large deformation 3D contact problems is still a challenge in nonlinear implicit structural analysis.

## 4.1 Introduction

A great number of contact algorithms developed in the past enforce the contact constraints at specific collocation points. Using a node-to-segment approach, the main idea is that a specific node on the slave side must not penetrate the opposing master side segment. Although this approach is quite popular and available in numerous commercial finite element codes, the robustness of these methods is still a limitation in certain applications. So-called single pass algorithms do not satisfy the contact patch test [89], where a flat contact surface should be able to transmit a spatially constant contact pressure from one body to the other. Using low order finite elements, the nonsmooth representation of the real geometry will have a significant influence on the performance of a node-to-segment approach. It will cause jumps in the contact forces once a slave node slides off the contacting master segment. Various smoothing algorithms have been proposed to overcome these deficiencies, e.g. [88, 91, 100]. Proposed two-pass algorithms indeed pass the contact patch test for 2D and some 3D mesh configurations, but they suffer in turn from locking.

Due to these drawbacks of the classical contact algorithms, the research on segment-to-segment contact strategies became quite active in recent years. Most of these new approaches use the so-called mortar method, initially introduced as a domain decomposition method by Bernardi and co-workers [14]. The reader is referred to [9, 11, 12] and the references therein for an overview of the mortar method in the context of contact problems. Yang et al. [115] describe a contact method for two dimensional large deformation frictional sliding. Puso and Laursen [92, 93] develop a mortar segment-to-segment contact method for large deformation solid mechanics for three dimensional applications. McDevitt and Laursen [82] introduce an intermediate mortar surface to project the contact conditions between two contacting surfaces. A 2D mortar formulation was also used by Fischer and Wriggers [38].

In this formulation, a primal-dual active set strategy [2, 27, 53] is used which allows to enforce the contact constraints without any additional parameter. The success of this strategy is based on the introduction of dual Lagrange multipliers [107] to discretize the contact pressure. Following the approach described in Hübner and Wohlmuth [57], one ends up with an algebraic structure of the problem which allows to locally eliminate the

introduced Lagrange multipliers. The contact pressure can be regarded as an external force which is dual to the contact displacement and therefore be easily recovered from the displacements in a variationally consistent way.

In this chapter, the contact formulation is presented in the context of implicit structural dynamics, using two different time discretization schemes, the Generalized- $\alpha$  Method [28] and the Generalized Energy-Momentum Method [74] to compare the development of energies. To end up with an energy conserving framework, an idea by Laursen and Love [77] is picked up who introduce a discrete contact velocity to update the velocity field in a post-processing step. The necessary generalization of this approach according to the utilized time integration schemes as well as the incorporation into the primal-dual active set strategy is presented. Finally, an algorithm for a surface oriented shell element, based on a three-dimensional 7-parameter formulation, including the thickness stretch of the shell [22], is included. Various examples show the good performance of the primal-dual active set strategy applied to the implicit dynamic analysis of thin-walled structures.

## 4.2 Problem description

In this section, the one-body, rigid obstacle problem for large deformations is briefly reviewed and the notation used throughout this chapter is introduced. The boundary value problem for nonlinear elastodynamics as well as its weak form is shortly presented.

### 4.2.1 One-body contact with rigid obstacle

A three dimensional, large deformation one-body contact problem is shown in Figure 4.1. The rigid obstacle is represented by  $\Omega_{obs}$  and its potential contact surface is denoted with  $\Gamma_{obs}$ . The deformable body is represented by  $\Omega$  and  $\bar{\Omega}$  in the reference and current configuration, respectively. The notation  $(\bar{\cdot})$  for all quantities corresponding to the current configuration will be used. The surface  $\partial\Omega = \Gamma$  is divided into  $\Gamma_u$  and  $\Gamma_\sigma$ , where displacements and tractions are prescribed and into  $\Gamma^{con}$ , where the contact constraints will be defined. For the sake of clarity, only the contact surfaces  $\Gamma^{con}$  and  $\bar{\Gamma}^{con}$  are shown in Figure 4.1. Considering a point  $\mathbf{x}$  on the contact surface of the deformable body, a corresponding point  $\hat{\mathbf{y}}(\mathbf{x})$  is defined (in analogy to  $\mathbf{x}_{tool}$  like in Chapter 3) on  $\Gamma_{obs}$  as a result of the closest point projection given by

$$\hat{\mathbf{y}}(\mathbf{x}) := \arg \min_{\mathbf{y} \in \Gamma_{obs}} \|\mathbf{x} - \mathbf{y}\| \quad (4.1)$$

The motion of the deformable body from the reference to the current configuration at time  $t$  is given by

$$\bar{\mathbf{x}} = \mathbf{x} + \mathbf{u} \quad (4.2)$$

The (not necessarily contact) tractions w.r.t. the reference configuration  $\mathbf{t}$  and the tractions w.r.t. the current configuration  $\bar{\mathbf{t}}$  at a specific point  $\mathbf{x} \in \Gamma^{con}$  are

$$\begin{aligned} \mathbf{t} &= \mathbf{P}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) \\ \bar{\mathbf{t}} &= \boldsymbol{\sigma}(\bar{\mathbf{x}}, t) \cdot \bar{\mathbf{n}}(\bar{\mathbf{x}}) \end{aligned}$$

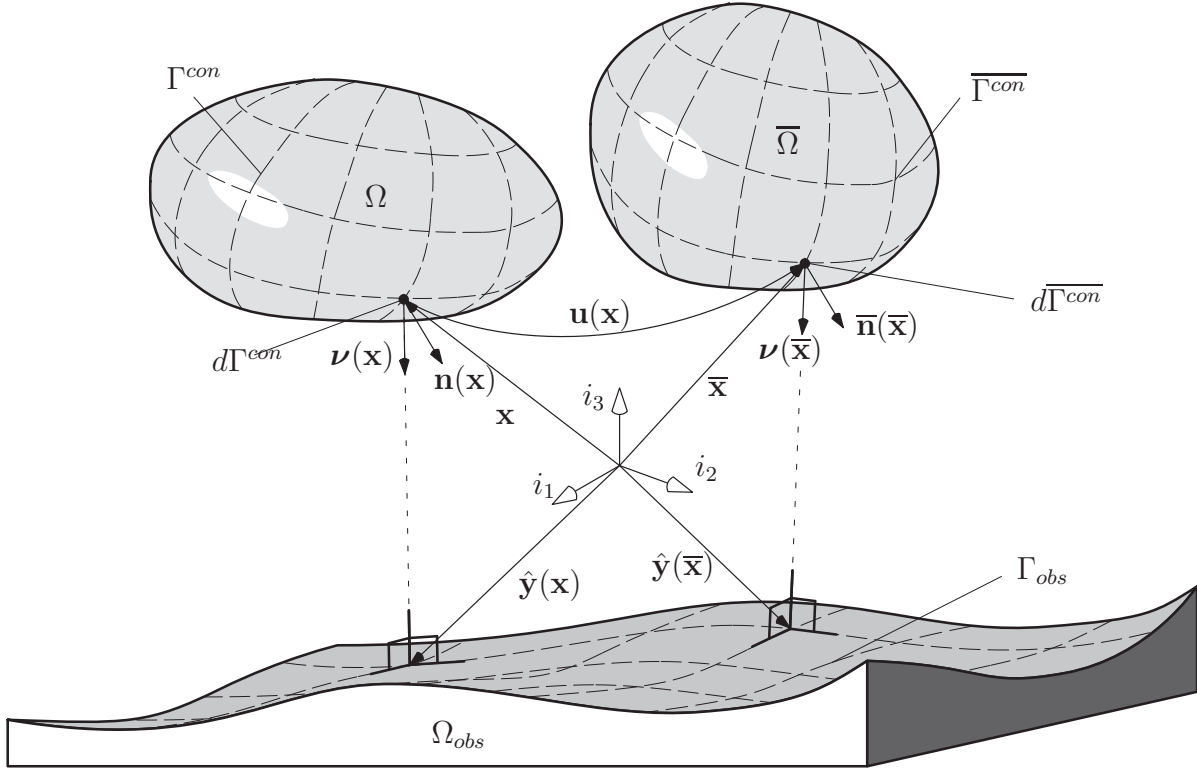


Figure 4.1: Notation for one-body large deformation contact problem.

where  $\mathbf{n}$  and  $\bar{\mathbf{n}}$  are the outward unit normal on  $\Omega$  and  $\bar{\Omega}$ , respectively. Due to the definition of the first Piola Kirchhoff (PK) stress tensor  $\mathbf{P}$ , the Cauchy stress tensor  $\boldsymbol{\sigma}$  and Nanson's formula (1878), one finds that

$$\mathbf{t} = \frac{d\bar{\Gamma}^{con}}{d\Gamma^{con}} \cdot \bar{\mathbf{t}} \quad (4.3)$$

where  $d\Gamma^{con}$ ,  $d\bar{\Gamma}^{con}$  are the differential areas in the reference and current configuration, respectively, see also Section 8.7.1. From this, one can see that the difference between the current contact tractions and the reference contact tractions is only the change in the area resulting from the deformation  $\frac{d\bar{\Gamma}^{con}}{d\Gamma^{con}}$  which is a positive number. Since each of the following considerations involving the contact tractions will only depend on their sign, one can work with the contact tractions measured in the reference configuration.

### 4.2.2 Boundary value problem of nonlinear elastodynamics

In the context of large deformation, nonlinear elastodynamics, the bivariate form  $f^{int}$  in Equation (3.7) is specified to the weak form of the following boundary value problem

$$\begin{aligned} \rho \ddot{\mathbf{u}} &= \operatorname{div} (\mathbf{F} \cdot \mathbf{T}) && \text{in } \Omega \\ \mathbf{u} &= \hat{\mathbf{u}} && \text{on } \Gamma_u \\ \mathbf{t} &= \hat{\mathbf{t}} && \text{on } \Gamma_\sigma \end{aligned} \quad (4.4)$$

where  $\mathbf{F}$  and  $\mathbf{T}$  are the material deformation gradient and the second PK stress tensor, respectively. Appropriate boundary conditions are given by the prescribed displacements  $\hat{\mathbf{u}}$  and tractions  $\hat{\mathbf{t}}$ , acting on the correlated boundaries  $\Gamma_u$  and  $\Gamma_\sigma$ . The second Piola-Kirchhoff stress tensor is given via an appropriate constitutive relation, e.g. for St. Venant-Kirchhoff material:

$$\mathbf{T} = \mathbf{C}^{el} : \mathbf{E} \quad (4.5)$$

with  $\mathbf{E}$  representing the Green-Lagrange strain tensor defined as

$$\mathbf{E} := \frac{1}{2} (\mathbf{F}^\top \mathbf{F} - \mathbf{1}) \quad (4.6)$$

### 4.3 Spatial discretization of contact virtual work

In Section 3.2.4, the notion of dual Lagrange multipliers was introduced. In case of quadrilateral elements on  $\Gamma^{con}$  with a constant functional determinant, the ansatz-functions for the Lagrange multipliers in the isoparametric space (functions, defined there, are marked with  $\hat{\cdot}$ ) can be specified at once. In Figure 4.2a and 4.2c, the standard bi-linear shape functions  $\hat{\phi}_1, \dots, \hat{\phi}_4 : [-1, 1]^2 \rightarrow \mathbb{R}$

$$\begin{aligned} \hat{\phi}_1 &= \frac{1}{4}(1 - \xi)(1 - \eta) && \text{(shown)} \\ \hat{\phi}_2 &= \frac{1}{4}(1 + \xi)(1 - \eta) \\ \hat{\phi}_3 &= \frac{1}{4}(1 + \xi)(1 + \eta) \\ \hat{\phi}_4 &= \frac{1}{4}(1 - \xi)(1 + \eta) \end{aligned}$$

and the corresponding dual basis functions  $\hat{\psi}_1, \dots, \hat{\psi}_4 : [-1, 1]^2 \rightarrow \mathbb{R}$  for parallelogram shaped elements are given in the isoparametric space. For arbitrarily shaped elements, the ansatz-functions for the discretized dual Lagrange multipliers have to be constructed [43]. Without detailing the theory, the necessary steps to derive the appropriate discretized Lagrange multipliers fulfilling the biorthogonality condition with the standard bi-linear shape functions are presented.



### 4.3.1 Discrete dual Lagrange multipliers for arbitrary shaped elements

Define with  $T$  and  $\hat{T} = [-1, 1]^2$  a 4-noded element on the contact boundary  $\Gamma^{con}$ , given in the physical and the isoparametric space, respectively. Furthermore by

$$\mathbf{F}_T : \hat{T} \rightarrow T \subset \mathbb{R}^3$$

the element mapping such that  $\hat{\phi}_i = \phi_i \circ \mathbf{F}_T$  is denoted. The aim is now to find functions  $\psi_i$  such that the biorthogonality condition (3.10) holds element-wise on  $T \subset \Gamma^{con}$ :

$$\int_T \psi_i \phi_j dT = \delta_{ij} \int_T \phi_j dT \quad (4.7)$$

Therefore, one constructs the ansatz-functions for the Lagrange multipliers in the following way:

$$\hat{\psi}_i := \sum_k a_{ik} \hat{\phi}_k, \quad \text{with} \quad \mathbf{A}_T := a_{ij} := \mathbf{D}_T \mathbf{M}_T^{-1} \in \mathbb{R}^{4 \times 4} \quad (4.8)$$

and therein

$$\mathbf{D}_T := \delta_{ij} \int_T \phi_j dT, \quad \mathbf{M}_T := \int_T \phi_i \phi_j dT$$

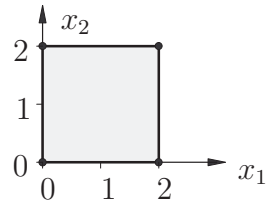
the desired diagonal matrix and the element mass matrix without density and sets  $\psi_i := \hat{\psi}_i \circ \mathbf{F}_T^{-1}$ . Then one verifies:

$$\begin{aligned} \int_T \psi_i \phi_j dT &= \int_{\hat{T}} J \hat{\psi}_i \hat{\phi}_j d\hat{T} \\ &= \int_{\hat{T}} J \sum_k a_{ik} \hat{\phi}_k \hat{\phi}_j d\hat{T} \\ &= \sum_k a_{ik} \int_{\hat{T}} J \hat{\phi}_k \hat{\phi}_j d\hat{T} \\ &= \sum_k a_{ik} [\mathbf{M}_T]_{kj} \\ &= [\mathbf{A}_T \mathbf{M}_T]_{ij} = [\mathbf{D}_T]_{ij} \end{aligned}$$

Herein  $J := J_T := \det(\mathbf{F}_T)$  is the Jacobian of an element  $T$ . To further illustrate this procedure, one discrete ansatz-function for the dual Lagrange multipliers for two different element shapes is exemplarily calculated. The first one has a regular shape, resulting in a constant Jacobian determinant  $J = const$  whereas the second one is of arbitrary shape with  $J \neq const$ .

**Undistorted Element:**

$$\begin{bmatrix} 0 & 2 & 2 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$



The diagonal matrix, mass matrix and the resulting transformation matrix are:

$$\mathbf{D}_T = \text{diag}([1 \ 1 \ 1 \ 1]), \quad \mathbf{M}_T = \frac{1}{9} \begin{bmatrix} 4 & 2 & 1 & 2 \\ 2 & 4 & 2 & 1 \\ 1 & 2 & 4 & 2 \\ 2 & 1 & 2 & 4 \end{bmatrix}, \quad \mathbf{A}_T = \begin{bmatrix} 4 & -2 & 1 & -2 \\ -2 & 4 & -2 & 1 \\ 1 & -2 & 4 & -2 \\ -2 & 1 & -2 & 4 \end{bmatrix}$$

So one can compute for example the discretized Lagrange multiplier  $\hat{\psi}_1$  in the isoparametric space  $[\xi, \eta] \in [-1, 1]^2$  from Equation (4.8):

$$\begin{aligned} \hat{\psi}_1 &= 4\hat{\phi}_1 - 2\hat{\phi}_2 + 1\hat{\phi}_3 - 2\hat{\phi}_4 \\ &= \frac{1}{4}(1 - 3\xi)(1 - 3\eta) \end{aligned}$$

$$\hat{\psi}_1(-1, -1) = 4, \quad \hat{\psi}_1(1, -1) = -2, \quad \hat{\psi}_1(1, 1) = 1, \quad \hat{\psi}_1(-1, 1) = -2$$

which is always the case for constant  $J$ .

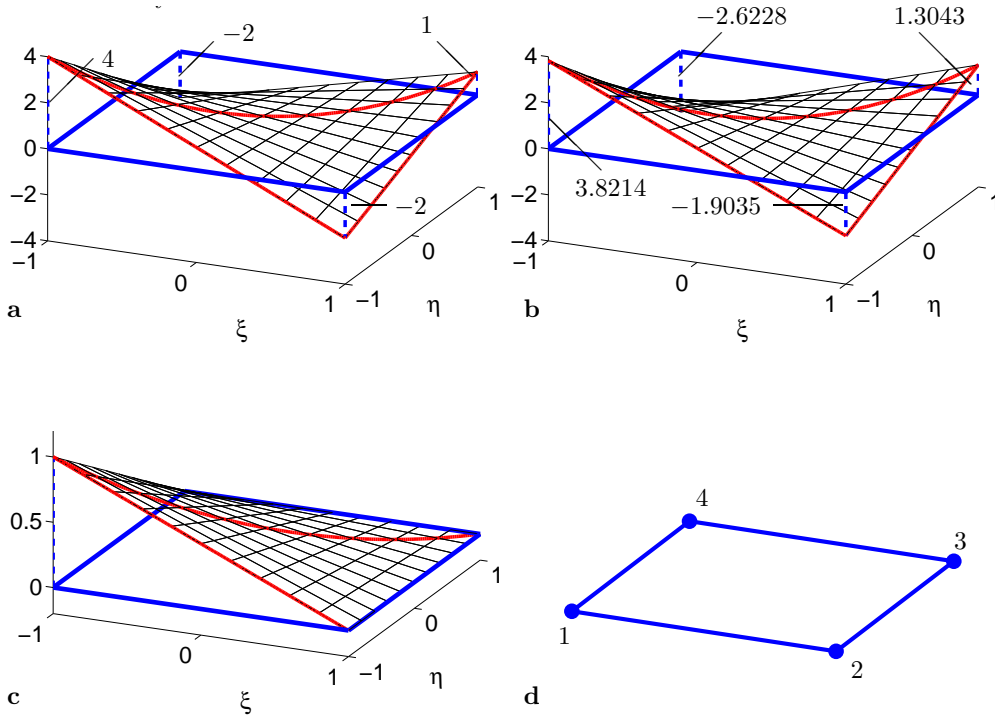
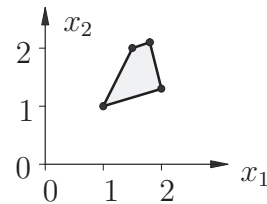


Figure 4.2: Plot of  $\hat{\psi}_1$  for the undistorted (a), for the distorted (b) element, of the shape function  $\hat{\phi}_1$  (c) and of the ordering of the shape functions  $\hat{\phi}_1, \dots, \hat{\phi}_4$  (d).

**Distorted Element:**

$$\begin{bmatrix} 1 & 2 & 1.8 & 1.5 \\ 1 & 1.3 & 2.1 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$



The diagonal matrix and mass matrix are:

$$\mathbf{D}_T = \text{diag} \left( \begin{bmatrix} 0.1633 \\ 0.1642 \\ 0.1142 \\ 0.1133 \end{bmatrix} \right), \quad \mathbf{M}_T = \begin{bmatrix} 0.0781 & 0.0392 & 0.0154 & 0.0307 \\ 0.0392 & 0.0786 & 0.0310 & 0.0154 \\ 0.0154 & 0.0310 & 0.0453 & 0.0225 \\ 0.0307 & 0.0154 & 0.0225 & 0.0447 \end{bmatrix}$$

The resulting transformation matrix is:

$$\mathbf{A}_T = \begin{bmatrix} 3.8214 & -1.9035 & 1.3043 & -2.6228 \\ -1.9132 & 3.8118 & -2.6075 & 1.3109 \\ 0.9116 & -1.8133 & 4.6024 & -2.3161 \\ -1.8199 & 0.9050 & -2.2992 & 4.6280 \end{bmatrix}$$

Again, one can compute the discretized Lagrange multiplier  $\hat{\psi}_1$ :

$$\hat{\psi}_1 = 3.8214\hat{\phi}_1 - 1.9035\hat{\phi}_2 + 1.3043\hat{\phi}_3 - 2.6228\hat{\phi}_4$$

$$\hat{\psi}_1(-1, -1) = 3.8214 \quad \hat{\psi}_1(1, -1) = -1.9035 \quad \hat{\psi}_1(1, 1) = 1.3043 \quad \hat{\psi}_1(-1, 1) = -2.6228$$

In Figure 4.2a and 4.2b the ansatz-functions  $\hat{\psi}_1$  is shown for the undistorted and the distorted element, respectively. The general structure of the interpolation functions remains the same, but the basis values at the nodes of the elements vary.

### 4.3.2 Semidiscrete initial value problem

Starting from the weak form of the nonlinear dynamic contact problem (4.4), the semidiscrete equation of motion is derived introducing the spatial discretization via finite elements.

$$\mathbf{M}_\rho \ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{F}^{int}(\mathbf{U}) + \mathbf{F}^c = \mathbf{F}^{ext} \quad (4.9)$$

Herein,  $\mathbf{M}_\rho$  represents the mass matrix,  $\mathbf{C}$  the viscous damping matrix,  $\mathbf{F}^{int}(\mathbf{U})$  the deformation dependent internal forces,  $\mathbf{F}^c$  the contact forces and  $\mathbf{F}^{ext}$  the external forces.  $\dot{\mathbf{U}}$  are the nodal velocities and  $\ddot{\mathbf{U}}$  the nodal accelerations. Consequently, the inertia forces and damping forces are given by  $\mathbf{M}_\rho \ddot{\mathbf{U}}$  and  $\mathbf{C}\dot{\mathbf{U}}$ , respectively. The damping matrix is assumed to be a linear combination of the mass matrix and the elastic stiffness matrix  $\mathbf{A}$  which corresponds to the so-called proportional or Rayleigh damping.

After appropriate ordering of the variables, the contact forces can be expressed in exactly the same manner as in Equation (3.9)

$$\mathbf{F}^c = \mathbf{B} \mathbf{\Lambda} \quad (4.10)$$

It is worth noting that the determination of the entries of  $\mathbf{B}$  can be performed in the reference configuration. This will have the advantage that the weighting factors have to be calculated only once, thus no linearization is necessary for the calculation of  $\mathbf{B}$ .

## 4.4 Time discretization

To integrate the semidiscrete problem (4.9), the time interval of interest  $[0, T]$  is subdivided into typical time intervals  $[t_n, t_{n+1}]$  with the corresponding time step  $\Delta t = t_{n+1} - t_n$ . Given the state variables  $\mathbf{U}_n$ ,  $\dot{\mathbf{U}}_n$  and  $\ddot{\mathbf{U}}_n$  at time  $t_n$ , Equation (4.9) is solved in an implicit way to obtain the solution of the state variables  $\mathbf{U}_{n+1}$ ,  $\dot{\mathbf{U}}_{n+1}$  and  $\ddot{\mathbf{U}}_{n+1}$  at the end of each time step. To reduce the set of unknown variables to the displacements  $\mathbf{U}_{n+1}$ , the classical Newmark approximations are used.

$$\begin{aligned}\mathbf{U}_{n+1}(\ddot{\mathbf{U}}_{n+1}) &= \mathbf{U}_n + \Delta t \dot{\mathbf{U}}_n + \frac{1-2\beta}{2} \Delta t^2 \ddot{\mathbf{U}}_n + \beta \Delta t^2 \ddot{\mathbf{U}}_{n+1} \\ \dot{\mathbf{U}}_{n+1}(\ddot{\mathbf{U}}_{n+1}) &= \dot{\mathbf{U}}_n + (1-\gamma)\Delta t \ddot{\mathbf{U}}_n + \gamma \Delta t \ddot{\mathbf{U}}_{n+1}\end{aligned}\quad (4.11)$$

Equivalently the velocities and accelerations are rewritten in terms of the displacement at time  $t_{n+1}$ .

$$\begin{aligned}\dot{\mathbf{U}}_{n+1}(\mathbf{U}_{n+1}) &= \frac{\gamma}{\beta \Delta t} (\mathbf{U}_{n+1} - \mathbf{U}_n) - \left(\frac{\gamma}{\beta} - 1\right) \dot{\mathbf{U}}_n - \left(\frac{\gamma}{2\beta} - 1\right) \Delta t \ddot{\mathbf{U}}_n \\ \ddot{\mathbf{U}}_{n+1}(\mathbf{U}_{n+1}) &= \frac{1}{\beta \Delta t^2} (\mathbf{U}_{n+1} - \mathbf{U}_n) - \frac{1}{\beta \Delta t} \dot{\mathbf{U}}_n - \left(\frac{1}{2\beta} - 1\right) \ddot{\mathbf{U}}_n\end{aligned}\quad (4.12)$$

In the following, two slightly different time integration schemes are described, both based on the given Newmark scheme. The Generalized- $\alpha$  Method (Gen- $\alpha$ ), originally proposed for linear dynamical systems by Chung and Hulbert [28] and the Generalized Energy-Momentum Method (GEMM) developed by Kuhl and Crisfield [73] for trusses and later adapted for nonlinear shell dynamics by Kuhl and Ramm [74]. Both methods apply the semidiscrete equation of motion (4.9) at a generalized mid-point configuration

$$\mathbf{M}_\rho \ddot{\mathbf{U}}_{n+1-\alpha_m} + \mathbf{C} \dot{\mathbf{U}}_{n+1-\alpha_f} + \mathbf{F}_{n+1-\alpha_f}^{int} + \mathbf{F}_{n+1-\alpha_f}^c = \mathbf{F}_{n+1-\alpha_f}^{ext} \quad (4.13)$$

described via two shift parameters  $\alpha_m$  and  $\alpha_f$ . Herein, the subscripts denote the time discrete combinations of the accelerations, velocities, displacements, internal, contact and external forces.

$$\begin{aligned}\ddot{\mathbf{U}}_{n+1-\alpha_m} &= (1-\alpha_m)\ddot{\mathbf{U}}_{n+1} + \alpha_m \ddot{\mathbf{U}}_n \\ \dot{\mathbf{U}}_{n+1-\alpha_f} &= (1-\alpha_f)\dot{\mathbf{U}}_{n+1} + \alpha_f \dot{\mathbf{U}}_n \\ \mathbf{U}_{n+1-\alpha_f} &= (1-\alpha_f)\mathbf{U}_{n+1} + \alpha_f \mathbf{U}_n \\ \mathbf{F}_{n+1-\alpha_f}^{int} &= \begin{cases} (1-\alpha_f)\mathbf{F}_{n+1}^{int} + \alpha_f \mathbf{F}_n^{int} & \text{for Gen-}\alpha \\ \mathbf{F}^{int} ((1-\alpha_f)\mathbf{U}_{n+1} + \alpha_f \mathbf{U}_n) & \text{for GEMM} \end{cases} \\ \mathbf{F}_{n+1-\alpha_f}^{ext} &= (1-\alpha_f)\mathbf{F}_{n+1}^{ext} + \alpha_f \mathbf{F}_n^{ext}\end{aligned}\quad (4.14)$$

The crucial difference between the Generalized- $\alpha$  Method and the Generalized Energy-Momentum Method is the calculation of the internal forces at the generalized mid-point configuration. In the Generalized- $\alpha$  Method, the internal forces are approximated using the general trapezoidal rule, whereas in the General Energy-Momentum Method, the internal forces are calculated at the mid-point configuration. This is essential for this

method and will have significant influence on the derivation of the deformation dependent algorithmic tangential stiffness matrix, which will be discussed separately when describing the surface oriented shell element in Section 4.5. According to Equation (4.10), the contribution of the contact forces in Equation (4.13) can be expressed as

$$\mathbf{F}_{n+1-\alpha_f}^c = \mathbf{B} \boldsymbol{\Lambda}_{n+1-\alpha_f} \quad (4.15)$$

Combining the approximations of the velocities and accelerations (4.12) and (4.14) with the equation of motion (4.13) together with (4.15) yields the effective structural equation which is nonlinear in the unknown displacements  $\mathbf{U}_{n+1}$

$$\begin{aligned} \mathbf{Z}(\mathbf{U}_{n+1}) &:= \mathbf{F}_{n+1-\alpha_f}^{int}(\mathbf{U}_{n+1}) + \frac{1-\alpha_m}{\beta\Delta t^2} \mathbf{M}_\rho \mathbf{U}_{n+1} + \frac{(1-\alpha_f)\gamma}{\beta\Delta t} \mathbf{C} \mathbf{U}_{n+1} - \\ &\mathbf{H}(\mathbf{U}_n, \dot{\mathbf{U}}_n, \ddot{\mathbf{U}}_n) + \mathbf{B} \boldsymbol{\Lambda}_{n+1-\alpha_f} - \mathbf{F}_{n+1-\alpha_f}^{ext} = \mathbf{0} \end{aligned} \quad (4.16)$$

where  $\mathbf{H}(\mathbf{U}_n, \dot{\mathbf{U}}_n, \ddot{\mathbf{U}}_n)$ , the history term, is only dependent on the given state variables at time  $t_n$ .

$$\begin{aligned} \mathbf{H}(\mathbf{U}_n, \dot{\mathbf{U}}_n, \ddot{\mathbf{U}}_n) &:= \mathbf{M}_\rho \left[ \frac{1-\alpha_m}{\beta\Delta t^2} \mathbf{U}_n + \frac{1-\alpha_m}{\beta\Delta t} \dot{\mathbf{U}}_n + \frac{1-\alpha_m-2\beta}{2\beta} \ddot{\mathbf{U}}_n \right] + \\ &\mathbf{C} \left[ \frac{(1-\alpha_f)\gamma}{\beta\Delta t} \mathbf{U}_n + \frac{(1-\alpha_f)\gamma-\beta}{\beta} \dot{\mathbf{U}}_n + \frac{(\gamma-2\beta)(1-\alpha_f)}{2\beta} \Delta t \ddot{\mathbf{U}}_n \right] \end{aligned} \quad (4.17)$$

#### 4.4.1 Linearization and iterative solution strategy

The nonlinearity of the effective structural equation needs an iterative solution procedure. Therefore Equation (4.16) is linearized in direction of

$$\Delta \mathbf{U}^{(j+1)} = \mathbf{U}_{n+1}^{(j+1)} - \mathbf{U}_{n+1}^{(j)} \quad (4.18)$$

and the Newton-Raphson iteration technique is applied. This leads to the effective iterative structural equation

$$\mathbf{J}^{(j)} \Delta \mathbf{U}^{(j+1)} = -\mathbf{Z}(\mathbf{U}_{n+1}^{(j)}) \quad (4.19)$$

where  $\mathbf{J}^{(j)}$  is the effective algorithmic tangential stiffness matrix

$$\mathbf{J}^{(j)} := \frac{\partial \mathbf{F}^{int}(\mathbf{U}_{n+1-\alpha_f}(\mathbf{U}_{n+1}))}{\partial \mathbf{U}_{n+1}} + \frac{1-\alpha_m}{\beta\Delta t^2} \mathbf{M}_\rho + \frac{(1-\alpha_f)\gamma}{\beta\Delta t} \mathbf{C} \quad (4.20)$$

and  $\mathbf{Z}(\mathbf{U}_{n+1}^{(j)})$  are the out-of-balance forces at iteration step  $j$ . Now, all these matrices are sorted according to the definition of the subsets  $\mathbb{N}$  and  $\mathbb{S}$ , and the contact term  $\mathbf{B} \boldsymbol{\Lambda}_{n+1-\alpha_f}$  is put onto the left hand side. With the ordering (3.11), one arrives at the following incremental representation of the effective structural Equation (4.16)

$$\begin{bmatrix} \mathbf{J}_{\mathbb{N}\mathbb{N}} & \mathbf{J}_{\mathbb{N}\mathbb{S}} & \mathbf{0} \\ \mathbf{J}_{\mathbb{S}\mathbb{N}} & \mathbf{J}_{\mathbb{S}\mathbb{S}} & \mathbf{D} \end{bmatrix}^{(j)} \begin{bmatrix} \Delta \mathbf{U}_{\mathbb{N}} \\ \Delta \mathbf{U}_{\mathbb{S}} \\ \Delta \boldsymbol{\Lambda}_{n+1-\alpha_f} \end{bmatrix}^{(j+1)} = - \begin{bmatrix} \mathbf{Z}_{\mathbb{N}} \\ \mathbf{Z}_{\mathbb{S}} + \mathbf{D} \boldsymbol{\Lambda}_{n+1-\alpha_f} \end{bmatrix}^{(j)} \quad (4.21)$$

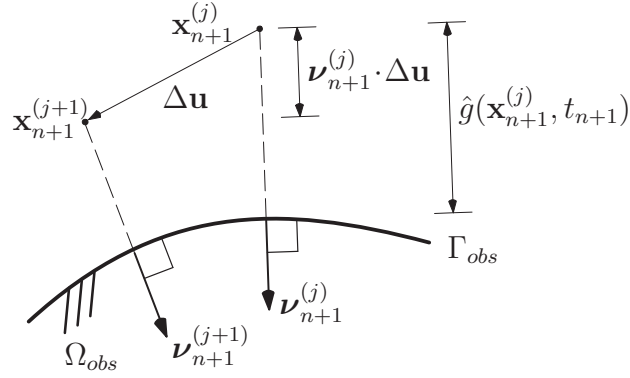


Figure 4.3: Incremental impenetrability condition.

#### 4.4.2 Global system with contact manipulation

Up to now, the gap has not been specified yet. The gap is defined as in Section 3.2.2 by

$$\hat{g}(\mathbf{x}_{n+1}^{(j)}, t_{n+1}) = -\boldsymbol{\nu}(\mathbf{x}_{n+1}^{(j)}) \cdot (\mathbf{x}_{n+1}^{(j)} - \hat{\mathbf{y}}(\mathbf{x}_{n+1}^{(j)})) \quad (4.22)$$

where the contact reference  $\mathbf{x}_{\text{ref}}$  is the last known Newton step  $\mathbf{x}_{n+1}^{(j)}$ , see Figure 4.3. Now, with (4.21) and the fact that the second part of the nonlinear problem (3.18) is the same in this setting except for the gap, one arrives at a global system which has intentionally been written the same way as System (3.22). This has been done to make clear that, although the structural and out-of-balance forces are completely different from the quasi-static case, the implementation of the contact is structurally the same.

$$\begin{bmatrix} \mathbf{J}_{\text{NN}} & \mathbf{J}_{\text{NI}} & \mathbf{J}_{\text{NA}} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_{\text{IN}} & \mathbf{J}_{\text{II}} & \mathbf{J}_{\text{IA}} & \mathbf{D}_{\text{I}} & \mathbf{0} \\ \mathbf{J}_{\text{AN}} & \mathbf{J}_{\text{AI}} & \mathbf{J}_{\text{AA}} & \mathbf{0} & \mathbf{D}_{\text{A}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}_{\text{I}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{N}_{\text{A}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{T}_{\text{A}} \end{bmatrix}^{(j)} \begin{bmatrix} \Delta \mathbf{U}_{\text{N}}^{(j+1)} \\ \Delta \mathbf{U}_{\text{I}_j}^{(j+1)} \\ \Delta \mathbf{U}_{\text{A}_j}^{(j+1)} \\ \Delta \boldsymbol{\Lambda}_{\text{I}_j}^{(j+1)} \\ \Delta \boldsymbol{\Lambda}_{\text{A}_j}^{(j+1)} \end{bmatrix} = - \begin{bmatrix} \mathbf{Z}_{\text{N}} \\ \mathbf{Z}_{\text{I}} + \mathbf{D}_{\text{I}} \boldsymbol{\Lambda}_{\text{I}} \\ \mathbf{Z}_{\text{A}} + \mathbf{D}_{\text{A}} \boldsymbol{\Lambda}_{\text{A}} \\ \boldsymbol{\Lambda}_{\text{I}} \\ -(\mathbf{N}_{\text{A}} \mathbf{U}_{\text{A}} - \mathbf{G}_{\text{A}}) \\ \mathbf{T}_{\text{A}} \boldsymbol{\Lambda}_{\text{A}} \end{bmatrix}^{(j)} \quad (4.23)$$

One does not directly solve system (4.23) because this would lead to a change in the size of the system matrices during the calculation which implies several drawbacks for the performance of a solver. Due to the dual Lagrange multiplier space, the unknown values of  $\boldsymbol{\Lambda}$  can again be locally eliminated using Equation (4.21).

$$\boldsymbol{\Lambda} = \mathbf{D}^{-1} (-\mathbf{Z}_{\text{S}} - \mathbf{J}_{\text{SN}} \Delta \mathbf{U}_{\text{N}} - \mathbf{J}_{\text{SS}} \Delta \mathbf{U}_{\text{S}}) \quad (4.24)$$

A static condensation yields the reduced system

$$\begin{bmatrix} \mathbf{J}_{\text{NN}} & \mathbf{J}_{\text{NI}} & \mathbf{J}_{\text{NA}} \\ \mathbf{J}_{\text{IN}} & \mathbf{J}_{\text{II}} & \mathbf{J}_{\text{IA}} \\ \mathbf{0} & \mathbf{0} & -\mathbf{N}_{\text{A}} \\ \mathbf{T}_{\text{A}}\mathbf{J}_{\text{AN}} & \mathbf{T}_{\text{A}}\mathbf{J}_{\text{AI}} & \mathbf{T}_{\text{A}}\mathbf{J}_{\text{AA}} \end{bmatrix}^{(j)} \begin{bmatrix} \Delta\mathbf{U}_{\text{N}}^{(j+1)} \\ \Delta\mathbf{U}_{\text{I}_j}^{(j+1)} \\ \Delta\mathbf{U}_{\text{A}_j}^{(j+1)} \end{bmatrix} = - \begin{bmatrix} \mathbf{Z}_{\text{N}} \\ \mathbf{Z}_{\text{I}} \\ \mathbf{G}_{\text{A}} - \mathbf{N}_{\text{A}}\mathbf{U}_{\text{A}} \\ \mathbf{T}_{\text{A}}\mathbf{Z}_{\text{A}} \end{bmatrix}^{(j)} \quad (4.25)$$

to be solved in each Newton step  $j$ . As mentioned before, the size of this system remains constant during the calculation and is determined only by the finite element discretization. In the present approach, the contact forces are calculated from the displacements in a variationally consistent way in a post processing step. The geometric contact constraint of an active node in  $\text{A}_j$  is directly introduced as a Dirichlet boundary condition in a weak integral sense. Therefore no user defined parameter like a penalty parameter is needed in this approach.

## 4.5 Spatial discretization by a finite shell element

The spatial discretization of the deformable body is realized by an eight noded, hexahedral surface oriented finite shell element of the Reissner/Mindlin type. Its formulation is based on the shell element presented by Büchter et al. ([22], [23]). A detailed description and several enhancements to this shell formulation are presented by Bischoff [15].

For the application to contact problems, the original description of the shell geometry was modified. In the original formulation, the kinematics of the nonlinear shell element is described via a displacement field of the shell mid-surface plus an extensible shell director field. Having the finite element nodes situated in the shell mid-surface, the formulation of contact problems becomes quite complicated as all the contact integrals have to be calculated with respect to the real surface of the shell body. Thus, contact forces acting on the actual shell surface have to be transmitted to the degrees of freedom living in the mid-surface. Although this approach has been successfully applied for a classical node-to-segment contact formulation by Gee [44], the geometry description is reformulated, leading to a surface-oriented shell element. With the finite element nodes now lying on the shell surface, one can apply the contact approach without any further modification as for any standard hexahedral solid element.

## 4.6 Energy conservation for frictionless contact

It is known [52, 59, 75] that the application of the presented time integration algorithms, without any special treatment of the contact constraints will not result in an energy conservative method. Dealing with frictionless dynamic contact problems, it is well known that the fulfillment of the so-called *persistence condition*

$$(\boldsymbol{\lambda}(\mathbf{x}, t) \cdot \boldsymbol{\nu}(\mathbf{x}, t))\dot{g}(\mathbf{x}, t) = 0 \quad (4.26)$$

will lead to an energy conserving scheme. Here, the gap  $g$  is the current gap

$$g(\mathbf{x}, t) = -\boldsymbol{\nu}(\bar{\mathbf{x}}) \cdot (\bar{\mathbf{x}} - \hat{\mathbf{y}}(\bar{\mathbf{x}})), \quad t \in [0, T]$$

instead of the gap in Equation (4.22). Considering the time continuous case, the *persistence condition* is automatically fulfilled if the Karush-Kuhn-Tucker conditions are satisfied. Various strategies to fulfill this additional constraint (e.g. [5] and [76]) have the drawback of violating the geometric admissibility instead. This work follows the idea of Laursen and Love [77] by introducing a discrete contact velocity  $\mathbf{V}^c$  into the velocity update:

$$\dot{\mathbf{U}}_{n+1}^{upd} = \dot{\mathbf{U}}_{n+1} + \mathbf{V}^c$$

This additional term is motivated by the fact that the analytical solution of a simple one-dimensional contact problem is characterized by discrete velocity jumps occurring within one discrete time step. Without an explicit compliance of an algorithmic persistence condition, this strategy allows the treatment of the geometric contact constraints in an unmodified manner. Thus the impenetrability condition is still satisfied in a weak integral sense. The framework described in [77] considers the classical midpoint rule. This approach is generalized according to the Generalized- $\alpha$  and the Generalized Energy-Momentum Method, leading to the following form of the velocity update, see Equation (4.12)

$$\dot{\mathbf{U}}_{n+1}^{upd} = \frac{\gamma}{\beta\Delta t}(\mathbf{U}_{n+1} - \mathbf{U}_n) - \left(\frac{\gamma}{\beta} - 1\right)\dot{\mathbf{U}}_n - \left(\frac{\gamma}{2\beta} - 1\right)\Delta t \ddot{\mathbf{U}}_n + \mathbf{V}^c \quad (4.27)$$

To derive the energy conservation conditions, a special variation of the discretized problem (4.13) is performed, excluding the damping term and the external forces since energy conservation only makes sense in this setting. Due to the correction term, the dynamic equilibrium

$$\mathbf{M}_\rho \left( \ddot{\mathbf{U}}_{n+1-\alpha_m} - \frac{1-\alpha_m}{\gamma\Delta t} \mathbf{V}^c \right) + \mathbf{F}_{n+1-\alpha_f}^{int} + \mathbf{F}_{n+1-\alpha_f}^c = \mathbf{0} \quad (4.28)$$

is solved and this equation is multiplied with (see Equation (4.27))

$$\Delta \mathbf{U} = \Delta \mathbf{U}^{upd} - \frac{\beta\Delta t}{\gamma} \mathbf{V}^c \quad (4.29)$$

with

$$\Delta \mathbf{U}^{upd} := \frac{\beta\Delta t}{\gamma} \left[ \dot{\mathbf{U}}_{n+1}^{upd} + \left(\frac{\gamma}{\beta} - 1\right)\dot{\mathbf{U}}_n + \left(\frac{\gamma}{2\beta} - 1\right)\Delta t \ddot{\mathbf{U}}_n \right] \quad (4.30)$$

In the absence of contact, the change in total energy can be expressed as

$$\ddot{\mathbf{U}}_{n+1-\alpha_m}^\top \mathbf{M}_\rho \Delta \mathbf{U}^{upd} + (\mathbf{F}_{n+1-\alpha_f}^{int})^\top \Delta \mathbf{U} = \Delta E^{tot} \quad (4.31)$$

In the following, only the terms arising from the contact forces and the discrete velocity update are considered to arrive at a condition which satisfies energy conservation for the contact part. Therefore Equation (4.28) is multiplied with  $\Delta \mathbf{U}$  (Equation (4.29))

$$0 = (4.28)_{\text{lhs}} \cdot \Delta \mathbf{U} = \Delta E^{tot} + \Delta E^{num}$$



with the pure numerical gain or loss of energy

$$\Delta E^{num} := \left( -\frac{1-\alpha_m}{\gamma\Delta t} \Delta \mathbf{U}^{upd} + \frac{\beta(1-\alpha_m)}{\gamma^2} \mathbf{V}^c - \frac{\beta\Delta t}{\gamma} \ddot{\mathbf{U}}_{n+1-\alpha_m} \right)^\top \mathbf{M}_\rho \mathbf{V}^c + (\mathbf{F}_{n+1-\alpha_f}^c)^\top \Delta \mathbf{U} \quad (4.32)$$

So the condition

$$\Delta E^{num} = 0 \quad (4.33)$$

guarantees energy conservation for the contact part. Therefore, the total energy conservation depends on the used time integration scheme.

After inserting all necessary interpolation definitions into Equation (4.33) and after some algebraic modifications, one can express the contact energy conservation condition in terms of known variables

$$\left( R_1 \Delta \mathbf{U} + R_2 \dot{\mathbf{U}}_n + R_3 \ddot{\mathbf{U}}_n + R_4 \mathbf{V}^c \right)^\top \mathbf{M}_\rho \mathbf{V}^c + (\mathbf{F}_{n+1-\alpha_f}^c)^\top \Delta \mathbf{U} = 0 \quad (4.34)$$

Herein the constants  $R_i$  are defined as

$$R_1 := \frac{2(1-\alpha_m)}{\gamma\Delta t}, \quad R_2 := -\frac{(1-\alpha_m)}{\gamma}, \quad R_3 := -\frac{\Delta t}{2\gamma}(2\beta + \alpha_m - 1), \quad R_4 := \frac{\beta(1-\alpha_m)}{\gamma^2} \quad (4.35)$$

Define now the vector of nodal normal contact forces  $\mathbf{\Lambda}_\nu \in \mathbb{R}^{nc}$  and the matrix

$$\hat{\mathbf{N}}_n := \begin{bmatrix} \mathbf{N}_{n,1} & & \\ & \ddots & \\ & & \mathbf{N}_{n,nc} \end{bmatrix} \in \mathbb{R}^{3nc \times nc} \quad (4.36)$$

which is the same matrix as in Equation (3.21) but without the weighting  $w_{pp}$ . The number of potential contact nodes is abbreviated by  $nc$ . To solve Equation (4.34) with respect to the unknown discrete velocity jump  $\mathbf{V}^c$ , Equation (4.10) is used

$$\mathbf{F}_{n+1-\alpha_f}^c = \mathbf{B} \hat{\mathbf{N}}_{n+1-\alpha_f} \mathbf{\Lambda}_\nu \quad (4.37)$$

to express the contact forces and similarly

$$\mathbf{M}_\rho \mathbf{V}^c = \mathbf{B} \hat{\mathbf{N}}_{n+\alpha_\nu} \mathbf{P}_\nu \quad (4.38)$$

to represent the magnitude of the impulses  $\mathbf{P}_\nu$  across the contact surface, presumed to be acting in the direction of the surface normal as well. Using Equation (4.38), one can express the discrete nodal velocity jump as

$$\mathbf{V}^c = \mathbf{M}_\rho^{-1} \mathbf{B} \hat{\mathbf{N}}_{n+1} \mathbf{P}_\nu \quad (4.39)$$

Now the expressions (4.37), (4.38) and (4.39) are inserted into Equation (4.34) to end up with a quadratic expression for the unknown nodal impulse values  $\mathbf{P}_\nu$  on  $\Gamma^{con}$ .

$$\mathbf{P}_\nu^\top \mathbf{A}^{LL} \mathbf{P}_\nu + \mathbf{B}^{LL\top} \mathbf{P}_\nu + c^{LL} = 0 \quad (4.40)$$

with

$$\begin{aligned}
 \mathbf{A}^{LL} &:= R_4 \hat{\mathbf{N}}_{n+1}^\top \mathbf{B}^\top \mathbf{M}_\rho^{-1} \mathbf{B} \hat{\mathbf{N}}_{n+1} \\
 \mathbf{B}^{LL} &:= (R_1 \Delta \mathbf{U} + R_2 \dot{\mathbf{U}}_n + R_3 \ddot{\mathbf{U}}_n)^\top \mathbf{B} \hat{\mathbf{N}}_{n+1} \\
 c^{LL} &:= \Delta \mathbf{U}^\top \mathbf{B} \hat{\mathbf{N}}_{n+1-\alpha_f} \boldsymbol{\Lambda}_\nu
 \end{aligned} \tag{4.41}$$

Of course, the quadratic Equation (4.40) does not have a unique solution, as the set of solutions  $\mathcal{P}$  is an ellipsoid in  $\mathbb{R}^{nc}$ . But the solution is unique if (4.40) is fulfilled individually for each nodal point  $i \in \Gamma^{con}$ , see (4.42), and Equation (4.44), given below, is satisfied.

$$\sum_{j=1}^{nc} (A_{ij}^{LL} P_{\nu,i} p_{\nu,j}) + B_i^{LL} P_{\nu,i} + c^{LL} = 0 \tag{4.42}$$

A reorganization of this equation will yield a simple quadratic form

$$A_{ii}^{LL} (P_{\nu,i})^2 + \left[ \sum_{j=1, j \neq i}^{nc} A_{ij}^{LL} P_{\nu,j} + B_i^{LL} \right] P_{\nu,i} + c^{LL} = A_{ii}^{LL} (P_{\nu,i})^2 + B_i^{LL} P_{\nu,i} + c^{LL} = 0 \tag{4.43}$$

which can easily be solved, giving two real-valued solutions. Laursen and Love [77] showed that enforcing the condition

$$\text{sign}(P_{\nu,i}) = \text{sign}(B^i) \tag{4.44}$$

guarantees that  $P_{\nu,i} = 0$  if  $\Lambda_{\nu,i} = 0$  which is physically meaningful. Once the values of the nodal impulses have been determined, the discrete velocity jump can be calculated using Equation (4.39), and finally, the update of the velocity field (4.27) is performed in a post processing step to ensure total energy conservation. For further details of this approach, the reader is referred to the work by Laursen and Love [77]. It must be noted that in case of impact problems the Lagrange multiplier must appropriately be smoothed in time since the accelerations and therefore the contact forces are a Dirac impulse which leads to a highly oscillating  $\boldsymbol{\lambda}$ , see Hauret and Le Tallec [52] where a penalty formulation is used which shows the problem for high penalty values. So the contact decision has to be modified accordingly.

## 4.7 Examples

Four examples are chosen to examine the performance of the primal-dual active set strategy applied to the described time integration schemes. All implementations were done in CCARAT by Dr. Ing. Stefan Hartmann and Dipl. Ing. Thomas Cichosz. CCARAT is a scientific FE package at the Institute for Structural Mechanics in Stuttgart. Various sets of integration parameters are analyzed in view of the evolution of the total energy and the effect of the velocity update. No physical damping has been assumed. The time integration parameters  $\beta$ ,  $\gamma$ ,  $\alpha_m$  and  $\alpha_f$  can be expressed as functions of the spectral radius or of the high frequency dissipation coefficient  $\rho_\infty \in [0, 1]$  as

$$\alpha_m := \frac{2\rho_\infty - 1}{\rho_\infty + 1}, \quad \alpha_f := \frac{\rho_\infty}{\rho_\infty + 1}, \quad \beta := \frac{1}{4}(1 - \alpha_m + \alpha_f)^2, \quad \gamma := \frac{1}{2} - \alpha_m + \alpha_f \tag{4.45}$$

The choice  $\rho_\infty = 1$  corresponds to the case of no algorithmic dissipation, while a smaller  $\rho_\infty < 1$  renders dissipation which increases with a decreasing value of  $\rho_\infty$ .

### 4.7.1 Weak non-penetration

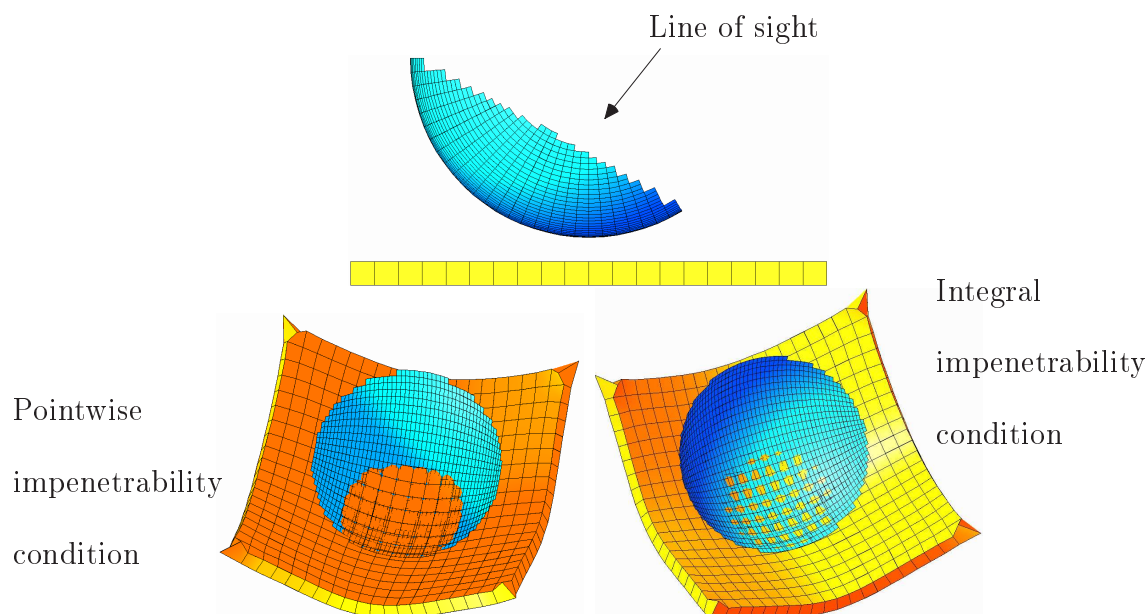


Figure 4.4: Weak non-penetration, taken from [30].

In this example, a rigid half sphere is pressed against a elastic plate. The weak non-penetration condition (3.14) is imposed as well as a point-wise one ( $\mathbf{D} \rightarrow \mathbf{1}$ ). The results are shown in Figure 4.4: The weak non-penetration condition leads to much less penetration than the pointwise one.

### 4.7.2 Toss rule

To analyze the two different time integration schemes, the different sets of time integration parameters and the effect of the velocity update, the example of the three-dimensional movement of a toss rule is chosen. The problem set up, including geometry, material and load time curve is given in Figure 4.5. For all calculations a time step  $\Delta t = 50\mu s$  is used. The spatial discretization is done with 30 eight-noded surface oriented hexahedral shell elements. In Figure 4.6, the motion of the toss rule is shown where the calculation is done by the Generalized Energy-Momentum Method with  $\rho_\infty = 1.0$ . The distribution of the total energy, the kinetic energy and the strain energy, calculated with the Generalized Energy-Momentum Method and the Generalized- $\alpha$  Method is shown for the case *with* and *without* the velocity update approach in the Figures 4.7 and 4.8, respectively. It can be seen that the full conservation of total energy can only be achieved with the

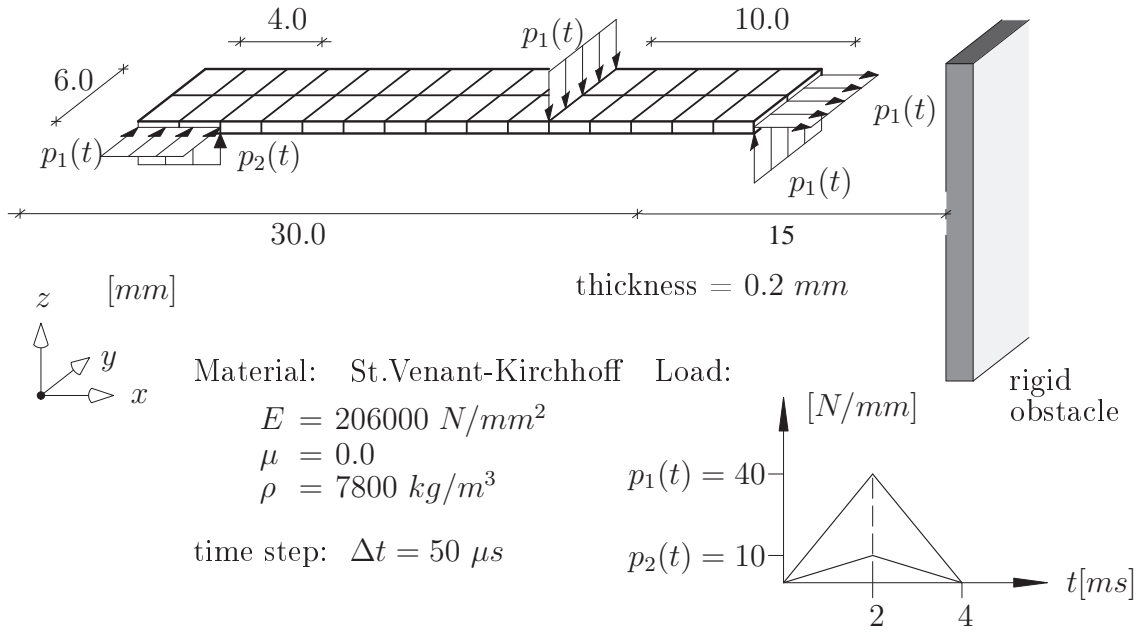


Figure 4.5: Toss rule - geometry, material and load.

Generalized Energy-Momentum Method, using  $\rho_\infty = 1.0$ , together with the application of the velocity update algorithm. It is interesting to see that the usage of the velocity update together with the Generalized- $\alpha$  Method does not lead to an improvement concerning the development of the total energy. In case of the General Energy-Momentum Method, the decrease of total energy for a spectral radius  $\rho_\infty < 1$  is smoother when the velocity update algorithm is used. It should be pointed out that the contact algorithm itself turns out to be very robust, regardless of which time integration scheme is used.

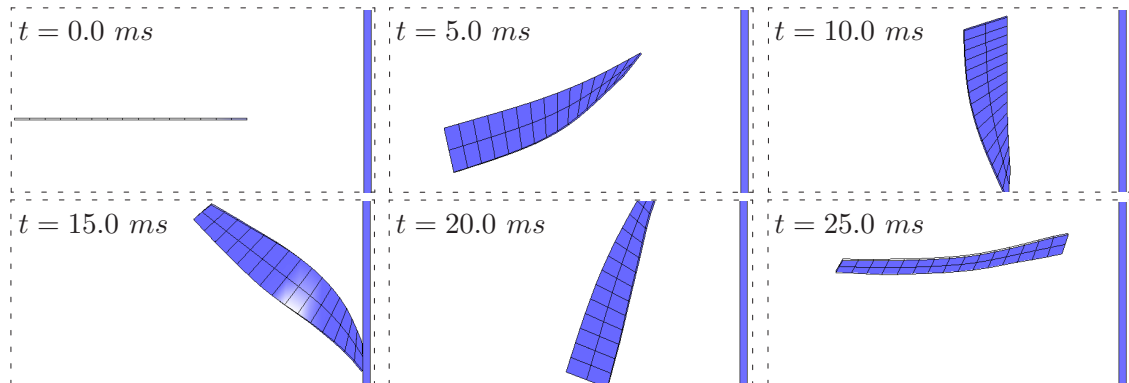


Figure 4.6: Motion of the toss rule - Generalized Energy-Momentum Method  $\rho_\infty = 1.0$ .

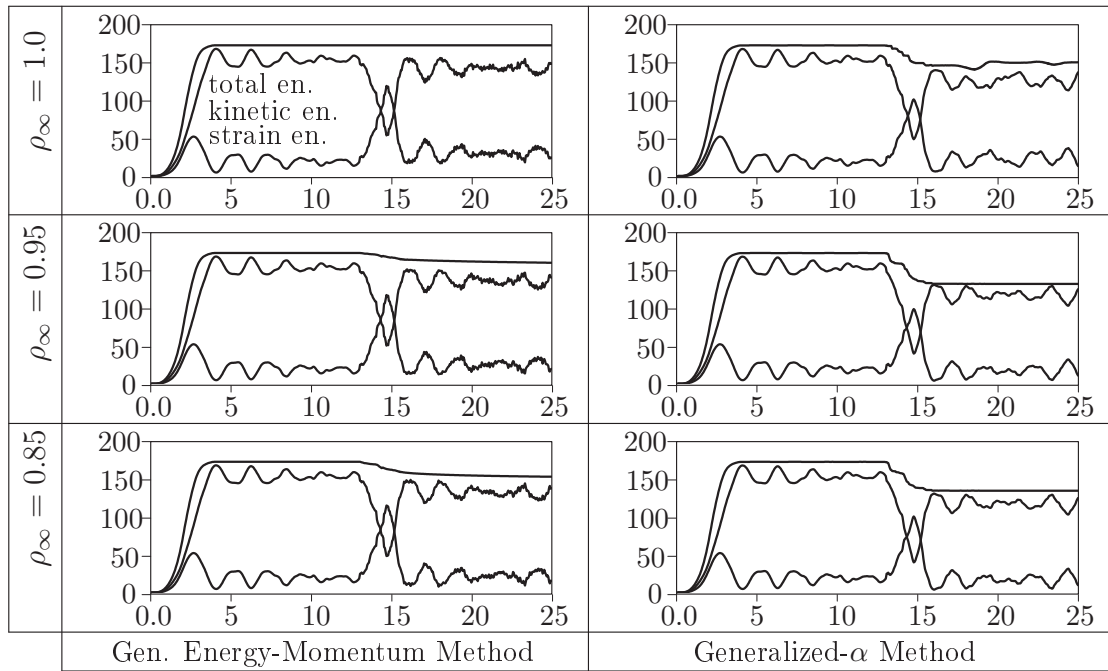


Figure 4.7: Toss rule - Energy E [J] vs time t [ $\mu$ s] for GEMM and Gen- $\alpha$  *with* velocity update.

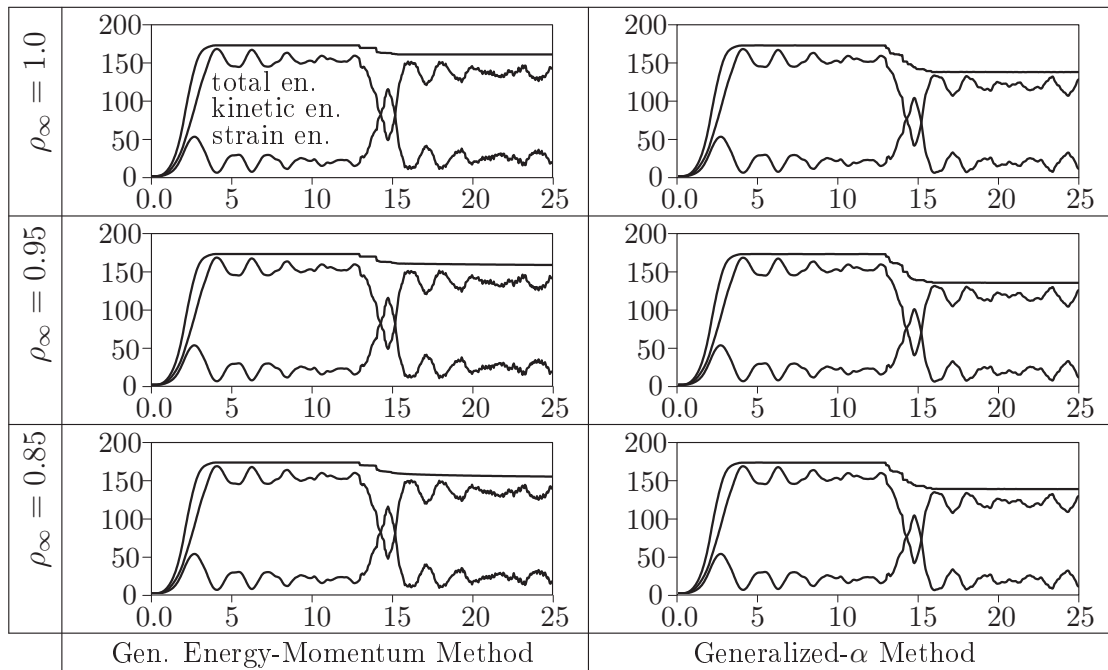


Figure 4.8: Toss rule - Energy E [J] vs time t [ $\mu$ s] for GEMM and Gen- $\alpha$  *without* velocity update.

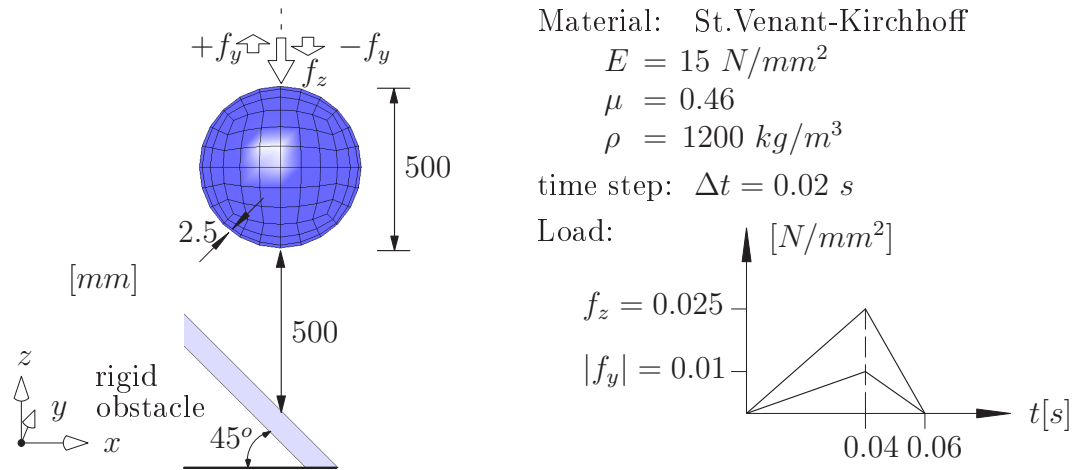


Figure 4.9: Ball - geometry, material and load.

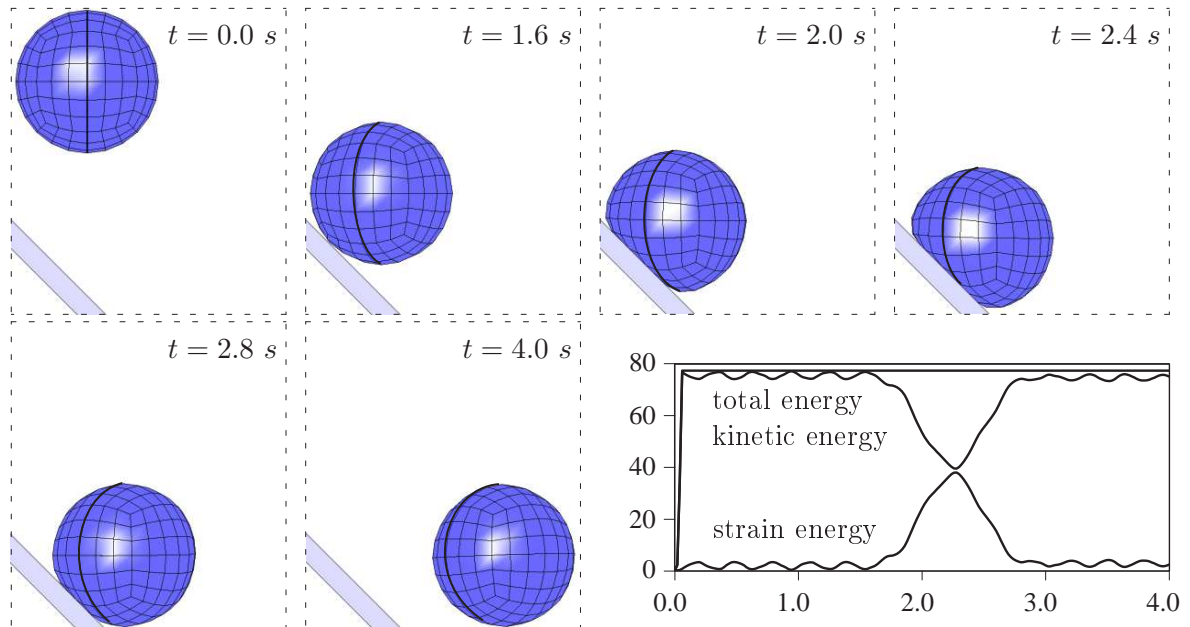


Figure 4.10: Ball - Motion and Energy  $E$  [J] vs time  $t$  [s] for GEMM  $\rho_\infty = 1.0$ .

### 4.7.3 Ball

The third example is a thin ball which is thrown onto an inclined rigid obstacle. Geometry, material and position of the applied loads as well as their load time curves are given in Figure 4.9. A uniform vertical force  $f_z$  is applied on the whole outer surface of the ball, whereas the forces in  $y$ -direction are applied in positive and negative coordinate direction on one half of the surface of the ball, respectively, which will introduce a slight rotation of the ball. Following the discussion on the time integration schemes in the

second example, the Generalized Energy-Momentum Method is employed with  $\rho_\infty = 1.0$  for the time discretization. The spatial discretization is done with 256 eight-noded hexahedral surface oriented shell elements. In Figure 4.10, the motion of the ball is shown for different integration times, as well as the development of the total energy. It can be seen that exact energy conservation can be achieved with the used time integration scheme combined with the velocity update algorithm.

#### 4.7.4 Torus

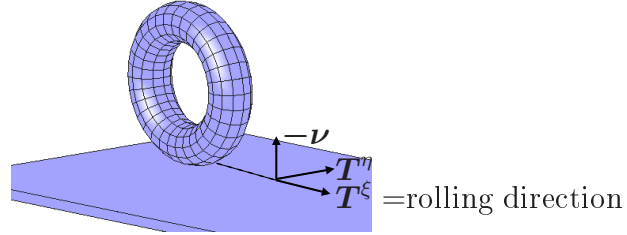


Figure 4.11: Choice of the normals and tangents for the torus example.

With this last example it is shown that the presented contact formulation can be easily extended to frictional contact conditions. The consideration is restricted to sticking friction where no tangential sliding is allowed. To show the performance of this approach, a simple rolling contact example is chosen where only sliding in rolling direction  $\mathbf{T}^\xi$  is suppressed, see Figure 4.11. For the implementation, the fourth block row of (4.25) must be split up in a  $\mathbf{T}^\xi$ -part and a  $\mathbf{T}^\eta$ -part. The latter remains unchanged since tangential sliding orthogonal to the sliding direction is allowed, but the  $\mathbf{T}^\xi$ -part changes to

$$\mathbf{T}_{\mathbb{A}_j} \mathbf{J}_{\mathbb{A}_j \mathbb{A}_j} \Delta \mathbf{U}_{\mathbb{A}_j}^{(j)} = \mathbf{0}$$

In addition, the energy conservation now demands to consider a tangential impulse. A torus is discretized with 384 elements and thrown onto a rigid floor. Geometry, material and loads are given in Figure 4.12, where the load  $f_r$  is applied at eight cross-section surfaces, equally distributed around the whole torus, to introduce a slight rotation of the torus. The translatory motion of the torus is initiated by the load  $f_t$  which acts at the inner surface of the torus with an inclined angle of  $22.5^\circ$  to the rigid obstacle. In Figure 4.13, the motion of the torus is shown in the case of pure normal contact constrains (no stick condition) and for the modified version with sticking condition in the direction of the motion. To see the different rotations of the toruses, one ring of elements is marked with a different color. It is obvious to see that in the case of pure normal contact, the torus slides over the rigid obstacle without getting any additional rotational impulse, whereas the example with stick conditions enforces the torus to roll on the surface. Both examples are calculated with the Generalized Energy-Momentum Method with  $\rho_\infty = 1.0$ .

The evolution of the energies are shown in Figure 4.14. Besides the fact that the total energy is conserved for both cases, it can be seen that the additional stick constraint will shift quite an amount of kinetic energy to the strain energy due to the larger deformations in this case.

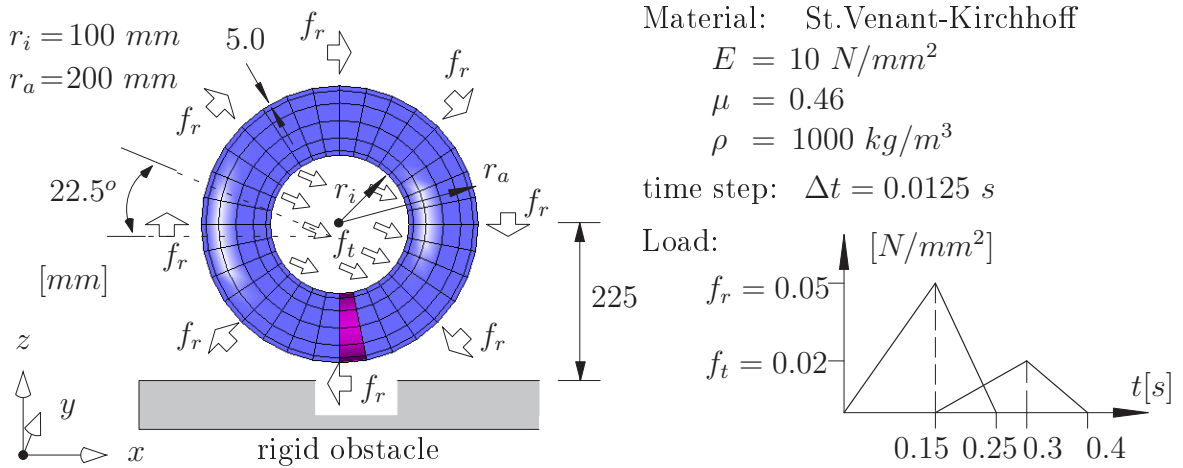


Figure 4.12: Torus - geometry, material and load.

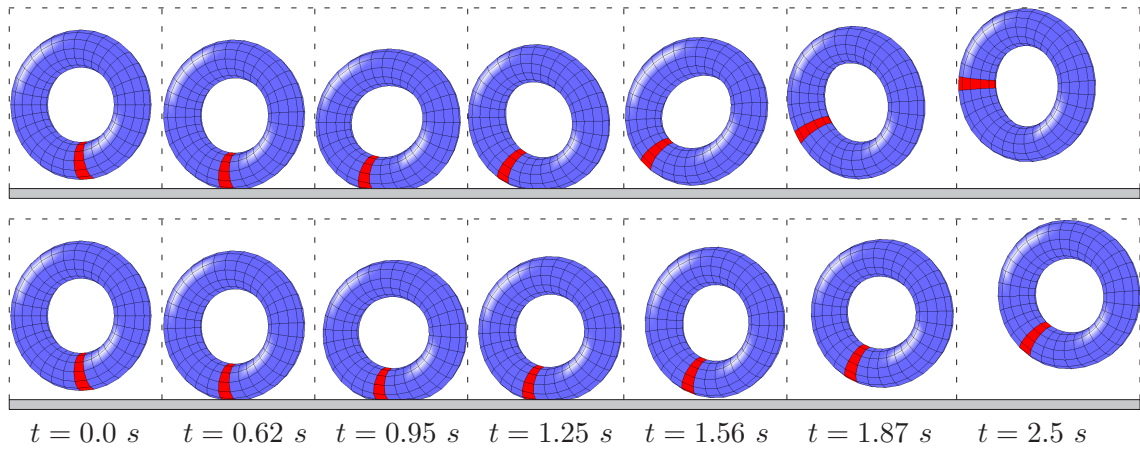


Figure 4.13: Motion of the torus - Generalized Energy-Momentum Method  $\rho_\infty = 1.0$  with (top row) and without (bottom row) stick condition.

## 4.8 Conclusions

A primal-dual active set strategy based on dual Lagrange multipliers for implicit non-linear analysis of dynamic contact problems is presented. For the time discretization, the Generalized Energy-Momentum Method and alternatively the Generalized- $\alpha$  Method



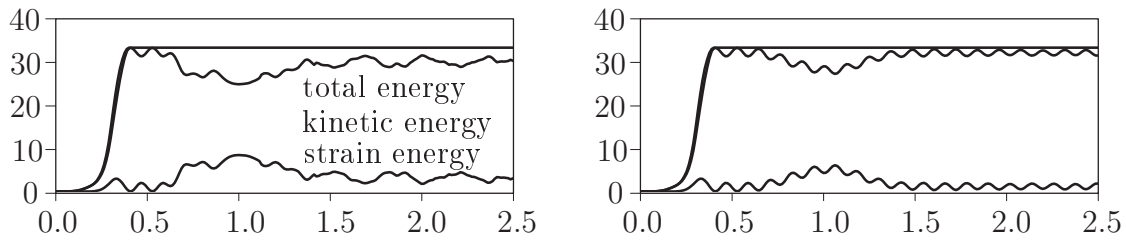


Figure 4.14: Torus - Energy  $E$  [J] vs time  $t$  [s] with (left) and without (right) stick condition.

are used. The spatial discretization is done by a surface oriented, three-dimensional 7-parameter finite shell element. An energy conserving framework using a velocity update, introduced in [77], is generalized to the application within the two mentioned time integration schemes. Furthermore it is adapted to the use of dual Lagrange multipliers for the impulse. The numerical examples show that the proposed strategy leads to a very robust algorithm which can guarantee total energy conservation if the Generalized Energy-Momentum Method is used with a spectral radius  $\rho_\infty = 1.0$ . In the last example, it is shown that this framework can be easily adapted to frictional contact problems.

It should be pointed out that the presented strategy does not need any user defined parameters such as a penalty parameter to enforce the contact constraints.



## 5 Velocity based contact problems

The aim of this chapter is to show the applicability of the pdASeS to velocity driven problems. The method is applied to the ring rolling of external spur gears, which is a rigid-plastic metal forming simulation. The existing displacement based contact formulation is adapted to a velocity based one. A numerical example is presented. The problem is restricted again to the case without inertia forces. Let  $\boldsymbol{\varphi}$  be the current, unknown deformation and define the Almansi strain  $\overline{\mathbf{E}}$  by

$$\overline{\mathbf{E}} := \frac{1}{2} (\nabla \boldsymbol{\varphi}(\mathbf{x}_0)(\mathbf{u})^\top + \nabla \boldsymbol{\varphi}(\mathbf{x}_0)(\mathbf{u}) - \nabla \boldsymbol{\varphi}(\mathbf{x}_0)(\mathbf{u})^\top \nabla \boldsymbol{\varphi}(\mathbf{x}_0)(\mathbf{u}))$$

Here is only important that  $\overline{\mathbf{E}}$  is a strain measure which is suited for the large deformation  $\boldsymbol{\varphi}$ , the workpiece is undergoing. The body  $\Omega$  gets into contact with a rigid obstacle  $\Omega_{obs}$  which has the boundary  $\Gamma_{obs}$ . By  $\mu^{plas}$ , the viscosity coefficient for the plastic flow is denoted and the bulk modulus  $\kappa^{plas}$  acts as a volume penalty which is usually set to a very high value to nearly guarantee the plastic incompressibility

$$\text{tr}(\dot{\overline{\mathbf{E}}}) \rightarrow 0$$

By  $p := -\frac{1}{d} \text{tr}(\boldsymbol{\sigma})$ , the pressure is denoted, and  $\mathbf{s} := \boldsymbol{\sigma} + p\mathbf{1}$  is the deviatoric stress. According to this, the constitutive law [4] formulated in the actual Cauchy stresses reads as

$$\mathbf{s} = 2\mu^{plas} \text{dev}(\dot{\overline{\mathbf{E}}}), \quad p = \kappa^{plas} \text{tr}(\dot{\overline{\mathbf{E}}})$$

Since the material law is velocity-driven, the Dirichlet boundary condition has to be given for the velocity  $\dot{\mathbf{u}}$ . The bivariate form of the internal energy is

$$f^{\text{rp}}(\dot{\mathbf{u}}, \mathbf{v}) := \int_{\varphi(\Omega)} \boldsymbol{\sigma}(\dot{\mathbf{u}}) : \overline{\mathbf{E}}(\mathbf{v})$$

Since the structural term  $f^{\text{rp}}(\cdot, \cdot)$  is velocity-driven, whereas the contact conditions are displacement-driven, one has to express (3.15) in terms of the velocity  $\dot{\mathbf{u}}_{n+1}$ . Therefore, a generalized trapezoidal rule is used with the parameter  $\gamma \in [0, 1]$

$$\Delta \mathbf{u}_{n+1} = \mathbf{u}_{n+1} - \mathbf{u}_n = (1 - \gamma) \Delta t \dot{\mathbf{u}}_n + \gamma \Delta t \dot{\mathbf{u}}_{n+1} \quad (5.1)$$

The difficulty of the large deformations is tackled in an updated Lagrange manner. Thus, the gap  $\hat{g}$  and the contact normal  $\boldsymbol{\nu}$  are chosen with respect to the last known configuration at time  $t_n$ . This means that  $\hat{g}(\mathbf{x}_n, t_{n+1})$  is the distance between the position  $\mathbf{x}_n$  at time  $t_n$  and its closest point projection onto  $\Gamma_{obs}(t_{n+1})$ . By  $\boldsymbol{\nu}(\mathbf{x}_n, t_{n+1})$ , the direction

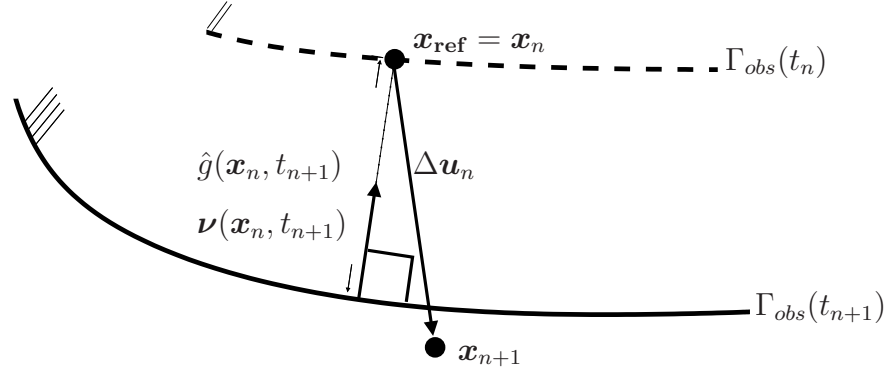


Figure 5.1: Gap and normal.

of this closest point projection is indicated, see Figure 5.1. Thus, gap and normal are defined as in Section 3.2.2 for the case  $\mathbf{x}_{\text{ref}} = \mathbf{x}_n$ . It should be remarked that the current position of the moving tool boundary  $\Gamma_{\text{obs}}(t_{n+1})$  is always known in advance. Now, a dynamic gap  $G_{n+1}$  is defined in such a way that the non-penetration condition (3.15) in terms of displacements at the current time step  $t_{n+1}$  is fulfilled. Setting

$$G_{n+1} := \frac{1}{\gamma \Delta t} (\hat{g}(\mathbf{x}_n, t_{n+1}) - (1 - \gamma) \Delta t \dot{\mathbf{u}}_n \cdot \boldsymbol{\nu}(\mathbf{x}_n, t_{n+1}))$$

for the case  $\gamma \in (0, 1]$ , the velocity-driven form of the non-penetration condition (3.15) is given by

$$\dot{u}_{\nu, n+1} - G_{n+1} \leq 0, \quad \lambda_{\nu, n+1} \geq 0, \quad \lambda_{\nu, n+1} (\dot{u}_{\nu, n+1} - G_{n+1}) = 0 \quad (5.2)$$

with the definitions  $\dot{u}_{\nu, n+1} := \dot{\mathbf{u}}_{n+1} \cdot \boldsymbol{\nu}(\mathbf{x}_n, t_{n+1})$  and  $\lambda_{\nu, n+1} := \boldsymbol{\lambda}^{n+1} \cdot \boldsymbol{\nu}(\mathbf{x}_n, t_{n+1})$ . Using (5.1), it can be readily verified that the displacement-driven non-penetration condition is fulfilled

$$\begin{aligned} \Delta \mathbf{u}_{n+1} \cdot \boldsymbol{\nu}(\mathbf{x}_n, t_{n+1}) &= [(1 - \gamma) \Delta t \dot{\mathbf{u}}_n + \gamma \Delta t \dot{\mathbf{u}}_{n+1}] \cdot \boldsymbol{\nu}(\mathbf{x}_n, t_{n+1}) \\ &\leq (1 - \gamma) \Delta t \dot{\mathbf{u}}_n \cdot \boldsymbol{\nu}(\mathbf{x}_n, t_{n+1}) + \gamma \Delta t G_{n+1} \\ &= \hat{g}(\mathbf{x}_n, t_{n+1}) \end{aligned}$$

*Remark:* For the special case  $\gamma = 0$  (explicit Euler), the desired non-penetration can only be guaranteed in the next time step  $t_{n+2}$  since the current displacements are completely determined by quantities from the last time step ( $\mathbf{u}_{n+2} - \mathbf{u}_{n+1} = \Delta \mathbf{u}_{n+2} = \Delta t \dot{\mathbf{u}}_{n+1}$ ). In this case, the choice

$$G_{n+2} := \frac{1}{\Delta t} \hat{g}(\mathbf{x}_{n+1}, t_{n+2}), \quad \dot{u}_{\nu, n+1} - G_{n+2} \leq 0$$

leads to

$$\Delta \mathbf{u}_{n+2} \cdot \boldsymbol{\nu}(\mathbf{x}_{n+1}, t_{n+2}) = \Delta t \dot{\mathbf{u}}_{n+1} \cdot \boldsymbol{\nu}(\mathbf{x}_{n+1}, t_{n+2}) \leq \hat{g}(\mathbf{x}_{n+1}, t_{n+2})$$

---

As only the frictionless case is considered, the LM cone  $\mathbb{M}^+$  is given by ( $\mathbb{W}$  is the trace space of  $[H_0^1]^d$  restricted to the potential contact boundary,  $\mathbb{M}$  is its dual space)

$$\mathbb{M}^+ := \{\boldsymbol{\mu} \in \mathbb{M} : \langle \boldsymbol{\mu}, \boldsymbol{\eta} \rangle \leq 0, \boldsymbol{\eta} \in \mathbb{W} \text{ with } \eta_\nu \leq 0\}$$

The time discretized variational formulation for the time step  $t_{n+1}$  reads as follows: find  $(\dot{\mathbf{u}}_{n+1}, \boldsymbol{\lambda}_{n+1}) \in [H_0^1]^d \times \mathbb{M}^+$  such that

$$\begin{aligned} f^{\text{rp}}(\dot{\mathbf{u}}_{n+1}, \mathbf{v}) + \langle \mathbf{v}, \boldsymbol{\lambda}_{n+1} \rangle &= 0, & \mathbf{v} &\in [H_0^1]^d \\ \langle \dot{u}_{\nu,n+1}, \mu_\nu - \lambda_{\nu,n+1} \rangle &\leq \langle G_{n+1}, \mu_\nu - \lambda_{\nu,n+1} \rangle, & \boldsymbol{\mu} &\in \mathbb{M}^+ \end{aligned} \quad (5.3)$$

Let  $\mathbf{K}^{\text{rp}}$  be the rigid plastic stiffness matrix resulting from the bivariate form  $f^{\text{rp}}(\cdot, \cdot)$  after discretization in space and consistent linearization of the geometric nonlinearity. Then, in analogy to (3.22), the algebraic form of (5.3) is given by

$$\mathbf{K}^{\text{rp}} \dot{\mathbf{U}}_{n+1} + \begin{bmatrix} \mathbf{0} \\ \mathbf{D} \end{bmatrix} \boldsymbol{\Lambda}_{n+1} = \mathbf{0}$$

together with the contact conditions

$$\dot{U}_{p\nu}^{n+1} - G_p^{n+1} \leq 0, \quad \Lambda_{p\nu}^{n+1} \geq 0, \quad (\dot{U}_{p\nu}^{n+1} - G_p^{n+1}) \Lambda_{p\nu}^{n+1} = 0, \quad \Lambda_{p\tau}^{n+1} = 0$$

Along with the definition  $\mathbf{N}_p^{n+1} := \boldsymbol{\nu}(\mathbf{x}_p^n, t_{n+1})$ , the abbreviations

$$\dot{U}_{p\nu}^{n+1} := \dot{\mathbf{U}}_p^{n+1} \cdot \mathbf{N}_p^{n+1}, \quad \Lambda_{p\nu}^{n+1} := \boldsymbol{\Lambda}_p^{n+1} \cdot \mathbf{N}_p^{n+1}, \quad \Lambda_{p\tau}^{n+1} := \boldsymbol{\Lambda}_p^{n+1} - \Lambda_{p\nu}^{n+1} \mathbf{N}_p^{n+1}$$

were used. As a numerical example, an incremental metal forming process is considered, see the left upper picture in Figure 5.2. One mobile tool presses the material into a rigid mold and forms a spur gear within 6000 time steps. The simulation was done in LARSTRAN. In Figures 5.2 and 5.3, one can see some snapshots out of the simulation process. The movement of the mobile inner tool is not plotted, but at the rigid outer tool, one can recognize that the method is performing well.

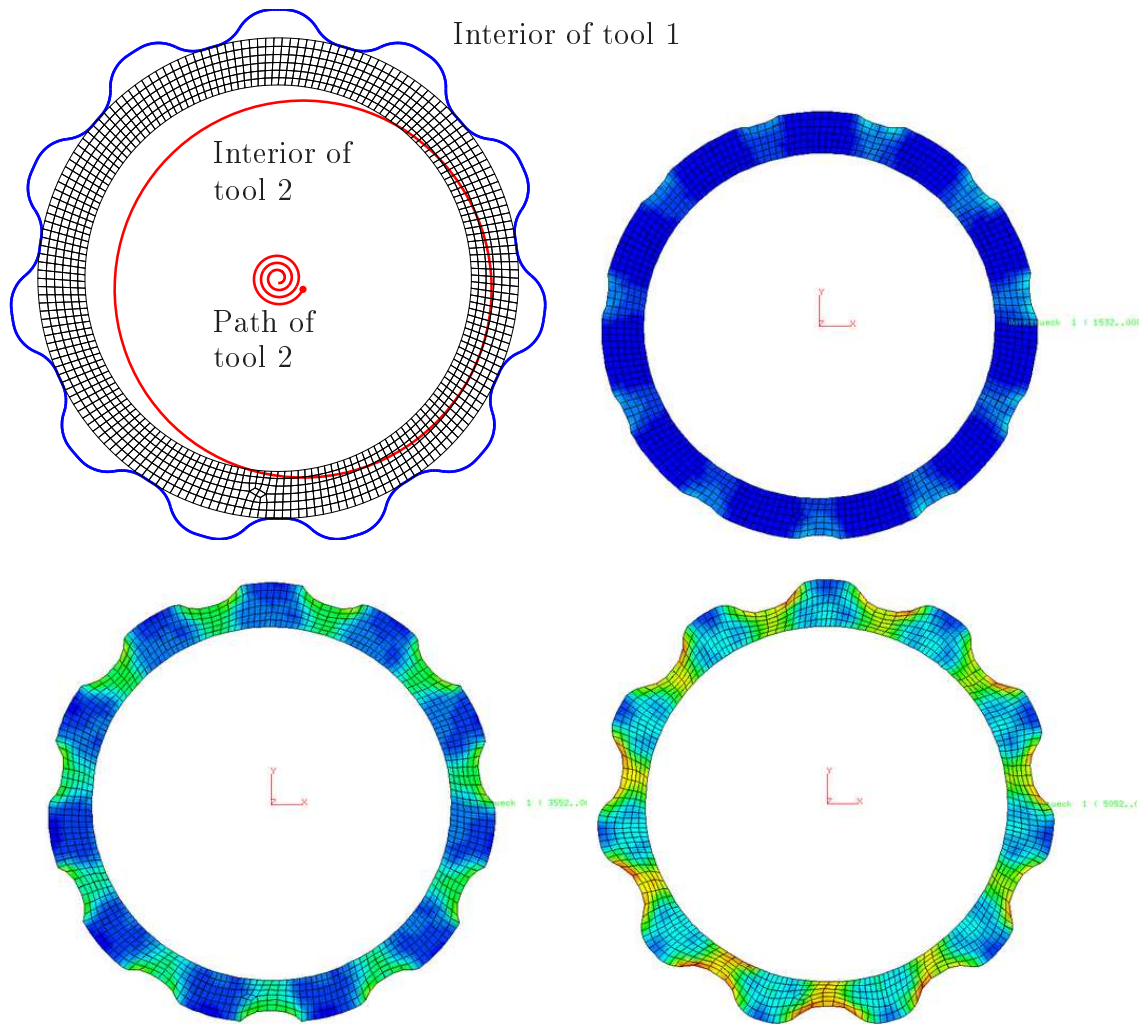


Figure 5.2: Ring rolling of external spur gears. Three snapshots of the simulation, the equivalent plastic strain is shown.

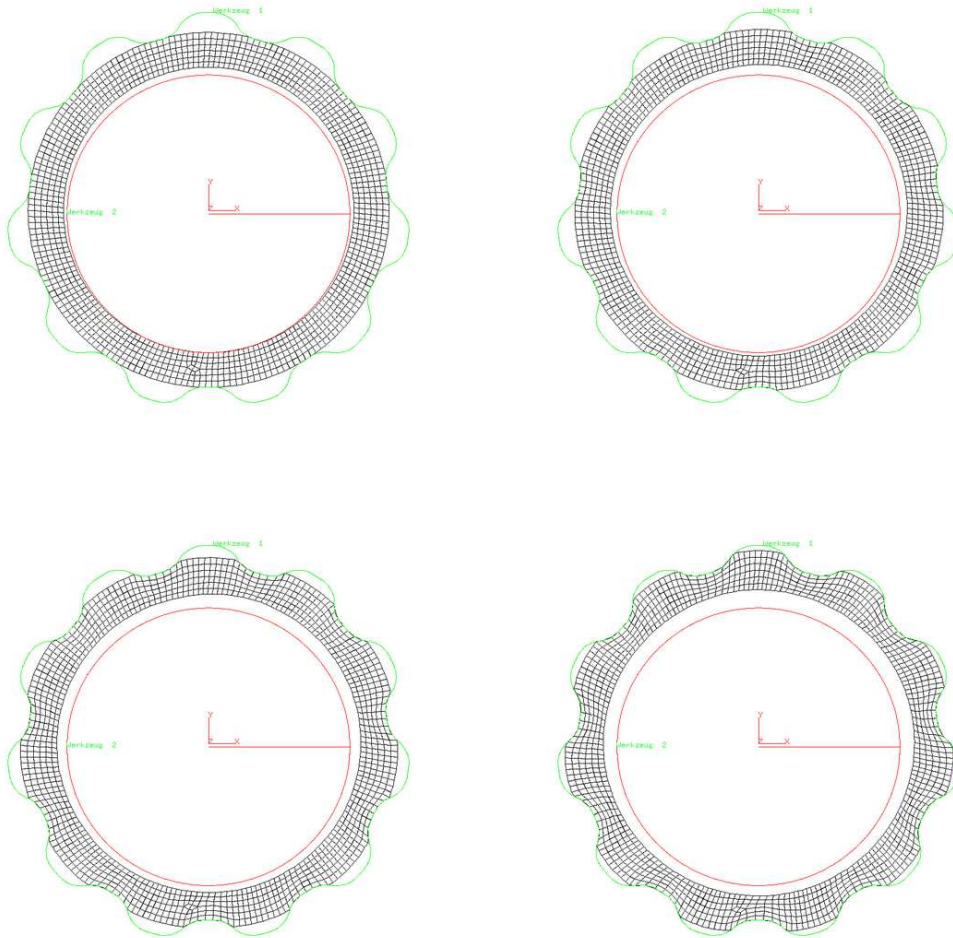


Figure 5.3: Ring rolling of external spur gears. Four snapshots of the simulation.





# 6 Coupling algorithm

The main idea to tackle the problems, mentioned in the introduction, is to use a ‘divide and conquer’ approach. The nonlinearities of the problem will be hidden from the global computation and left to a local computation. This is possible due to the fact that contact and plasticity are strictly localized in the class of forming processes, discussed in this work.

## 6.1 Dynamic ODDM, main concepts

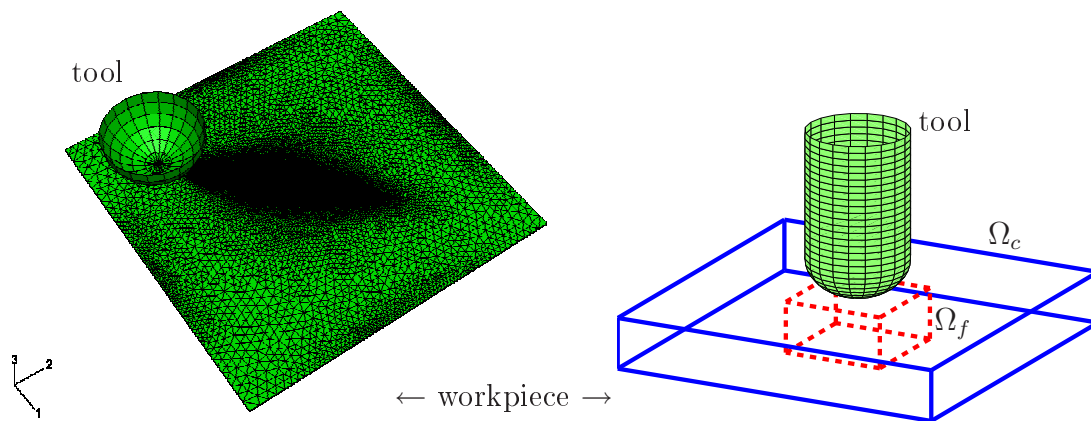


Figure 6.1: Left: ©IBF, Aachen, Abaqus simulation with triangular shell elements and adaptive mesh refinement, right: ODDM.

A dynamic overlapping domain decomposition (ODDM), shown in Figure 6.1, is employed. The workpiece (solid line,  $\Omega_c$ ) is discretized with a relatively (c)oarse mesh and the forming zone (dashed line,  $\Omega_f$ ) is meshed with a relatively (f)ine mesh which is assumed to overlap the zone of plastic deformation at any time. This latter term was defined exactly in Equation (2.13). With *dynamic* ODDM it is meant that within the time loop the fine grid is displaced due to the movement of the tool. In this chapter, the assumption of small deformations is still valid which means that  $\Gamma$  and  $\Omega$  always denote the undeformed reference configurations of contact boundary and structure. In Chapter 8, an outlook is given how this restriction can be omitted. Firstly, an elastic step is computed on the coarse grid. Then, this approximate coarse solution which is

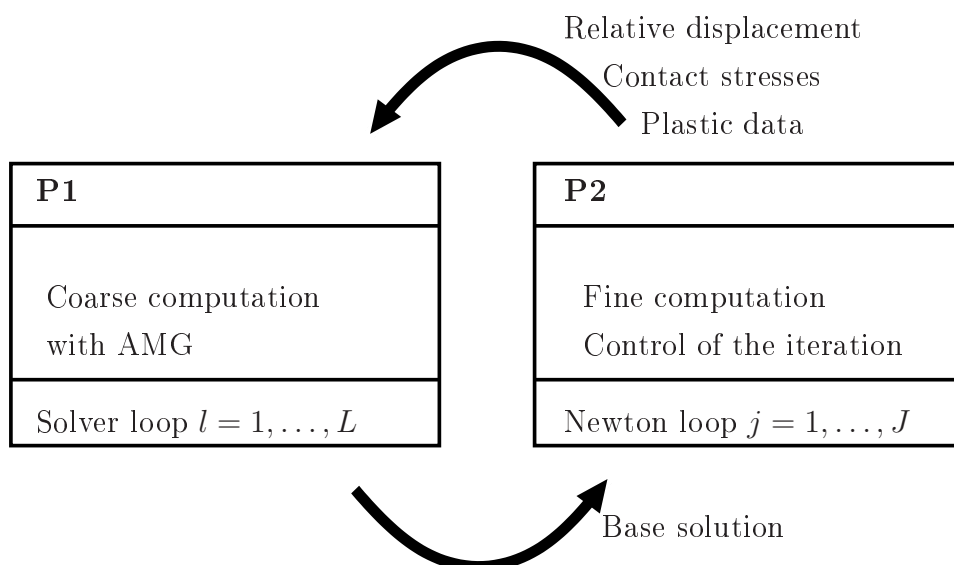


Figure 6.2: Coupling FE program P1 (coarse computation) and program P2 (fine computation).

not yet correct since the plastic effect was not taken into account at all, is prolonged and transmitted to the fine grid where a nonlinear plastic computation with contact is performed. After that, the plastic strain is computed and transferred as load correction to the coarse grid where a new and better approximation of the coarse solution can be computed and so on. The terms coarse and global are used interchangeably. The terms fine and local mean the same here and also the terms grid and mesh. Sometimes terms like fine patch or fine overlay grid are used and sometimes the tool is called punch. Of course the two FE domains have to interchange informations:

1. Plastic information, mainly the tensor of plastic deformation: The main idea is to use the additive split of the strain tensor into an elastic part and a plastic part within the global equilibrium and to perform the plastic computation only on the fine grid.
2. Contact information: The contact computation for the interaction of the tool and the workpiece is only performed on the fine grid and the contact stresses, computed there, are transferred to the coarse grid. A mortar operator is employed for this contact coupling.

These local nonlinear effects enter the coarse grid computation only on the right hand side. It should be stressed that the second point has to be modified in order to suit ISF operations with partial die or rigid support, see Figure 1.3 b and 1.3 c, since in these processes, there is not only a localized contact with the tool but also a global contact. For simplicity of notation, the following considerations are restricted to the case where

the contact between support and sheet does not need to be modeled, see Figure 1.3 a. It is assumed here that the clamping of the sheet can be modeled by Dirichlet boundary conditions. This strategy has the following advantages:

- All nonlinearities of the present problem can be separated from the coarse grid at the expense that some iterations between the fine and the coarse grid have to be done. In view of the numerical examples in Section 6.4.5, this additional cost turns out to be acceptable.
- Using the fine patch, the plastic zone and the contact problem can be resolved very well without
  - discretizing the workpiece with a global fine grid, which would lead to high computational cost. The problem is not only to solve the large systems of linear equations as one could argue that efficient iterative solvers are available anyway. It is also very time consuming to frequently reassemble the element tangential stiffness matrices. In principle it would be possible to use the knowledge about the tool path to reassemble only those small parts of the stiffness matrix which are subject to new plastification. This would necessitate considerable changes in the structure of the FE program which can be avoided by the algorithm proposed here. Actually, the algorithm can be implemented using two different programs P1 and P2, see Figure 6.2, and the interface between these two programs is small as will be shown in the following. In the numerical examples, one has either  $P1 = P2 = \text{Matlab}$  or  $P1 = \text{LARSTRAN}$  and  $P2 = \text{Matlab}$ . Furthermore, the convergence of commonly used nonlinear solvers is still mesh dependent, see for example [105], and therefore it is advantageous to keep the FE domain of nonlinear computation as small as possible.
  - performing an adaptive computation with remeshing and coarsening which is also quite expensive since the forming zone is moving through the whole workpiece, so remeshing is very complicated. In addition, the need for a transition zone between coarse zone and fine zone is eliminated. So the generation of unnecessarily many fine elements is avoided. Even approaches which use hanging nodes make this transition zone mandatory. Another point is that refining and coarsening makes complicated procedures for the data mapping of plastic history data between the previous mesh and the current mesh necessary.

## 6.2 Review of the literature

For an overview over error controlled discretization and model adaptivity, the reader is referred to [1, 6, 101, 104]. Advanced mesh superposition techniques have been around since the early nineties [39, 40, 80], often but not exclusively in the context of crack propagation. A very good overview over those methods can be found in the recent works [58, 116]. In [35, 71, 94], the coupling of two regions, one discretized with an h-version overlay mesh and one discretized with a p-version coarse mesh, is solved with a

block Gauss-Seidel iteration. The mechanical interpretation is quite similar: In principle body forces are interchanged iteratively in a block Gauss-Seidel manner between the two domains until convergence is reached. In this work, this idea is extended insofar as the fine patch is completely mobile and contact comes into play. An overlapping domain decomposition where the decomposition is introduced via a partition of unity of the energy and the coupling conditions are imposed via an Augmented Lagrangian has been recently been given in [90]. ODDMs have been analytically investigated in the linear case, see for example in [45] for the case of volumetric coupling. Of course more abstract frameworks like the nonlinear multigrid method or Jacobian-free Newton-Krylov methods, see the survey paper [64], should also be acknowledged here. Parallels can be drawn between the proposed scheme and partitioned algorithms and in particular sequentially staggered schemes for the simulation of fluid-structure-interaction. Finally, multiscale coupling methods like the so called bridging method [113], where a molecular dynamics model is coupled with a dynamic continuum model, employ similar ideas.

### 6.3 Coupling scheme

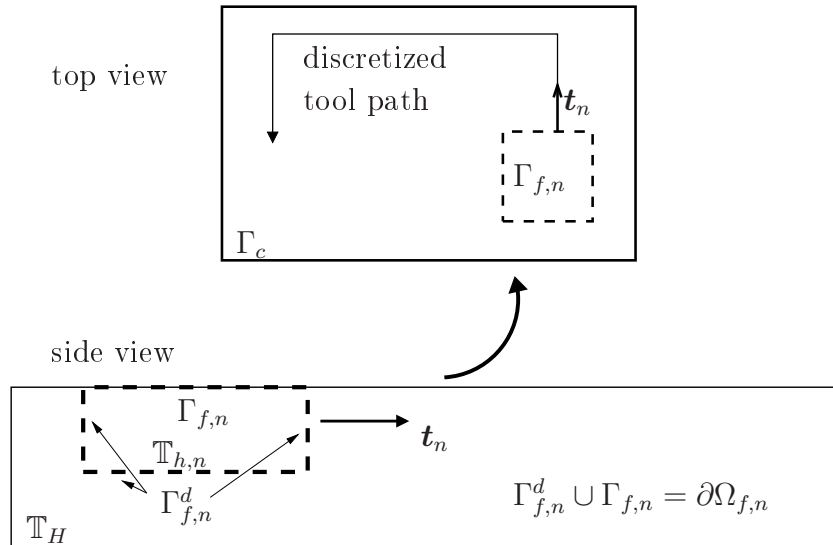


Figure 6.3: Dynamic ODDM.

Due to the domain decomposition, suggested in Section 6.1, the governing nonlinear equation system is discretized in the following way: Let

$$\mathbb{T}_H := \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{x} \text{ is a node in the coarse grid}\}$$

be the set of the coarse grid points and let

$$\mathbb{T}_{h,n} := \left\{ \mathbf{x} + \sum_{i=0}^n \mathbf{t}_i \in \mathbb{R}^3 : \mathbf{x} \text{ is a node in the initial position of the fine grid} \right\}$$

be the set of fine grid points. The potential contact zone and the Dirichlet boundary of the fine grid will be denoted by  $\Gamma_{f,n}$  and  $\Gamma_{f,n}^d$  which depend both on time since the fine grid is moving with the tool. By  $\Gamma_c$ , the potential contact zone of the coarse grid is indicated. The vector  $\mathbf{t}_n \in \mathbb{R}^3$  shall again mean the feed of the tool at time  $t_n$ , see also Figure 6.3.

### 6.3.1 Fine domain

On the fine domain  $\Omega_f$ , the full nonlinear elastoplastic contact problem is to be solved like in (3.18), with the only difference that the fine solution  $\mathbf{U}_h$  is a relative displacement w.r.t.  $\mathbf{U}_H$  which is brought up onto  $\mathbb{T}_h$  with the aid of a standard prolongation operator  $\mathbf{P} \in \mathbb{R}^{3n_h \times 3n_H}$  with  $n_h$  the number of fine nodes and  $n_H$  the number of coarse nodes. Find  $\mathbf{U}_h, \mathbf{\Lambda}_h$  such that  $\mathbf{u}_h|_{\Gamma_{f,n}^d} = \mathbf{0}$  and:

$$\mathbf{F}_h^{int}(\mathbf{P}\mathbf{U}_H + \mathbf{U}_h) + \mathbf{B}\mathbf{\Lambda}_h = \mathbf{0} \quad (6.1)$$

$$\mathbf{F}_h^{con}(\mathbf{P}\mathbf{U}_H + \mathbf{U}_h, \mathbf{\Lambda}_h) = \mathbf{0} \quad (6.2)$$

It should be remarked that (6.1) and (6.2) are solved in the condensed form but the uncondensed notation is more convenient here. The homogeneous Dirichlet condition on  $\Gamma_{f,n}^d$ , see Figure 6.3, guarantees the continuity of the composite solution  $\mathbf{P}\mathbf{U}_H + \mathbf{U}_h$ , which is of course a desired feature of the method.

### 6.3.2 Coarse domain

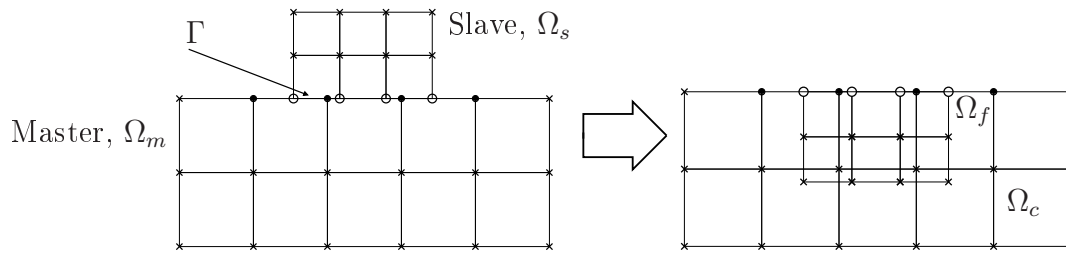


Figure 6.4: Use technology from the two body contact problem (left) for ODDM (right).

On the coarse domain  $\Omega_c$ , the linear problem is solved. Since the contact problem is not solved on  $\mathbb{T}_H$ , the discretized contact stresses in  $\mathbf{\Lambda}_h$  are coupled to the right hand side with the aid of the mortar operator  $\mathbf{M}$  which is derived in the following: In the

discretization of two-body contact problems, mortar methods have been around for a while, see for example [38, 57, 82, 92]. Like depicted schematically in Figure 6.4, two domains, denoted by  $\Omega_s$  (slave) and  $\Omega_m$  (master), come into contact at the interface  $\Gamma$  where the meshes are in general nonconforming. The idea is now to let  $\Omega_f$  play the role of  $\Omega_s$  while  $\Omega_c$  takes the role of  $\Omega_m$ . Let  $\boldsymbol{\nu}$  be the outward normal on  $\Gamma_c$ . The equilibrium stated on  $\Omega_c$  is:

$$\int_{\Omega_c} (\boldsymbol{\varepsilon}(\mathbf{u}_c) - \boldsymbol{\varepsilon}_p) : \mathbf{C}^{el} : \boldsymbol{\varepsilon}(\mathbf{v}) + \int_{\Gamma_c} (\boldsymbol{\sigma}(\mathbf{u}_c) \cdot \boldsymbol{\nu}) \cdot \mathbf{v} = 0, \quad \mathbf{v} \in [H_0^1(\Omega_c)]^3 \quad (6.3)$$

*Remark:* It is assumed here a priori that the fine correction  $\mathbf{u}_f$  is zero which is the case in the continuous setting and which also makes sense in the discrete equations since the global equilibrium is mostly influenced by the contact stresses and the plastic strain and not by the fine relative displacements. However, in Section 7.2, this problem will be discussed in more detail.

Returning to Equation (6.3), the contact stress  $\boldsymbol{\lambda} = \boldsymbol{\sigma}(\mathbf{u}_c) \cdot \boldsymbol{\nu}$  which is equal to  $\boldsymbol{\sigma}(\mathbf{u}_f) \cdot \boldsymbol{\nu}$  is discretized by dual Lagrange multipliers  $\psi_p^h$  which are defined on the slave side of  $\Gamma_{f,n}$ . The mortar operator  $\mathbf{M}$  couples between these  $\psi_p^h$  and the coarse standard shape functions on the potential contact zone  $\Gamma_c$  and is defined by:

$$\mathbf{M}[p, q] := \int_{\Gamma_c} \psi_p^h \phi_q^H \mathbf{I}_3 \quad (6.4)$$

so it is well suited for the task of transmitting the contact stresses from the fine grid to the coarse grid. Now, it is assumed that the contact stress  $\boldsymbol{\lambda}$  as well as the plastic strain  $\boldsymbol{\varepsilon}_p$  are known from the fine grid computation. Then, with the definition of the load correction term

$$l_c^{pl}(\mathbf{v}) := - \int_{\Omega_c} \boldsymbol{\varepsilon}_p : \mathbf{C}^{el} : \boldsymbol{\varepsilon}(\mathbf{v})$$

one can proceed from Equation (6.3):

$$\int_{\Omega_c} \boldsymbol{\varepsilon}(\mathbf{u}_c) : \mathbf{C}^{el} : \boldsymbol{\varepsilon}(\mathbf{v}) = - \int_{\Gamma_c} \boldsymbol{\lambda} \cdot \mathbf{v} - l_c^{pl}(\mathbf{v}), \quad \mathbf{v} \in [H_0^1(\Omega_c)]^3 \quad (6.5)$$

Discretizing this linear problem one arrives at: Find  $\mathbf{U}_H$  s.t.  $\mathbf{u}_H|_{\Gamma_c^d} = \mathbf{0}$ :

$$\mathbf{A}_H \mathbf{U}_H = -\mathbf{F}_H^{pl} - \mathbf{M}^\top \boldsymbol{\Lambda}_h \quad (6.6)$$

where  $\mathbf{A}_H$  is the elastic stiffness matrix belonging to the coarse discretization, and  $\mathbf{F}_H^{pl}$  can be obtained by standard finite element assembly of  $l_c^{pl}$ . Since the transfer values of  $\mathbf{U}_H$  as well as of  $\boldsymbol{\varepsilon}^p$  and  $\boldsymbol{\Lambda}_h$  are computed with some error due to the decoupling, an iteration between  $\Omega_f$  and  $\Omega_c$  needs to be performed. The plastic data is stored on a global storage grid  $\mathbb{T}_{\hat{h}}$  with the mesh size  $\hat{h}$ . Finally the two-grid algorithm is described in the next section.

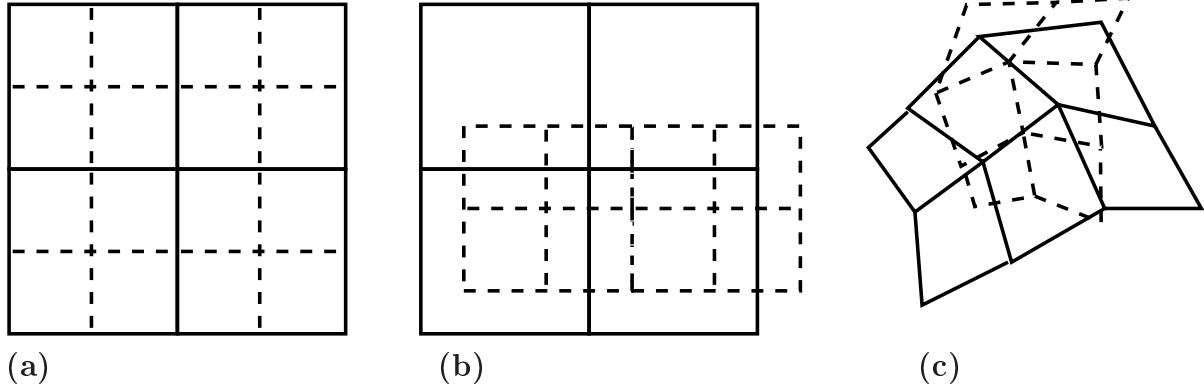


Figure 6.5: Hierarchical situation (a), non-hierarchical situation (b), unstructured situation (c), coarse interface grid (solid line), fine interface grid (dashed line).

### 6.3.3 Solving the nonlinear problem with a coarse global and a fine local mesh

First of all, some definitions and remarks have to be made: Let  $\mathbf{P}^{GPS \rightarrow c}$  be the interpolation operator for the plastic data between (G)auss (p)oints on the (c)oarse grid and on the (s)orage grid  $\mathbb{T}_{\hat{h}}$ . Analogously  $\mathbf{P}^{GPS \rightarrow F}$  and  $\mathbf{P}^{GF \rightarrow s}$  transfer data between (f)ine and (s)orage Gauss points. The discrete plastic history data defines the set

$$\mathbb{E}_{h,n} := \{[\varepsilon^p(\boldsymbol{\xi}, t_n), \alpha(\boldsymbol{\xi}, t_n)] : \boldsymbol{\xi} \in \mathbb{G}_{h,n}\}$$

and the set  $\mathbb{E}_{\hat{h},n}$  accordingly. Herein

$$\mathbb{G}_{h,n}, \mathbb{G}_H \text{ and } \mathbb{G}_{\hat{h}}$$

denote the positions of the fine, coarse and storage grid integration points at time  $t_n$ , respectively. With  $(\cdot)^{(k)}$ , the  $k$ th iterate of  $(\cdot)$  within the coarse-fine iteration is indicated. If a notation like  $(\cdot)^{(\bullet, \star)}$  is used, then  $\bullet$  denotes the index of a coarse-fine iteration and  $\star$  indicates the index of the Newton iteration. Let  $\text{TOL}_{newt}$  denote the tolerance for the Newton stopping criterion and  $j_{max}$  the maximum number of Newton iterations to be performed. The implementation of the aforementioned prolongation and restriction operators in the case of structured, axis-parallel grids is straightforward ('do-not-search-just-compute'). As a consequence, the assembly of those operators does not contribute very much to the overall computation time. To keep the efficiency in the case of unstructured grids, see Figure 6.5(c), one has to employ hierarchical search algorithms like in two body contact problems with very large sliding or rigid body motions, see [87, 114]. Two remarks have to be made:

- In step 6, when the plastic history data of the last time step is read from the storage grid, one has to take the new position of the displaced fine grid into account.

---

**Algorithm 3** Coupling.

---

```

1: assemble  $\mathbf{P}^{GPS \rightarrow C} := \mathbf{P}^{GPS \rightarrow C}(\mathbb{G}_{\hat{h}}, \mathbb{G}_H)$ 
2: time loop
3: for  $n = 1, \dots$  do
4:   perform initialization in Algorithm 4
5:   coarse-fine iteration
6:   for  $k = 1, \dots$  do
7:     assemble  $\mathbf{F}_{H,n+1}^{pl} := \mathbf{F}^{pl}(\mathbf{P}^{GPS \rightarrow C}[\mathbb{E}_{\hat{h},n+1}^{(k-1)}])$ 
8:     solve  $\mathbf{A}_H \mathbf{U}_{H,n+1}^{(k)} = -\mathbf{M}_{n+1}^\top \boldsymbol{\Lambda}_{h,n+1}^{(k-1)} - \mathbf{F}_{H,n+1}^{pl}$ 
9:      $\mathbb{A}_{n+1}^{(k,0)} := \begin{cases} \mathbb{A}_{n+1}^{(k-1)} & \text{update strategy (U1)} \\ \emptyset & \text{otherwise} \end{cases}$ 
10:    init. fine Newton: use as starting iterate
                                
$$\mathbf{U}_{h,n+1}^{(k,0)} := \begin{cases} \mathbf{U}_{h,n+1}^{(k-1)} & \text{update strategy (U2)} \\ \mathbf{0} & \text{otherwise} \end{cases}$$

11:     $j_{max} := \begin{cases} 1 & \text{update strategy (U3)} \\ \infty & \text{otherwise} \end{cases}$ 
12:    using Algorithm 1 with the input
                                
$$[\mathbb{A}_{n+1}^{(k,0)}, j_{max}, \text{TOL}_{newt}, \mathbf{U}_{h,n+1}^{(k,0)} + \mathbf{P}[\mathbf{U}_{H,n+1}^{(k)}], \mathbb{E}_{h,n}, \text{DIRECT} = 1]$$

    to find
                                
$$(\boldsymbol{\Lambda}_{h,n+1}^{(k)}, \mathbf{U}_{h,n+1}^{(k)}, \mathbb{E}_{h,n+1}^{(k)}, \mathbb{A}_{n+1}^{(k)})$$

    such that Equations (6.1) and (6.2) are fulfilled
13:    update  $\mathbb{E}_{\hat{h},n+1}^{(k)} := \mathbf{P}_{n+1}^{GPF \rightarrow S}[\mathbb{E}_{h,n+1}^{(k)}]$ 
14:     $k := k + 1$ 
15:    if relative coarse-fine error  $< \text{TOL}_{cf}$  goto 16 else goto 7 endif
16:    end coarse-fine iteration
17:  end for
18:  if  $\mathbb{G}(\Omega_{pl,n+1}) \subset \mathbb{G}_{h,n+1}$  then  $n := n + 1$  else expand fine grid and goto 1
19: end for

```

---



**Algorithm 4** Initialization.

- 
- 1: displace fine grid  $\mathbb{T}_{h,n+1} := \mathbb{T}_{h,n} + \mathbf{t}_{n+1}$
  - 2: assemble mortar operator  $\mathbf{M}_{n+1} := \mathbf{M}(\mathbb{T}_{h,n+1}, \mathbb{T}_H)$
  - 3: assemble  $\mathbf{P}_{n+1} := \mathbf{P}(\mathbb{T}_H, \mathbb{T}_{h,n+1})$
  - 4: assemble  $\mathbf{P}_{n+1}^{GPF \rightarrow S} := \mathbf{P}^{GPF \rightarrow S}(\mathbb{G}_{h,n+1}, \mathbb{G}_{\hat{h}})$
  - 5: assemble  $\mathbf{P}_{n+1}^{GPS \rightarrow F} := \mathbf{P}^{GPS \rightarrow F}(\mathbb{G}_{\hat{h}}, \mathbb{G}_{h,n+1})$
  - 6: read fine data from  $\mathbb{G}_{\hat{h}}$ :  $\mathbb{E}_{h,n} := \mathbf{P}_{n+1}^{GPS \rightarrow F}[\mathbb{E}_{\hat{h},n}]$
  - 7: init.  $\mathbb{E}_{h,n+1}^{(0)} := \mathbb{E}_{\hat{h},n}$ ,  $\mathbb{A}_{n+1}^{(0)} := \mathbb{A}_n$ ,  $\mathbb{\Lambda}_{h,n+1}^{(0)} := \mathbf{0}$ ,  $\mathbf{U}_{h,n+1}^{(0)} = \mathbf{0}$
- 

- Without going into details: In step 18, it is checked whether plastification occurs somewhere outside the fine domain. In this case, one has to distinguish between two cases:
  1. The plastic zone is just a bit larger than assumed: Expand the fine domain accordingly.
  2. New plastification occurred far outside the fine domain: The computation has to be stopped, since the intention of incremental forming processes is to have plastification only in the forming zone, otherwise it is impossible to control these operations. In this case, the tool path has to be reconsidered carefully.

With the remark that a careful use of the update strategies (U1)-(U3) is crucial to the performance of the algorithm, see the numerical example in Part 6.4.5, this section is concluded.

## 6.4 Numerical results

In this section, some numerical examples are presented to show the performance of the proposed coupling methods.

### 6.4.1 Material data

The following set of material data is chosen for all the subsequent numerical examples:  $E = 69000$ ,  $\nu = 0.33$ . For the examples with elastoplastic material behavior, linear isotropic hardening is assumed with  $Y(\alpha) = \sigma_0 + K\alpha$  and with the additional parameters  $K = 2538.930$  and  $\sigma_0 = 279.618$ . The process control parameters are  $\text{TOL}_{newt} = \text{TOL}_{cf} = 10^{-8}$ .

### 6.4.2 Elastic example, non-hierarchical case

It will be demonstrated in this example that if one considers only the contact problem and not the material nonlinearity, the fine grid is able to move independently from the coarse grid and does not necessarily have to be a local refinement of the global coarse grid. A

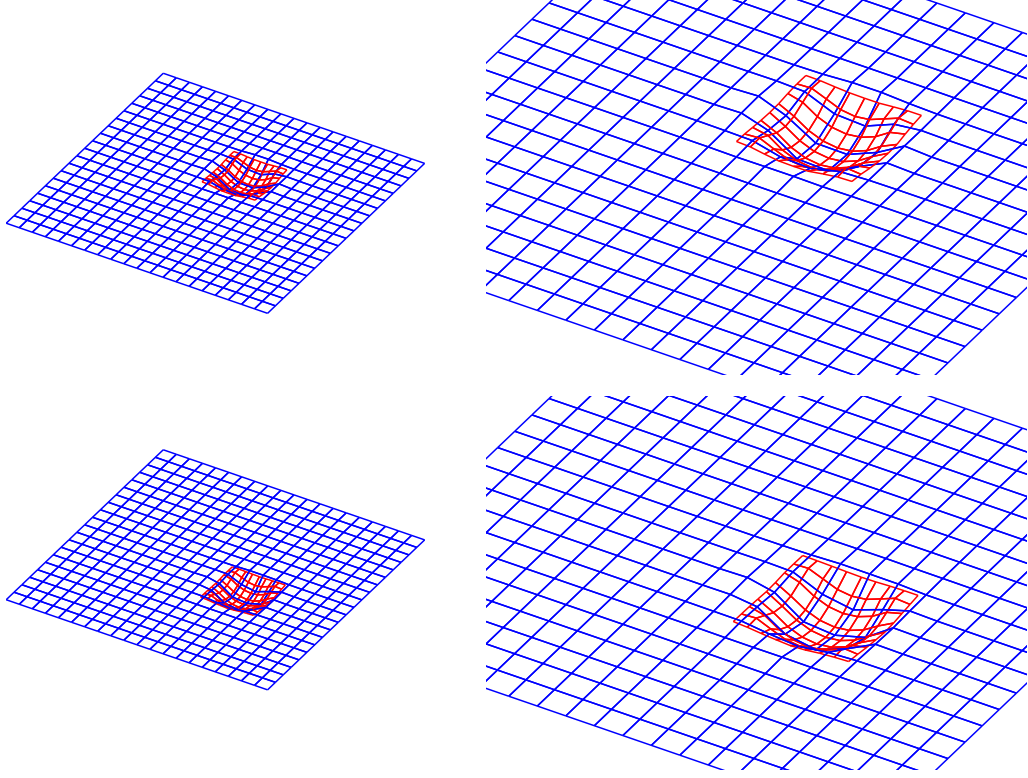


Figure 6.6: Elastic non-hierarchical example, time step 6 (bottom) and 11 (top), scaling factor = 100, contact surface (left), zoom (right).

sphere headed tool (radius = 200) is indenting the workpiece ( $\Omega_c = [0, 20] \times [0, 20] \times [0, 1]$ ) by  $d_z = 0.01$  and then moving in direction  $[0.77 \ 0.44 \ 0]$  over the workpiece. Two time steps out of the process are shown in Figure 6.6. The mesh sizes are  $h = 0.5$  and  $H = 1.0$ .

### 6.4.3 Numerical testing of the mortar operator

The idea is to separate the test of the mortar coupling from the structural problem. We consider three surface loads

$$\mathbf{f}_{ext}^{c-f} : \Gamma_c - \Gamma_f \rightarrow \mathbb{R}^3, \quad \mathbf{f}_{ext}^c : \Gamma_c \rightarrow \mathbb{R}^3, \quad \mathbf{f}_{ext}^f : \Gamma_f \rightarrow \mathbb{R}^3$$

being defined such that:

$$\mathbf{f}_{ext}^{c-f} = \mathbf{f}_{ext}^c = \mathbf{f}_{ext}^f = \begin{bmatrix} f_x^{ext} \neq 0 \\ f_y^{ext} \neq 0 \\ f_z^{ext} \neq 0 \end{bmatrix} \quad (6.7)$$

Now let  $\mathbb{T}_h$  and  $\mathbb{T}_H$  be the overlapping and not necessarily nested triangulations of  $\Gamma_f$  and  $\Gamma_c$  with shape functions  $\phi_1^h, \dots, \phi_{n_h}^h$  and  $\phi_1^H, \dots, \phi_{n_H}^H$ . Then the standard finite element

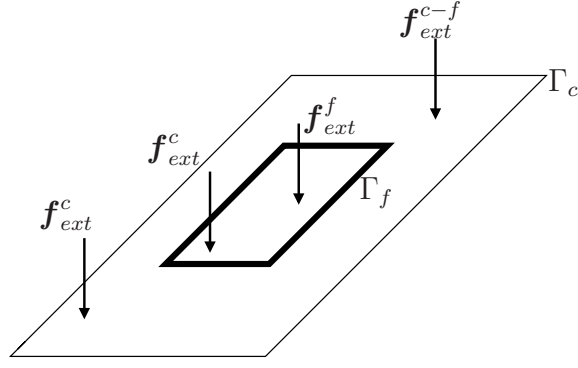


Figure 6.7: Mortar test setting.

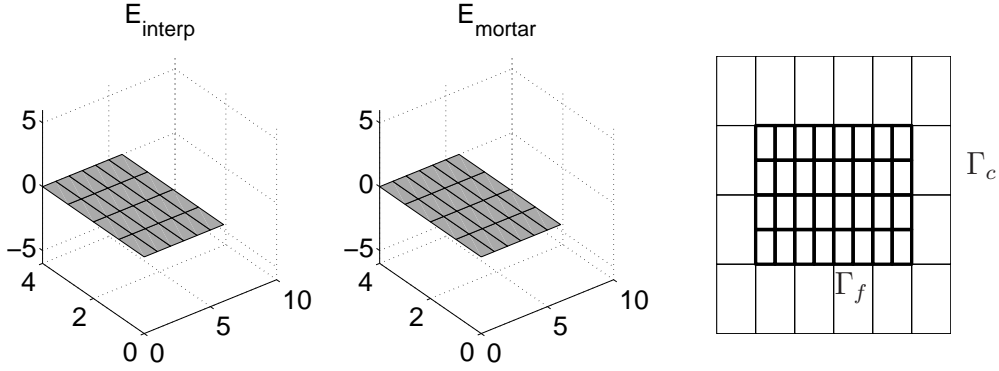


Figure 6.8: Z-component of the error of the standard interpolation coupling (left) and of the mortar coupling (middle) plotted on the coarse grid, hierarchical case.

assemblies of these three surface loads are  $\mathbf{F}_{ext}^h$ ,  $\mathbf{F}_{ext}^H$  and  $\mathbf{F}_{ext}^{H-h}$ . Let  $\mathbf{P} := \mathbf{P}(\mathbb{T}_h, \mathbb{T}_H)$  be the Lagrange interpolation operator as before. In terms of (3.23) it can be written:

$$\mathbf{\Lambda}_h = \mathbf{D}^{-1} \mathbf{F}_{ext}^h \quad (6.8)$$

A test which must necessarily be fulfilled by the mortar operator  $\mathbf{M}$  is the following: In the continuous setting, bringing up the constant load  $\mathbf{f}_{ext}^f$  on the fine interface  $\Gamma_f$  and the same constant load  $\mathbf{f}_{ext}^{c-f}$  on  $\Gamma_c - \Gamma_f$  results in the same constant load  $\mathbf{f}_{ext}^c$ , brought upon the whole coarse interface  $\Gamma_c$ . However, it turns out that in the discrete setting, the standard interpolation operator  $\mathbf{P}$  fails to transfer  $\mathbf{F}_{ext}^h$  correctly to the coarse grid. As soon as some coarse shape function  $\phi_i^H$  cannot be reproduced by the fine ones  $\phi_j^h$  any more, the equation

$$\mathbf{F}_{ext}^{H-h} + \mathbf{P}^\top \mathbf{F}_{ext}^h = \mathbf{F}_{ext}^H \quad (6.9)$$

does not hold any more. The patch test is passed, if  $\mathbf{M}$  is used: The equation

$$\mathbf{F}_{ext}^{H-h} + \mathbf{M}^\top \mathbf{\Lambda}_h = \mathbf{F}_{ext}^H \quad (6.10)$$

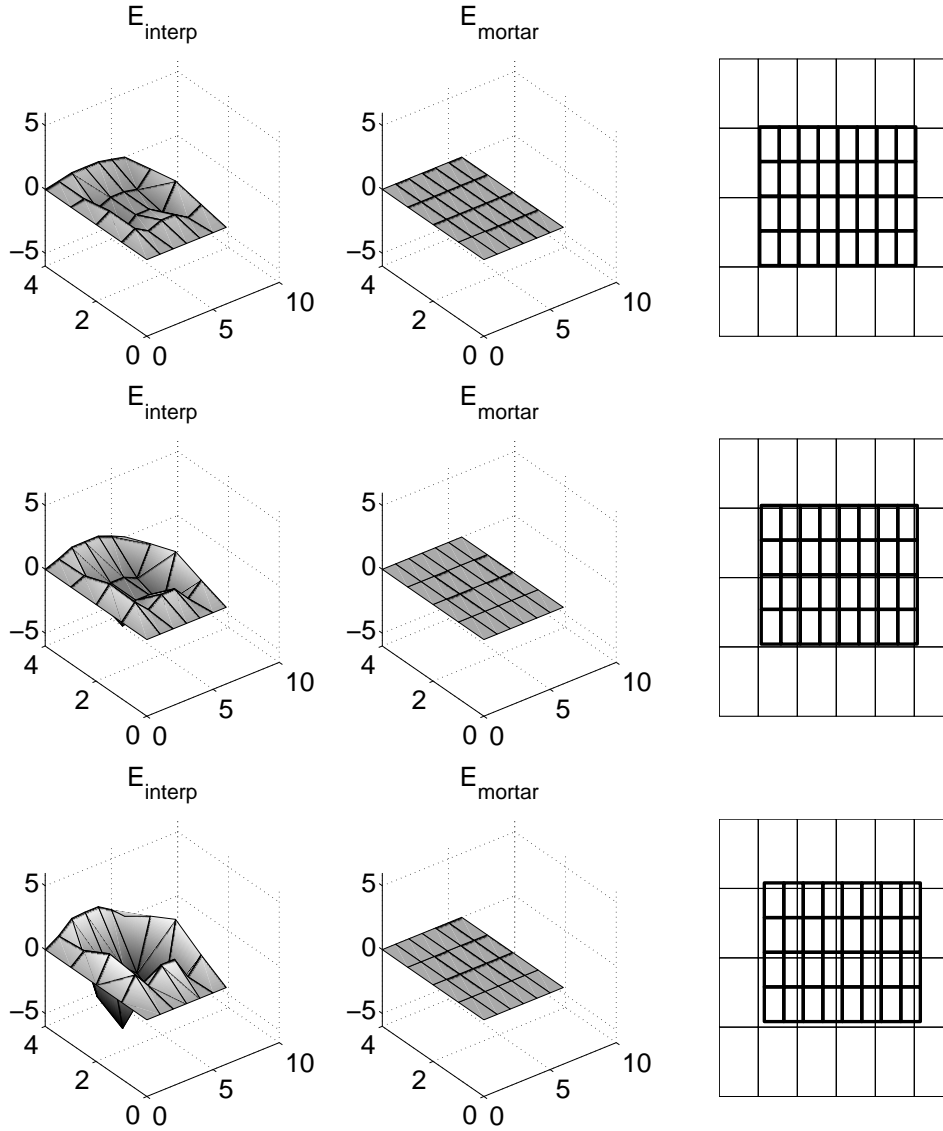


Figure 6.9: Z-component of the error of the standard interpolation coupling and of the mortar coupling plotted on the coarse grid, non-hierarchical case, no scaling,  $h = 0.5$ ,  $H = 1.0$ ,  $\Gamma_c = [0, 6] \times [0, 4]$ , starting position of  $\Gamma_f = [1, 5] \times [1, 3]$ .

holds even in the non-hierarchical situation, see Figure 6.5(b). One feature of this patch test is that it is possible to test the transfer of stress/force fields with more than one nonzero component, unlike classical patch tests like described in, e.g., [89]. In Figure 6.8, one can see that in the hierarchical case, see Figure 6.5(a), the interpolation operator  $\mathbf{P}$  transfers  $\mathbf{f}_{ext}$  as accurately as  $\mathbf{M}$  does. In Figure 6.9, the fine interface  $\Gamma_f$  is displaced

by the vector  $0.05 \mathbf{d}$ ,  $0.1 \mathbf{d}$  and  $0.2 \mathbf{d}$  with

$$\mathbf{d} := \begin{bmatrix} 0.77 \\ 0.4 \end{bmatrix}$$

Even in this non-hierarchical situation, the transfer error produced by  $\mathbf{M}$  is zero while that stemming from  $\mathbf{P}$  is increasing. It is possible to pass this patch test in the totally unstructured situation with methods introduced in [42]. The constant test load was  $\mathbf{f}_{ext} = [20 \ 37 \ 100]$  in this example. In the left picture of Figure 6.9, the z-component of the error

$$\mathbf{E}_{interp} := \mathbf{F}_{ext}^{H-h} + \mathbf{P}^\top \mathbf{F}_{ext}^h - \mathbf{F}_{ext}^H$$

is plotted on the coarse grid of  $\Gamma_c$ , while in the middle, the z-component of the error

$$\mathbf{E}_{mortar} := \mathbf{F}_{ext}^{H-h} + \mathbf{M}^\top \mathbf{\Lambda}_h - \mathbf{F}_{ext}^H \quad (6.11)$$

is displayed.

**Remark to the meaning of the patch test** Although the proof that  $\mathbf{E}_{mortar} = \mathbf{0}$  is a direct consequence of the theory of the mortar method, it is written down here in a direct form to explain what the practical meaning of the mortar operator is. It turns out that  $\mathbf{M}$  is constructed in such a way that it incorporates the information about the non-conforming overlap between  $\Gamma_h$  and  $\Gamma_H$ . The notations  $\phi_{i_r} := \phi_i \mathbf{e}_r$  and  $\psi_{j_r} := \psi_j \mathbf{e}_r$  with the unit vectors  $\mathbf{e}_r, r = 1, 2, 3$  are introduced. Due to the test setting and (6.7), it can be stated that the following continuous equation is true:

$$\mathbf{f}_{ext}^{c-f} + \mathbf{f}_{ext}^f - \mathbf{f}_{ext}^c \equiv \mathbf{0} \in \mathbb{R}^3$$

So for all  $i = 1, \dots, n_h, r = 1, 2, 3$  holds:

$$\begin{aligned} & \int_{\Gamma_H} \mathbf{f}_{ext}^{c-f} \cdot \phi_{i_r}^H + \int_{\Gamma_H} \mathbf{f}_{ext}^f \cdot \phi_{i_r}^H - \int_{\Gamma_H} \mathbf{f}_{ext}^c \cdot \phi_{i_r}^H = 0 \\ \Rightarrow & \int_{\Gamma_H - \Gamma_h} \mathbf{f}_{ext}^{c-f} \cdot \phi_{i_r}^H + \int_{\Gamma_h} \mathbf{f}_{ext}^f \cdot \phi_{i_r}^H - \int_{\Gamma_H} \mathbf{f}_{ext}^c \cdot \phi_{i_r}^H = 0 \end{aligned}$$

The constant force density  $\mathbf{f}_{ext}^f$  can be reproduced by dual Lagrange multipliers  $\psi_{j_r}$

$$\mathbf{f}_{ext}^f = \sum_{j \in \mathbb{T}_h, r=1,2,3} \mathbf{\Lambda}_h[j_r] \psi_{j_r}^h \quad (6.12)$$

as well as the standard LMs which are identical to the fine standard shape functions  $\phi_{j_r}$  in this setting:

$$\mathbf{f}_{ext}^f = \sum_{j \in \mathbb{T}_h, r=1,2,3} \mathbf{F}_{ext}^h[j_r] \phi_{j_r}^h \quad (6.13)$$

For example with the first variant, one can proceed:

$$\int_{\Gamma_H - \Gamma_h} \mathbf{f}_{ext}^{c-f} \cdot \phi_{i_r}^H + \int_{\Gamma_h} \left( \sum_{j \in \mathbb{T}_h, r=1,2,3} \mathbf{\Lambda}_h[j_r] \psi_{j_r} \right) \cdot \phi_{i_r}^H - \int_{\Gamma_H} \mathbf{f}_{ext}^c \cdot \phi_{i_r}^H = 0$$

Due to the definitions, this is:

$$\mathbf{F}_{ext}^{H-h}[i_r] + \sum_{j \in \mathbb{T}_h, r=1,2,3} \mathbf{M}^\top[i_r, j_r] \mathbf{\Lambda}_h[j_r] - \mathbf{F}_{ext}^H[i_r] = \mathbf{0} \quad (6.14)$$

For the complete vectors:

$$\mathbf{F}_{ext}^{H-h} + \mathbf{M}^\top \mathbf{\Lambda}_h - \mathbf{F}_{ext}^H = \mathbf{0} \quad (6.15)$$

By comparing (6.12) and (6.13), one can easily see that

$$\mathbf{M}^\top \mathbf{\Lambda}_h = \mathbf{M}^\top \mathbf{D}^{-1} \mathbf{F}_{ext}^h$$

Note that it is also possible to use the standard LMs for discretizing  $\mathbf{f}_{ext}^f$ . Then the coupling operator is not  $\mathbf{M}^\top$  but  $\mathbf{M}^\top \mathbf{D}^{-1}$  since the coefficients in  $\mathbf{F}_{ext}^h$  are different from those in  $\mathbf{\Lambda}_h$ . Only in the hierarchical situation, the coupling operator  $\mathbf{M}^\top \mathbf{D}^{-1}$  equals the standard pointwise interpolation operator  $\mathbf{P}$  as one can easily derive using the Equations (3.10). By  $\mathbf{x}_p^h$ , the coordinates of the  $p$ -th fine node are denoted.

$$\begin{aligned} \mathbf{M}[p, q] &= \int_{\Gamma_H} \psi_p^h \phi_q^H \mathbf{1} \\ &= \int_{\Gamma_H} \psi_p^h \left( \sum_{k=1}^4 \phi_q^H(\mathbf{x}_k^h) \phi_k^h \right) \mathbf{1} \\ &= \phi_q^H(\mathbf{x}_p^h) \int_{\Gamma_H} \phi_p^h \mathbf{1} \\ &= \phi_q^H(\mathbf{x}_p^h) \mathbf{D}[p, p] \\ &= \mathbf{P}[p, q] \mathbf{D}[p, p] \end{aligned}$$

With the operator  $\mathbf{P}$  defined accordingly:

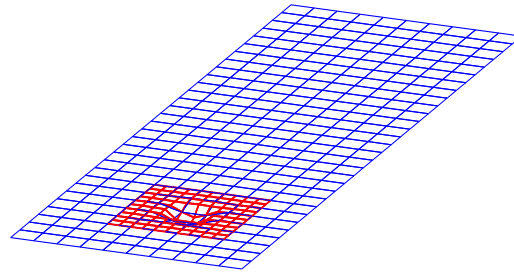
$$\mathbf{P}[p, q] = \phi_q^H(\mathbf{x}_p^h) \mathbf{1} \quad (6.16)$$

#### 6.4.4 Relative error

This numerical example considers the relative error of the two-grid solution w.r.t. a global fine one-grid solution. Four two-grid solutions, see Table 6.1, all with  $H = 1$  and  $h = 0.5$ , are compared with the one-grid displacement solution  $\mathbf{U}^{ref}$  with  $H = 0.5$  for the coarse domain  $\Omega_c = [0, 30] \times [0, 10] \times [0, 1]$ . In all four examples, only one time step is computed, elastoplastic material behavior is chosen, the geometry of the tool and the indentation depth are the same as in Section 6.4.2. The relative error

$$\frac{\|\mathbf{U}^i - \mathbf{U}^{ref}\|_\infty}{\|\mathbf{U}^{ref}\|_\infty}, \quad i = 1, \dots, 4$$

is shown in Figure 6.11. The expected behavior is that the relative error of a coarse grid solution (0 nodes in the fine mesh) w.r.t the global fine solution can be already



tool path for Section 6.4.5

Figure 6.10: Contact surface of the fine and the coarse FE grid for two-grid solution 2, see Table 6.1.

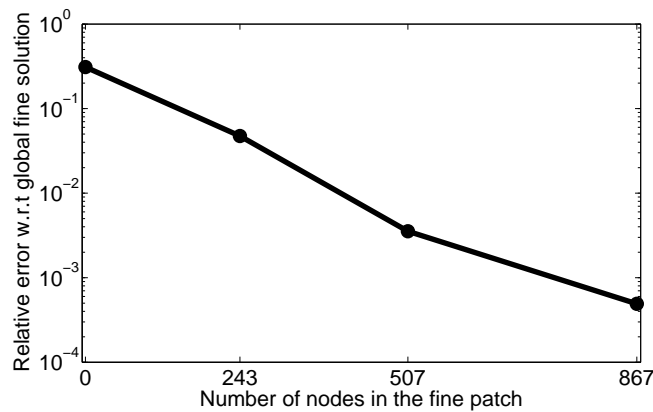


Figure 6.11: Relative error w.r.t. the global fine solution.

significantly reduced, using a fine grid with a moderate number of nodes. In this example, one can see that it is not necessary to choose much more than 500 fine nodes to achieve a relative error of less than one percent. This number of fine nodes has to be added to the 682 nodes in the coarse grid, whereas for the global fine solution 3843 nodes were needed. So the conclusion can be drawn that due to the small size of the forming zone and the very localized loading, a small fine patch already leads to an acceptable accuracy. Of course, the size of the local patch also depends strongly on the size of the tool, the indentation depth and other process parameters. To preserve the black box character of a given FE package, an adaptivity should be implemented such that the size of the small patch adapts not only to the location of the tool but also to the size of the plastic zone  $\Omega_{p,n}$ , defined in Equation (2.13). Due to the incremental character of the forming process, it can be expected that a drastic change does not happen very often.

Table 6.1: Various domain decompositions.

number	fine domain $\Omega_f$	number of fine nodes
1	$\emptyset$	0
2	$[3, 7] \times [3, 7] \times [0, 1]$	243, see Figure 6.10
3	$[2, 8] \times [2, 8] \times [0, 1]$	507, rel. error < 1%
4	$[1, 9] \times [1, 9] \times [0, 1]$	867, rel. error < 0.1%

### 6.4.5 Update strategies

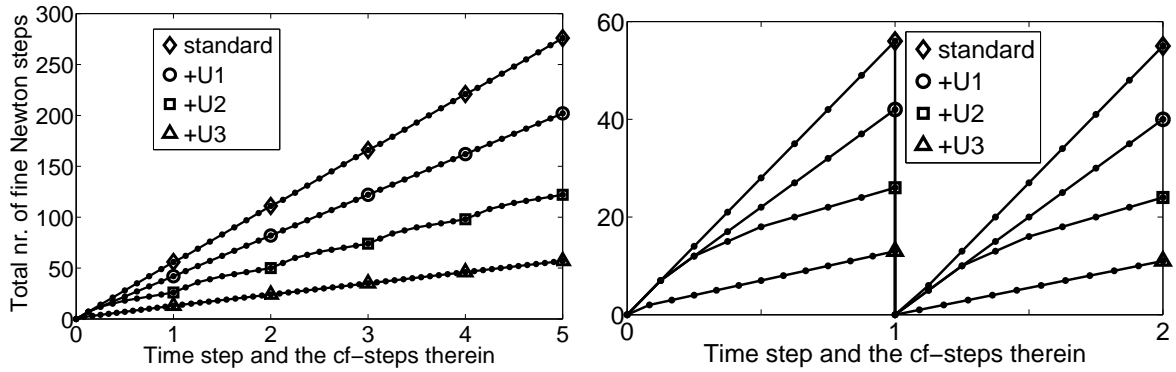


Figure 6.12: Left: total number of fine Newton steps, right: number of fine Newton steps within time step 1 and 2.

A good performance of Algorithm 3 can be achieved if a careful update strategy is employed. In order to demonstrate this, the following numerical example is presented. The coarse domain is  $\Omega_c = [0, 30] \times [0, 10] \times [0, 1]$  and the starting fine domain is  $\Omega_{f,0} = [3, 7] \times [3, 7] \times [0, 1]$ . The mesh sizes are again  $H = 1$  and  $h = 0.5$ . Elastoplastic material behavior is chosen and the geometry of the tool and the indentation are the same as in Section 6.4.2. The tool, once indenting, is following a straight line parallel to the boundary of  $\Omega_c$ , defined by  $\mathbf{t}_n = [1 \ 0 \ 0]$ , see Figure 6.10. In Figure 6.12, four different computations are compared. For a rather stationary problem as depicted in Figure 6.10, it is sufficient to observe the first 5 time steps. The total number of fine Newton iterations is plotted versus the time steps and the coarse-fine steps therein. The symbol  $\bullet$  marks, where a coarse-fine step is completed.

1. A standard computation ( $\diamond$ ) is done as in Algorithm 2 without any update strategy.
2. In the second computation ( $\circ$ ), the update strategy (U1) was used. Not surprisingly, the convergence is significantly faster if the information about the active nodes of the last coarse-fine iteration is used in the current step.



3. In the computation ( $\square$ ), the update strategies (U1) *and* (U2) were used. One can see that in each coarse-fine iteration the number of fine Newton steps until convergence decreases since increasingly better Newton starting iterates are used.
4. In the computation ( $\triangle$ ), all three update strategies (U1)-(U3) are used, so in each coarse-fine iteration only one fine Newton step is made.

The conclusion is that a significant saving of fine Newton iterations and computation time can be achieved. Actually, in the version ( $\triangle$ ) the number of fine Newton iterations turns out to be not much higher than the total number of Newton iterations that a one-grid computation would take. In addition, one has to keep in mind that the fine linear systems to be solved are much smaller than the coarse ones, especially in examples of an industrial scale where the forming zone is very small in comparison to the workpiece. But it depends strongly on the efficiency of the coarse grid linear solver - this is the reason why the total number of fine Newton steps is chosen as an indicator for the computational cost and not the total computation time - whether strategy ( $\square$ ) or ( $\triangle$ ) can be recommended as the fastest one, since the number of coarse grid computations is smaller in strategy ( $\square$ ). It should be pointed out that using the update strategies, no accuracy is lost. Algorithm 2 without and Algorithm 2 with the update strategies (U1)–(U3) lead to the same results.

### 6.4.6 Deep-rolling like example

The final numerical example is a two-grid example with contact and with elastoplastic coupling of a  $(30 \times 30 \times 1)$  coarse domain, discretized with  $H = 1$  and a  $(6 \times 6 \times 1)$  fine domain, discretized with  $h = 0.5$ . The sphere headed rigid tool is forming a small groove into the surface. It has the same geometry and indention as in Section 6.4.2. In Figure 6.13, the real displacements  $\mathbf{U}_H$  on  $\Gamma_c$  and  $\mathbf{P}\mathbf{U}_H + \mathbf{U}_h$  on  $\Gamma_{f,n}$  are shown at the time steps  $n = 181, 185, 194$  and  $198$ . In Figure 6.14, time step 30 is shown. There, the tool path and the (r)emaining displacements (without load)  $\mathbf{u}_r$  are shown, which are defined as the solution of the problem: Find  $\mathbf{u}_r \in [H_0^1(\Omega_c)]^3$ , such that

$$\int_{\Omega_c} \boldsymbol{\varepsilon}(\mathbf{u}_r) : \mathbf{C}^{el} : \boldsymbol{\varepsilon}(\mathbf{v}) = \int_{\Omega_c} \boldsymbol{\varepsilon}^p : \mathbf{C}^{el} : \boldsymbol{\varepsilon}(\mathbf{v}), \quad \mathbf{v} \in [H_0^1(\Omega_c)]^3 \quad (6.17)$$

The remaining displacements are computed and plotted on the storage grid. It is important to note that this large elastic system has to be solved only for visualization purposes and is not needed for the current computation. One can see in Figure 6.14 that the remaining plastic deformation is nearly as accurately resolved as a global fine one-grid computation would resolve it, although only a coarse grid was used for the global problem.

## 6.5 Conclusions and future work

The method, proposed in this chapter, can be characterized with the following points:

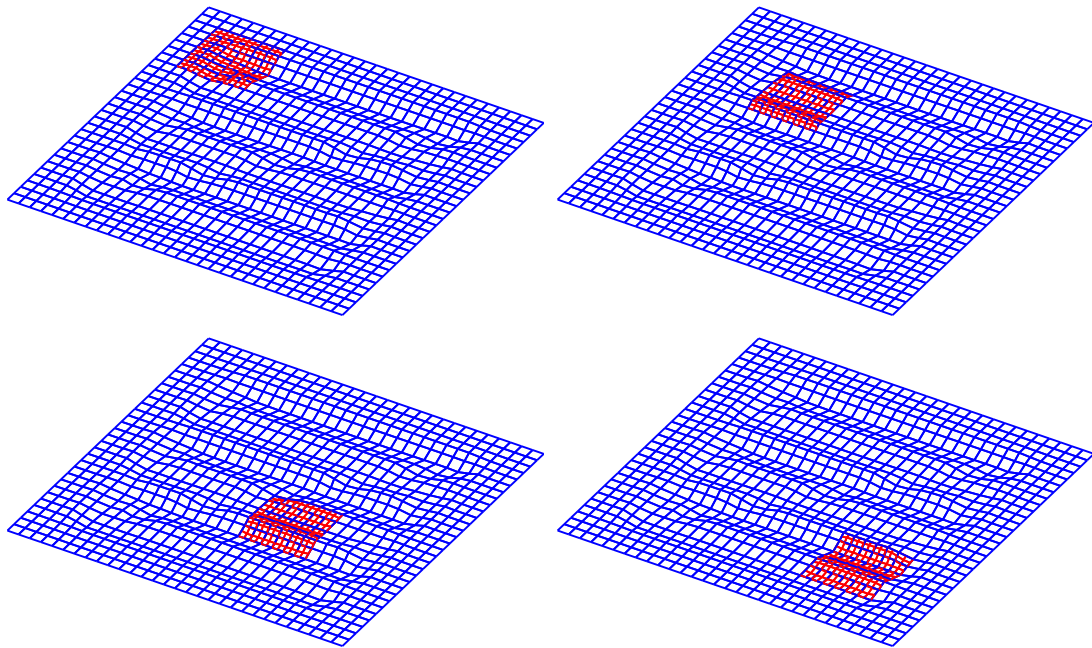


Figure 6.13: Several time steps at the end of the process, displacements of fine and coarse grid (scaling factor = 200), fine grid moving without load over the coarse grid.

- First steps were taken to accelerate the simulation of metal forming processes with the aid of a coupled elastoplastic algorithm.
- The number of dof and the computational cost can be reduced by the use of an ODDM with a two-scale approach without the need for sophisticated remeshing algorithms.
- Mortar operators are in some sense optimal transfer operators for stresses. They are easy to implement and easy to test. A patch test setting, independent of the underlying structural FE problem was proposed.
- The two grids are in principle independent and the fine patch is not only a local refinement of the coarse mesh which is still a slight advantage over classical superposition methods.
- The ODDM is a dynamic one, in the sense that the local grid is moving with the ‘hot spot’.
- A nonlinear contact problem can be handled by the coupling algorithm.
- The nonlinearities of contact and plasticity enter the coarse computation only as load correction terms on the right hand side.

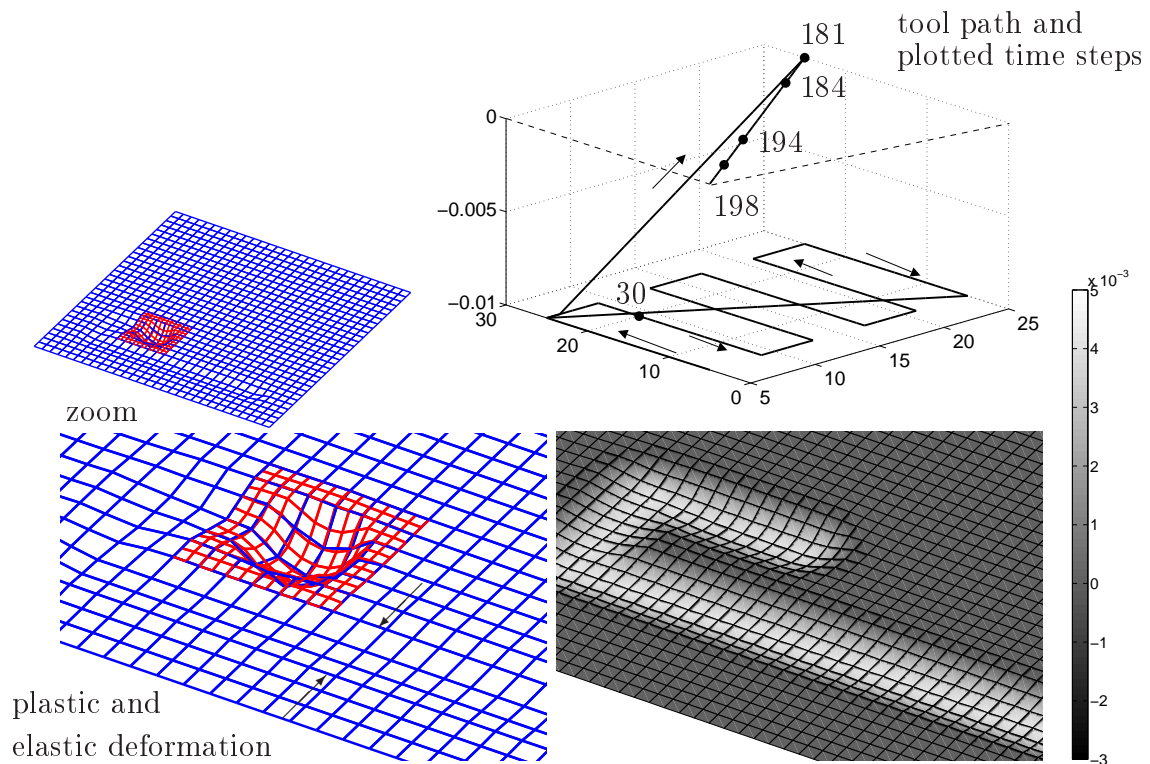


Figure 6.14: Time step 30, remaining displacements and  $\alpha$ , scaling factor = 300 (right lower picture), coarse and fine grid, scaling factor = 200 (left pictures), tool path (right upper picture).

- In contrast to other approaches, mentioned in Section 6.2, more knowledge about the nonlinearities is utilized in the iterations which leads to an efficient algorithm.

One point that is completely left untouched by this work is the question how the mapping of the plastic history data is done if coarse, fine and storage mesh are completely non-conforming and non-hierarchical, see Figure 6.5(c). This is a non-trivial task for which the reader is referred to the literature, since this problem has been addressed by many researchers who have worked on remeshing in order to guarantee a persistently good mesh quality in simulations with very large plastic deformation [24, 47, 79, 103]. Of course, this short list is by no means complete. In Chapter 8, an outlook is given how this work can be extended to the regime of large deformations. Also a comparison with adaptive methods with hanging nodes [85] would be strongly desirable.



# 7 An inexact Newton block scheme

The aim of this chapter is to review some algorithmic aspects of the coupling method in Section 6. For simplicity, the notation is still restricted to the regime of small deformations.

## 7.1 Motivation

The structure of Algorithm 3 was basically this one:

coarse-fine loop		
	coarse linear system	solver loop
	fine Newton	

In this section, also this structure is investigated:

Newton		
	coarse tangential system	solver loop
	fine tangential system	

It turns out that, organized in this manner, the numerical scheme can be put much easier into the perspective of more classical inexact Newton schemes. This is expected to make the algorithm more reliable and easier to globalize in the sense that damping schemes from the literature can be employed to guarantee global convergence. The questions which arise are:

- Is this method still efficient?
- Is the algorithmic interface between coarse and fine still small?
- Is it even possible to omit the coarse-fine iteration?

## 7.2 Block iterative scheme: Linear elasticity

The following discrete FE spaces are introduced:

$$\mathbb{V}_h \subset [H^1(\Omega_f)]^d, \quad \mathbb{V}_H \subset [H^1(\Omega_c)]^d, \quad \mathbb{V}_{0h} \subset [H_0^1(\Omega_f)]^d, \quad \mathbb{V}_{0H} \subset [H_0^1(\Omega_c)]^d$$

The subject of publications like for instance [45] is to solve multiscale elliptic problems on a locally enriched space. Written in weak form: Find  $\mathbf{u} \in \mathbb{V}_{0h} \oplus \mathbb{V}_{0H}$  such that

$$a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \mathbf{f}^{ext} \cdot \mathbf{v}, \quad \mathbf{v} \in \mathbb{V}_{0h} \oplus \mathbb{V}_{0H}$$

In [45], the bilinear form  $a(\cdot, \cdot)$  comes from the weak formulation of the Laplace equation, whereas here it stems from the weak formulation of the elasticity equation. After discretization one arrives at

$$\begin{bmatrix} \hat{\mathbf{A}}_h & \hat{\mathbf{A}}_{hH} \\ \hat{\mathbf{A}}_{Hh} & \hat{\mathbf{A}}_H \end{bmatrix} \begin{bmatrix} \hat{\mathbf{U}}_h \\ \hat{\mathbf{U}}_H \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{F}}_h^{ext} \\ \hat{\mathbf{F}}_H^{ext} \end{bmatrix} \quad (7.1)$$

with  $\hat{\mathbf{A}}_{hH}$  the FE assembly of  $a(\mathbf{u}, \mathbf{v})$  in  $\mathbb{V}_H \times \mathbb{V}_h$  and  $\hat{\mathbf{A}}_{Hh}$  the FE assembly of  $a(\mathbf{u}, \mathbf{v})$  in  $\mathbb{V}_h \times \mathbb{V}_H$ . With the hat symbol is meant that the (zero) Dirichlet boundary conditions are not yet included in this system, so, for example

$$\hat{\mathbf{A}}_h = \begin{bmatrix} \mathbf{A}_h & \mathbf{A}_{h, \mathbb{N}_f \mathbb{D}_f} \\ \mathbf{A}_{h, \mathbb{D}_f \mathbb{N}_f} & \mathbf{A}_{h, \mathbb{D}_f \mathbb{D}_f} \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{U}}_h = \begin{bmatrix} \mathbf{U}_h \\ \mathbf{0} \end{bmatrix}$$

with  $\mathbb{N}_f$  and  $\mathbb{D}_f$  denoting the free structure dof and the Dirichlet dof of the fine system with  $\mathbb{N}_f \cap \mathbb{D}_f = \emptyset$ . After the removal of Dirichlet rows and columns, the following reduced system is obtained:

$$\begin{bmatrix} \mathbf{A}_h & \mathbf{A}_{hH, \mathbb{N}_f \mathbb{N}_c} \\ \mathbf{A}_{Hh, \mathbb{N}_c \mathbb{N}_f} & \mathbf{A}_H \end{bmatrix} \begin{bmatrix} \mathbf{U}_h \\ \mathbf{U}_H \end{bmatrix} = \begin{bmatrix} \mathbf{F}_h^{ext} \\ \mathbf{F}_H^{ext} \end{bmatrix} \quad (7.2)$$

In [45], the mixed matrices  $\mathbf{A}_{hH}$  and  $\mathbf{A}_{Hh}$  are set up correctly, which involves computing the intersections of elements of the discretization of  $\mathbb{V}_h$  and  $\mathbb{V}_H$ . After that, a Xu type successive subspace correction method is employed to solve this  $2 \times 2$  block system iteratively. This way is not pursued entirely in this work:

1. First of all, the coupling  $\mathbf{A}_{Hh}$  is neglected completely, since it is assumed that the fine grid correction does not substantially influence the coarse grid solution apart from the contact stresses and the plastic deformation history. A block iterative solution of (7.2) is then not necessary any more, because of the block triangular structure which saves a lot of computation time. Another reason for these simplifications is to avoid algorithmic difficulties arising from the necessity to guarantee that the sum of  $\mathbb{V}_h(t_n)$  and  $\mathbb{V}_H$  is a direct sum at any time  $t_n$ , which is a necessary requirement that the matrix in (7.2)<sub>lhs</sub> is invertible. Due to the movement of the tool, the space  $\mathbb{V}_h(t_n)$  depends on the position of the tool and therefore it can happen very easily that the sum of the spaces is not direct any more. Therefore, the assumption of a direct sum is omitted.
2. Secondly, the coupling  $\hat{\mathbf{A}}_{hH}$  is approximated by  $\hat{\mathbf{A}}_h \mathbf{P}$  with  $\mathbf{P}$  from Equation (6.16), so the computation of element intersections is not necessary any more, which is in particular in 3D advantageous, when it comes to the use of general hexahedral elements.

With this and the auxiliary computation

$$\mathbf{A}_{hH, \mathbb{N}_f \mathbb{N}_c} \mathbf{U}_H = [\hat{\mathbf{A}}_h \mathbf{P}]_{\mathbb{N}_f \mathbb{N}_c} \mathbf{U}_H = [\hat{\mathbf{A}}_h \mathbf{P} \mathbf{U}_H]_{\mathbb{N}_f} = \mathbf{A}_h [\mathbf{P} \mathbf{U}_H]_{\mathbb{N}_f} + \mathbf{A}_{h, \mathbb{N}_f \mathbb{D}_f} [\mathbf{P} \mathbf{U}_H]_{\mathbb{D}_f} \quad (7.3)$$

(here it is assumed that  $\mathbf{P}$  maps only the free coarse structure dof in  $\mathbb{N}_c$  and not  $\mathbb{D}_c$  to the fine grid, so  $\mathbf{P} \in \mathbb{R}^{\mathbb{N}_f + \mathbb{D}_f \times \mathbb{N}_c}$ ), one ends up with the block iterative scheme

$$\begin{bmatrix} \mathbf{A}_h & [\hat{\mathbf{A}}_h \mathbf{P}]_{\mathbb{N}_f \mathbb{N}_c} \\ \mathbf{0} & \mathbf{A}_H \end{bmatrix} \begin{bmatrix} \mathbf{U}_h \\ \mathbf{U}_H \end{bmatrix} = \begin{bmatrix} \mathbf{F}_h^{ext} \\ \mathbf{F}_H^{ext} \end{bmatrix} \quad (7.4)$$

Thus, the focus of this work is not a very high resolution of the underlying elliptic problem, but an efficient formulation which is able to hide the difficulties of contact and plasticity from the global domain. This will be explained in more detail in the following sections, where System (7.4) is extended by two block rows and block columns to account for contact and material nonlinearity.

## 7.3 Newton Block iterative scheme: Elasticity with contact

Due to the reasons illustrated in Section 7.2 it is not attempted here, to start from a weak form over the locally enriched space, although this would be desirable for analytical reasons. The starting point here is rather a slightly simplified, full, discrete, nonlinear problem:

$$\mathbf{F}(\mathbf{U}_h, \boldsymbol{\Lambda}_h, \mathbf{U}_H) := \begin{bmatrix} \mathbf{F}_h \\ \mathbf{F}_H \end{bmatrix} := \begin{bmatrix} \mathbf{A}_h \mathbf{U}_h + [\hat{\mathbf{A}}_h \mathbf{P} \mathbf{U}_H]_{\mathbb{N}_f} + \mathbf{B} \boldsymbol{\Lambda}_h \\ \mathbf{F}_h^{con}(\mathbf{U}_h + \mathbf{P} \mathbf{U}_H, \boldsymbol{\Lambda}_h) \\ \mathbf{A}_H \mathbf{U}_H + \mathbf{M}^\top \boldsymbol{\Lambda}_h \end{bmatrix} \stackrel{!}{=} \mathbf{0}$$

Herein,  $\mathbf{A}_h$  and  $\mathbf{A}_H$  are the elastic stiffnesses of  $\mathbb{T}_h$  and  $\mathbb{T}_H$ . Meaning of the three lines of the system:

1. Fine equilibrium with the fine solution as correction of the coarse solution according to the first block row of (7.4) and with the contact forces.
2. Nonlinear complementarity (NCP) function.
3. Coarse equilibrium according to the second block row of (7.4) with the load correction due to contact.

Now, following Equations (3.16), (3.21) and (3.22), this nonlinear system can be consistently linearized: With

$$\mathbf{K} := \frac{\partial \mathbf{F}}{\partial (\mathbf{U}_h, \boldsymbol{\Lambda}_h, \mathbf{U}_H)}$$

the Jacobi matrix

$$\mathbf{K} = \left[ \begin{array}{c|c} \mathbf{K}_{hh} & \mathbf{K}_{hH} \\ \mathbf{K}_{Hh} & \mathbf{K}_{HH} \end{array} \right] = \left[ \begin{array}{cc|c} \mathbf{A}_h & \mathbf{B} & [\hat{\mathbf{A}}_h \mathbf{P}]_{\mathbb{N}_f \bullet} \\ \mathbf{N}_h & \mathbf{T}_h & \mathbf{N}_h \mathbf{P}_{\mathbb{N}_f \bullet} \\ \mathbf{0} & \mathbf{M}^\top & \mathbf{A}_H \end{array} \right] \quad (7.5)$$

is defined with the abbreviations:

$$\begin{aligned} \mathbf{N}_h &:= \frac{\partial \mathbf{F}_h^{con}}{\partial \mathbf{U}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{N}_\mathbb{A} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \text{using the ordering } \mathbf{U} = \begin{bmatrix} \mathbf{U}_\mathbb{N} \\ \mathbf{U}_\mathbb{A} \\ \mathbf{U}_\mathbb{I} \end{bmatrix} \\ \mathbf{T}_h &:= \frac{\partial \mathbf{F}_h^{con}}{\partial \boldsymbol{\Lambda}_h} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_\mathbb{A} \end{bmatrix}, \quad \text{using the ordering } \boldsymbol{\Lambda}_h = \begin{bmatrix} \boldsymbol{\Lambda}_\mathbb{I} \\ \boldsymbol{\Lambda}_\mathbb{A} \end{bmatrix} \\ \mathbf{U} &:= \mathbf{U}(\mathbf{U}_h, \mathbf{U}_H) := \mathbf{U}_h + \mathbf{P}\mathbf{U}_H \end{aligned}$$

Finally, in every Newton step  $j$ , the  $2 \times 2$  block system

$$\begin{bmatrix} \mathbf{K}_{hh} & \mathbf{K}_{hH} \\ \mathbf{K}_{Hh} & \mathbf{K}_{HH} \end{bmatrix}^{(j-1)} \begin{bmatrix} \Delta \mathbf{X}_h \\ \Delta \mathbf{X}_H \end{bmatrix}^{(j)} = - \begin{bmatrix} \mathbf{F}_h \\ \mathbf{F}_H \end{bmatrix}^{(j-1)} \quad (7.6)$$

has to be solved. Here, the following definitions were made:

$$\Delta \mathbf{X}_h^{(j)} := \left[ \Delta \mathbf{U}_{h,\mathbb{N}}^{(j)}, \Delta \mathbf{U}_{h,\mathbb{I}_{j-1}}^{(j)}, \Delta \mathbf{U}_{h,\mathbb{A}_{j-1}}^{(j)}, \Delta \boldsymbol{\Lambda}_{h,\mathbb{I}_{j-1}}^{(j)}, \Delta \boldsymbol{\Lambda}_{h,\mathbb{A}_{j-1}}^{(j)} \right]^\top, \quad \Delta \mathbf{X}_H^{(j)} := \Delta \mathbf{U}_H^{(j)}$$

Thus, the domain decomposition into coarse and fine defines in a very natural way the block decomposition for the block iterative scheme, like described in [96] and the references therein.

## 7.4 Newton Block iterative scheme: Plasticity with contact

In this section, also the material nonlinearity is taken into account. The idea is to introduce an additional unknown  $\mathbf{u}_f^p \in [H_0^1(\Omega_f)]^d$  as the Galerkin projection of the plastic strain increment with respect to the elasticity bilinear form  $a(\cdot, \cdot)$  (see also formula (6.17))

$$\int_{\Omega_f} \boldsymbol{\varepsilon}(\mathbf{u}_f^p) : \mathbf{C}^{el} : \boldsymbol{\varepsilon}(\mathbf{v}) = \int_{\Omega_f} \Delta \boldsymbol{\varepsilon}_{f,n+1}^p : \mathbf{C}^{el} : \boldsymbol{\varepsilon}(\mathbf{v}), \quad \mathbf{v} \in \mathbb{V}_{0h}$$

and to define  $\mathbf{u}_f^{tr}$  such that  $\mathbf{u}_f$  can be split up additively into  $\mathbf{u}_f = \mathbf{u}_f^{tr} - \mathbf{u}_f^p$ . This is a purely fictitious decomposition. In particular  $\nabla^{sym} \mathbf{u}_f^p = \boldsymbol{\varepsilon}_{f,n+1}^p - \boldsymbol{\varepsilon}_{f,n}^p$  does not hold. Another problem is that the plastic dislocations which are captured by the incompatible [72] strain field  $\boldsymbol{\varepsilon}_f^p$ , are now smoothed out by this projection. In particular, also the eigen strains are contained in  $\mathbf{u}_f^p$ . This is the reason why finally the  $\mathbf{u}_f^p$  will not be stored on the storage grid but the plastic strain  $\boldsymbol{\varepsilon}_f^p$ . With this, the overall nonlinear problem is:

$$\begin{aligned} \mathbf{F}(\mathbf{X}) &:= \mathbf{F}(\mathbf{X}_h, \mathbf{X}_H) := \mathbf{F}(\mathbf{U}_h^{tr}, \boldsymbol{\Lambda}_h, \mathbf{U}_h^p, \mathbf{U}_H) \\ &:= \begin{bmatrix} \mathbf{F}_h(\mathbf{U}_h^{tr}, \boldsymbol{\Lambda}_h, \mathbf{U}_h^p, \mathbf{U}_H) \\ \mathbf{F}_H(\boldsymbol{\Lambda}_h, \mathbf{U}_h^p, \mathbf{U}_H) \end{bmatrix} \\ &:= \begin{bmatrix} \mathbf{A}_h(\mathbf{U}_h^{tr} - \mathbf{U}_h^p) + [\hat{\mathbf{A}}_h \mathbf{P}\mathbf{U}_H]_{\mathbb{N}_f} + \mathbf{B}\boldsymbol{\Lambda}_h + \mathbf{F}_{h,n}^{pl} \\ \mathbf{F}_h^{con}(\mathbf{U}_h^{tr} + \mathbf{P}\mathbf{U}_H, \boldsymbol{\Lambda}_h) \\ \mathbf{F}_h^{int}(\mathbf{U}_h^{tr} + \mathbf{P}\mathbf{U}_H) - \mathbf{A}_h(\mathbf{U}_h^{tr} - \mathbf{U}_h^p) - [\hat{\mathbf{A}}_h \mathbf{P}\mathbf{U}_H]_{\mathbb{N}_f} - \mathbf{F}_{h,n}^{pl} \\ \mathbf{A}_H \mathbf{U}_H - \mathbf{P}^\top \hat{\mathbf{A}}_{h,\bullet \mathbb{N}_f} \mathbf{U}_h^p + \mathbf{M}^\top \boldsymbol{\Lambda}_h + \mathbf{F}_{H,n}^{pl} \end{bmatrix} \stackrel{!}{=} \mathbf{0} \quad (7.7) \end{aligned}$$



with  $\mathbf{F}_h^{int}$  the standard finite element assembly of

$$\begin{aligned} & \int_{\Omega_f} \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}), \quad \text{relation } \boldsymbol{\sigma}(\mathbf{u}) \text{ fully nonlinear} \\ &= \int_{\Omega_f} (\boldsymbol{\varepsilon}(\mathbf{u}_f^{tr} + \mathbf{u}_c) - \boldsymbol{\varepsilon}_{f,n}^p - \Delta \boldsymbol{\varepsilon}_{f,n+1}^p) : \mathbf{C}^{el} : \boldsymbol{\varepsilon}(\mathbf{v}) \end{aligned}$$

and with  $\mathbf{F}_{h,n}^{pl}$  the standard finite element assembly of

$$l_f^{pl} := - \int_{\Omega_f} \boldsymbol{\varepsilon}_{f,n}^p : \mathbf{C}^{el} : \boldsymbol{\varepsilon}(\mathbf{v})$$

The latter term is included in the first and in the third block row of system (7.7) to account for the plastic deformation history until time step  $t_n$ . The meaning of the four lines of system (7.7) is:

1. Fine equilibrium with the fine solution as correction of the coarse solution and with the contact forces. Under the assumption that  $\mathbf{u}_f^p$  is known, this would be an equation for the nonlinear relation  $\boldsymbol{\sigma}(\mathbf{u})$ .
2. NCP function.
3. This is again the spatial discretization of the fine equilibrium. Now under the assumption that the nonlinear relation  $\boldsymbol{\sigma}(\mathbf{u})$  is known, this would be an equation for  $\mathbf{u}_f^p$ . In this block row, the plastic constitutive law comes into play.
4. Coarse equilibrium with the load correction terms due to contact and plasticity.

With

$$\mathbf{K} := \frac{\partial \mathbf{F}}{\partial (\mathbf{U}_h^{tr}, \boldsymbol{\Lambda}_h, \mathbf{U}_h^p, \mathbf{U}_H)}$$

the Jacobi matrix is defined:

$$\mathbf{K} = \left[ \begin{array}{cc|cc} \mathbf{A}_h & \mathbf{B} & -\mathbf{A}_h & [\hat{\mathbf{A}}_h \mathbf{P}]_{\mathbb{N}_f \bullet} \\ \mathbf{N}_h & \mathbf{T}_h & \mathbf{0} & \mathbf{N}_h \mathbf{P}_{\mathbb{N}_f \bullet} \\ \mathbf{K}_h^p & \mathbf{0} & \mathbf{A}_h & [\hat{\mathbf{K}}_h^p \mathbf{P}]_{\mathbb{N}_f \bullet} \\ \hline \mathbf{0} & \mathbf{M}^\top & -\mathbf{P}^\top \hat{\mathbf{A}}_{h, \bullet \mathbb{N}_f} & \mathbf{A}_H \end{array} \right] \quad (7.8)$$

Herein,  $\mathbf{K}^p$  is defined as the FE assembly of

$$- \int_{\Omega_f} \boldsymbol{\varepsilon}(\mathbf{u}) : \mathbf{C}^p : \boldsymbol{\varepsilon}(\mathbf{v})$$

with the fourth order tensor  $\mathbf{C}^p$  defined in Equation (2.19).

*Remark:* If one cancels the third row and the third column of (7.8) (problem without plasticity) one arrives again at (7.5).

## 7.5 Block iterative schemes

For the solution of problem (7.6), several block iterative schemes are applicable. Two of them are given here:

**Block-Jacobi:** Fix  $j$ , for  $k = 1, \dots, K$  solve

$$\mathbf{K}_{hh}^{(j-1)} \Delta \mathbf{X}_h^{(j,k)} = -\mathbf{F}_h^{(j-1)} - \mathbf{K}_{hH}^{(j-1)} \Delta \mathbf{X}_H^{(j,k-1)} \quad (7.9)$$

$$\mathbf{K}_{HH}^{(j-1)} \Delta \mathbf{X}_H^{(j,k)} = -\mathbf{F}_H^{(j-1)} - \mathbf{K}_{Hh}^{(j-1)} \Delta \mathbf{X}_h^{(j,k-1)} \quad (7.10)$$

In principle it is possible to parallelize (7.9), (7.10). But also the solving of the coarse linear system can be parallelized on its part.

$K$	case
some value $K_j$	inexact Newton, see [37] and Section 7.7
1	Newton = coarse-fine iteration, included in Algorithm 3 when update strategy (U3) is used
not fixed	iteration stops when convergence criterion is met

**Block Gauss Seidel:** Fix  $j$ , for  $k = 1, \dots, K$  solve

$$\mathbf{K}_{hh}^{(j-1)} \Delta \mathbf{X}_h^{(j,k)} = -\mathbf{F}_h^{(j-1)} - \mathbf{K}_{hH}^{(j-1)} \Delta \mathbf{X}_H^{(j,k-1)} \quad (7.11)$$

$$\mathbf{K}_{HH}^{(j-1)} \Delta \mathbf{X}_H^{(j,k)} = -\mathbf{F}_H^{(j-1)} - \mathbf{K}_{Hh}^{(j-1)} \Delta \mathbf{X}_h^{(j,k)} \quad (7.12)$$

As before, (7.11) can be solved in the condensed form (3.24).

## 7.6 Schur complement method

Having a closer look at the Equations (7.11) and (7.12), one recognizes that it is possible to make the coarse-fine iteration obsolete at the expense that the application of  $\mathbf{A}_h^{-1}$  on the matrix  $\mathbf{B}$  has to be computed. Furthermore, good convergence of the Newton iteration cannot be achieved, as it will be shown.

### 7.6.1 Schur complement method, elastic case

Equation (7.11), written explicitly is

$$\begin{bmatrix} \mathbf{A}_h & \mathbf{B}^{(j-1)} \\ \mathbf{N}_h^{(j-1)} & \mathbf{T}_h^{(j-1)} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{U}_h \\ \Delta \Lambda_h \end{bmatrix}^{(j,k)} = - \begin{bmatrix} \mathbf{F}_{h,1} \\ \mathbf{F}_{h,2} \end{bmatrix}^{(j-1)} - \begin{bmatrix} [\hat{\mathbf{A}}_h \mathbf{P}]_{\mathbb{N}_f \bullet} \\ \mathbf{N}_h \mathbf{P}_{\mathbb{N}_f \bullet} \end{bmatrix}^{(j-1)} \Delta \mathbf{U}_H^{(j,k-1)} \quad (7.13)$$

Using <sup>1</sup> a Schur complement formula from standard textbooks like e.g. [96], the Newton increment of the contact stresses can be computed using Equation (7.3):

$$\begin{aligned}
 & (\mathbf{T}_h - \mathbf{N}_h \mathbf{A}_h^{-1} \mathbf{B})^{(j-1)} \Delta \mathbf{\Lambda}_h^{(j,k)} \\
 = & -\mathbf{F}_{h,2}^{(j-1)} - \mathbf{N}_h^{(j-1)} \mathbf{P}_{\mathbb{N}_f} \Delta \mathbf{U}_H^{(j,k-1)} + \mathbf{N}_h^{(j-1)} \mathbf{A}_h^{-1} \mathbf{F}_{h,1}^{(j-1)} + \mathbf{N}_h^{(j-1)} \mathbf{A}_h^{-1} [\hat{\mathbf{A}}_h \mathbf{P} \Delta \mathbf{U}_H^{(j,k-1)}]_{\mathbb{N}_f} \\
 = & -\mathbf{F}_{h,2}^{(j-1)} + \mathbf{N}_h^{(j-1)} \mathbf{A}_h^{-1} \mathbf{F}_{h,1}^{(j-1)} + \mathbf{F}_{h,\mathbb{D}_f}^{(j,k-1)} \tag{7.14}
 \end{aligned}$$

with the abbreviation

$$\mathbf{F}_{h,\mathbb{D}_f}^{(j,k-1)} := \mathbf{N}_h^{(j-1)} \mathbf{A}_h^{-1} \mathbf{A}_{h,\mathbb{N}_f \mathbb{D}_f} [\mathbf{P} \Delta \mathbf{U}_H^{(j,k-1)}]_{\mathbb{D}_f}$$

The matrix

$$\mathbf{T}_h - \mathbf{N}_h \mathbf{A}_h^{-1} \mathbf{B} \tag{7.15}$$

is nonsingular as long as the matrix in Equation (7.13)<sub>lhs</sub> is nonsingular, see page 457 of [96]. Equation (7.14) reveals that, except from

$$\mathbf{F}_{h,\mathbb{D}_f}^{(j,k-1)}$$

all parts of the right hand side, containing  $\Delta \mathbf{U}_H$  can be eliminated. Neglecting this term, the right hand does not depend on  $k$  any more which means that approximations of  $\Delta \mathbf{\Lambda}_h$ ,  $\Delta \mathbf{U}_H$  and  $\Delta \mathbf{U}_h$  can be computed directly in this order:

$$(\mathbf{T}_h - \mathbf{N}_h \mathbf{A}_h^{-1} \mathbf{B})^{(j-1)} \Delta \tilde{\mathbf{\Lambda}}_h^{(j)} = -\mathbf{F}_{h,2}^{(j-1)} + \mathbf{N}_h^{(j-1)} \mathbf{A}_h^{-1} \mathbf{F}_{h,1}^{(j-1)} \tag{7.16}$$

$$\mathbf{A}_H \Delta \tilde{\mathbf{U}}_H^{(j)} = -\mathbf{A}_H \tilde{\mathbf{U}}_H^{(j-1)} - \mathbf{M}^\top \tilde{\mathbf{\Lambda}}_h^{(j-1)} - \mathbf{M}^\top \Delta \tilde{\mathbf{\Lambda}}_h^{(j)} \tag{7.17}$$

$$\begin{aligned}
 \mathbf{A}_h \Delta \tilde{\mathbf{U}}_h^{(j)} &= -\mathbf{A}_h (\tilde{\mathbf{U}}_h + \mathbf{P} \tilde{\mathbf{U}}_H)^{(j-1)} \\
 &\quad - \mathbf{B} (\tilde{\mathbf{\Lambda}}_h^{(j-1)} + \Delta \tilde{\mathbf{\Lambda}}_h^{(j)}) - \mathbf{A}_h \mathbf{P} \Delta \tilde{\mathbf{U}}_H^{(j)} \tag{7.18}
 \end{aligned}$$

For the Equations (7.17) and (7.18), the formulas (7.11) and (7.12) were just resolved after the desired variables. This can be simplified to:

$$(\mathbf{T}_h - \mathbf{N}_h \mathbf{A}_h^{-1} \mathbf{B})^{(j-1)} \Delta \tilde{\mathbf{\Lambda}}_h^{(j)} = -\mathbf{F}_{h,2}^{(j-1)} + \mathbf{N}_h^{(j-1)} \mathbf{A}_h^{-1} \mathbf{F}_{h,1}^{(j-1)} \tag{7.19}$$

$$\mathbf{A}_H \tilde{\mathbf{U}}_H^{(j)} = -\mathbf{M}^\top \tilde{\mathbf{\Lambda}}_h^{(j)} \tag{7.20}$$

$$\mathbf{A}_h \Delta \tilde{\mathbf{U}}_h^{(j)} = -\mathbf{A}_h (\tilde{\mathbf{U}}_h^{(j-1)} + \mathbf{P} \tilde{\mathbf{U}}_H^{(j)}) - \mathbf{B} \tilde{\mathbf{\Lambda}}_h^{(j)} \tag{7.21}$$

Neglecting the boundary term  $\mathbf{F}_{h,\mathbb{D}_f}^{(j,k-1)}$  will of course deteriorate the convergence of the Newton iteration significantly, see Figure 7.2 in Section 7.9.3. Nevertheless, this method can be used to compute a very good starting iterate for the inexact Newton block Gauss Seidel (GS) method, as described in Section 7.7.

<sup>1</sup>Idea from Dipl. Math. Corinna Hager, IANS, Stuttgart

### 7.6.2 Schur complement method, plastic case

In the plastic case, Equation (7.11) is written explicitly with the aid of Equations (7.7) and (7.8),

$$\begin{bmatrix} \mathbf{A}_h & \mathbf{Z}_1 \\ \mathbf{Z}_2 & \mathbf{Z}_3 \end{bmatrix}^{(j-1)} \begin{bmatrix} \Delta \mathbf{U}_h^{tr} \\ \Delta \boldsymbol{\Xi} \end{bmatrix}^{(j,k)} = \begin{bmatrix} -\mathbf{F}_{h,1} \\ -\mathbf{F}_{h,[2;3]} \end{bmatrix}^{(j-1)} - \begin{bmatrix} [\hat{\mathbf{A}}_h \mathbf{P}]_{\mathbb{N}_f \bullet} \\ \hat{\mathbf{Z}}_2 \end{bmatrix} \Delta \mathbf{U}_H^{(j,k-1)} \quad (7.22)$$

with the abbreviations

$$\mathbf{Z}_1 := [\mathbf{B} \quad -\mathbf{A}_h], \quad \mathbf{Z}_2 := \begin{bmatrix} \mathbf{N}_h \\ \mathbf{K}_h^p \end{bmatrix}, \quad \hat{\mathbf{Z}}_2 := \begin{bmatrix} \mathbf{N}_h \mathbf{P}_{\mathbb{N}_f \bullet} \\ [\hat{\mathbf{K}}_h^p \mathbf{P}]_{\mathbb{N}_f \bullet} \end{bmatrix}, \quad \mathbf{Z}_3 := \begin{bmatrix} \mathbf{T}_h & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_h \end{bmatrix}$$

and

$$\boldsymbol{\Xi} := \begin{bmatrix} \boldsymbol{\Lambda}_h \\ \mathbf{U}_h^p \end{bmatrix}$$

Setting up the Schur complement leads to:

$$\begin{aligned} & (\mathbf{Z}_3 - \mathbf{Z}_2 \mathbf{A}_h^{-1} \mathbf{Z}_1)^{(j-1)} \Delta \boldsymbol{\Xi}^{(j,k)} \\ &= -\mathbf{F}_{h,[2;3]}^{(j-1)} + \mathbf{Z}_2 \mathbf{A}_h^{-1} \mathbf{F}_{h,1}^{(j-1)} + \mathbf{Z}_2 \mathbf{A}_h^{-1} \mathbf{A}_h \mathbf{P} \Delta \mathbf{U}_H^{(j,k-1)} - \hat{\mathbf{Z}}_2 \Delta \mathbf{U}_H^{(j,k-1)} \end{aligned}$$

In analogy to (7.16),  $\hat{\mathbf{Z}}_2$  is substituted by  $\mathbf{Z}_2 \mathbf{P}$ . Then the terms with  $\Delta \mathbf{U}_H^{(j,k-1)}$  are cancelled out again, such that the approximations  $\Delta \tilde{\boldsymbol{\Xi}}, \Delta \tilde{\mathbf{U}}_H, \Delta \tilde{\mathbf{U}}_h^{tr}$  can be computed in this order:

$$\mathbf{S}_h^{(j-1)} \Delta \tilde{\boldsymbol{\Xi}}^{(j)} = -\mathbf{F}_{h,[2;3]}^{(j-1)} + \mathbf{Z}_2 \mathbf{A}_h^{-1} \mathbf{F}_{h,1}^{(j-1)} \quad (7.23)$$

$$\mathbf{A}_H \Delta \tilde{\mathbf{U}}_H^{(j)} = \mathbf{P}^\top \hat{\mathbf{A}}_{h,\bullet \mathbb{N}_f} \Delta \tilde{\mathbf{U}}_h^{p,(j)} - \mathbf{M}^\top \Delta \tilde{\boldsymbol{\Lambda}}_h^{(j)} - \mathbf{F}_{h,4}^{(j-1)} \quad (7.24)$$

$$\mathbf{A}_h \Delta \tilde{\mathbf{U}}_h^{tr,(j)} = \mathbf{A}_h (\Delta \tilde{\mathbf{U}}_h^{p,(j)} - \mathbf{P} \Delta \tilde{\mathbf{U}}_H^{(j)}) - \mathbf{B} \Delta \tilde{\boldsymbol{\Lambda}}_h^{(j)} - \mathbf{F}_{h,1}^{(j-1)} \quad (7.25)$$

with the Schur matrix  $\mathbf{S}_h$  defined by

$$\mathbf{S}_h := \mathbf{Z}_3 - \mathbf{Z}_2 \mathbf{A}_h^{-1} \mathbf{Z}_1$$

It has the form

$$\mathbf{S}_h = \begin{bmatrix} \mathbf{T}_h & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_h \end{bmatrix} - \begin{bmatrix} \mathbf{N}_h \mathbf{A}_h^{-1} \mathbf{B} & -\mathbf{N}_h \\ \mathbf{K}_h^p \mathbf{A}_h^{-1} \mathbf{B} & -\mathbf{K}_h^p \end{bmatrix} = \begin{bmatrix} \mathbf{T}_h - \mathbf{N}_h \mathbf{A}_h^{-1} \mathbf{B} & \mathbf{N}_h \\ -\mathbf{K}_h^p \mathbf{A}_h^{-1} \mathbf{B} & \mathbf{A}_h + \mathbf{K}_h^p \end{bmatrix} \quad (7.26)$$

Not surprisingly,  $\mathbf{S}_{h,11}$  turns out to be the Schur complement defined in (7.15). One remark on efficiency: Since it is necessary to perform the application of  $\mathbf{A}_h^{-1}$  on  $\mathbf{B}$  (solve  $3|\mathbb{S}|$  linear equations), this method is only efficient if the small patch is significantly smaller than the coarse domain.

## 7.7 Inexact Newton methods

A very important issue in combining the Newton method for the nonlinear block-problem  $\mathbf{F}(\mathbf{U}) = \mathbf{0}$  with an iterative linear solver is to avoid the so-called oversolving, an effect, extensively illustrated in [37]. Oversolving can happen on two different levels:

1. Interaction between Newton loop and block Gauss Seidel iteration.
2. For the large coarse part (7.12), the use of the AMG solver is again an attractive alternative to the direct solution process.

### 7.7.1 Avoiding oversolving

---

**Algorithm 5** Inexact Newton Method.

---

```

1: input: [ $\mathbf{X}^{(j-1)}$ ,  $\mathbf{F}(\mathbf{X}^{(j-1)})$ ,  $\mathbf{K}(\mathbf{X}^{(j-1)})$ ,  $\eta_j$ ]
2: for ITER = 1, ... do
3:    $\Delta\mathbf{X}^{(j, \text{ITER})} := \text{ITERATIVE\_SOLVER}(\Delta\mathbf{X}^{(j, \text{ITER}-1)}, \mathbf{K}, \mathbf{F})$ 
4:   if  $\|\mathbf{F}(\mathbf{X}^{(j-1)}) + \mathbf{K}(\mathbf{X}^{(j-1)})\Delta\mathbf{X}^{(j, \text{ITER})}\| \leq \eta_j \|\mathbf{F}(\mathbf{X}^{(j-1)})\|$  break
5: end for
6: output:  $\mathbf{X}^{(j)} := \mathbf{X}^{(j-1)} + \Delta\mathbf{X}^{(j, \text{ITER})}$ 

```

---

Oversolving means that too much accuracy is imposed on the computation of the Newton step such that the nonlinear function and its local linear model disagree too much. Several choices for the so called forcing terms are proposed to overcome this problem. Algorithm 5 is based on [32]. The last Newton iterate, the residual, the Jacobi matrix of the last Newton step and the forcing term for the current Newton step are the input for the algorithm. The forcing term  $\eta_j$  is computed according to suggestions made in [37]. Only one of the possible choices is presented here:

$$\eta_j := \min \left\{ \frac{\|\mathbf{F}(\mathbf{X}^{(j-1)}) - \mathbf{F}(\mathbf{X}^{(j-2)}) - \mathbf{K}(\mathbf{X}^{(j-2)})\Delta\mathbf{X}^{(j-1)}\|}{\|\mathbf{F}(\mathbf{X}^{(j-2)})\|}, 1 - \epsilon \right\} \quad (7.27)$$

with a small number  $\epsilon$ . This criterion reflects the agreement between  $\mathbf{F}$  and its local linear model at the previous Newton step. Only if

$$\mathbf{F}(\mathbf{X}^{(j-1)}) \approx \mathbf{F}(\mathbf{X}^{(j-2)}) + \mathbf{K}(\mathbf{X}^{(j-2)})\Delta\mathbf{X}^{(j-1)}$$

(and in consequence  $\eta_j$  small), then it is worth it to perform many linear solver iterations since otherwise  $\mathbf{X}^{(j)}$  will be too far away from the nonlinear solution path. In the numerical examples of this chapter, the maximum norm  $\|\cdot\|_\infty$  is used as long as not stated otherwise.

### 7.7.2 Iterative solver

As an example for a solver in Algorithm 5, the following combined iterative solvers are given:

**Schur complement method** If one wants to use the Schur complement method, despite of the loss of superlinear convergence, then one has to perform the following steps:

- **Small deformations, elastic and contact:** Solve in this order:
  1. (7.19) directly
  2. (7.20) using the AMG solver
  3. (7.21) directly with the approximation  $\mathbf{U}_H^{(j,l_{fin})}$  ( $l_{fin}$  = final AMG step)
- **Small deformations, plastic and contact:** Solve in this order:
  1. (7.23) directly
  2. (7.24) using the AMG solver
  3. (7.25) directly with the approximation  $\mathbf{U}_H^{(j,l_{fin})}$

**Block Gauss Seidel** If one does not want to work with the Schur complement, then one has to iterate the Block-GS and solve

1. (7.11) directly
2. (7.12) using the AMG solver

Then, a good strategy to coordinate the coarse-fine loop (index  $k$ ) and the AMG solver loop (index  $l$ ) has to be found to avoid oversolving now with respect to the Block Gauss Seidel and the AMG solver. Therefore, the residual  $\|\mathbf{R}^{(j,k,l)}\|_\infty$  of the system [(7.11), (7.12)] in the  $j$ th Newton step, in the  $k$ th Block-GS step and in the  $l$ th AMG step is investigated,

$$\mathbf{R}^{(j,k,l)} := \begin{bmatrix} \mathbf{R}_h^{(j,k,l)} \\ \mathbf{R}_{AMG}^{(j,k,l)} \end{bmatrix}$$

The fine part of the residual is due to (7.11)

$$\mathbf{R}_h^{(j,k,l)} = \mathbf{K}_{hh}^{(j-1)} \Delta \mathbf{X}_h^{(j,k)} + \mathbf{F}_h^{(j-1)} + \mathbf{K}_{hH}^{(j-1)} \Delta \mathbf{X}_H^{(j,k,l)}$$

where  $\Delta \mathbf{X}_H^{(j,k,l)}$  is the  $l$ th iteration in the solution of (7.12). The coarse part is due to (7.12)

$$\mathbf{R}_{AMG}^{(j,k,l)} = \mathbf{K}_{HH}^{(j-1)} \Delta \mathbf{X}_H^{(j,k,l)} + \mathbf{F}_H^{(j-1)} + \mathbf{K}_{Hh}^{(j-1)} \Delta \mathbf{X}_h^{(j,k)}$$

Since

$$\|\mathbf{R}^{(j,k,l)}\|_\infty = \max\{\|\mathbf{R}_h^{(j,k,l)}\|_\infty, \|\mathbf{R}_{AMG}^{(j,k,l)}\|_\infty\} \quad (7.28)$$

a straightforward strategy to avoid oversolving would be to stop the AMG iteration as soon as

$$\|\mathbf{R}_{AMG}^{(j,k,l)}\|_\infty \leq \|\mathbf{R}_h^{(j,k,l)}\|_\infty$$

two times in a row.

## 7.8 Complete algorithm

The complete improved coupling algorithm for the plastic case is given in Algorithm 6.

---

**Algorithm 6** Improved coupling algorithm, plastic case.

---

```

1: assemble  $\mathbf{P}^{GPS \rightarrow C} := \mathbf{P}^{GPS \rightarrow C}(\mathbb{G}_{\hat{h}}, \mathbb{G}_H)$ 
2: time loop
3: for  $n = 1, \dots$  do
4:   displace fine grid  $\mathbb{T}_{h,n+1} := \mathbb{T}_{h,n} + \mathbf{t}_{n+1}$ 
5:   assemble mortar operator  $\mathbf{M}_{n+1} := \mathbf{M}(\mathbb{T}_{h,n+1}, \mathbb{T}_H)$ 
6:   assemble  $\mathbf{P}_{n+1} := \mathbf{P}(\mathbb{T}_H, \mathbb{T}_{h,n+1})$ 
7:   assemble  $\mathbf{P}_{n+1}^{GPF \rightarrow S} := \mathbf{P}^{GPF \rightarrow S}(\mathbb{G}_{h,n+1}, \mathbb{G}_{\hat{h}})$ 
8:   assemble  $\mathbf{P}_{n+1}^{GPS \rightarrow F} := \mathbf{P}^{GPS \rightarrow F}(\mathbb{G}_{\hat{h}}, \mathbb{G}_{h,n+1})$ 
9:   read fine data from  $\mathbb{G}_{\hat{h}}$  :  $\mathbb{E}_{h,n} := \mathbf{P}_{n+1}^{GPS \rightarrow F}[\mathbb{E}_{\hat{h},n}]$ 
10:  Newton iteration
11:  for  $j = 1, \dots$  do
12:    assemble  $\mathbf{F}_{H,n}^{pl} := \mathbf{F}^{pl}(\mathbf{P}^{GPS \rightarrow C}[\mathbb{E}_{\hat{h},n}])$ 
13:    assemble  $\mathbf{F}_{h,n}^{pl} := \mathbf{F}^{pl}(\mathbb{E}_{h,n})$ 
14:    assemble Equations (7.7) and (7.8)
15:    use Algorithm 5 with the input
        
$$[\mathbf{X}^{(j-1)}, \mathbf{F}(\mathbf{X}^{(j-1)}), \mathbf{K}(\mathbf{X}^{(j-1)}), \eta_j]$$

        to compute  $[\mathbf{X}^{(j)}, \mathbb{E}_{h,n+1}^{(j)}]$ 
16:    if  $\|\mathbf{F}(\mathbf{X}^{(j)})\| < \text{TOL}_{newt}$  goto 17 else goto 12 endif
17:  end for
18:  update  $\mathbb{E}_{\hat{h},n+1} := \mathbf{P}_{n+1}^{GPF \rightarrow S}[\mathbb{E}_{h,n+1}^{(jfin)}]$ 
19: end for

```

---

## 7.9 Numerical results

Two numerical results are presented in this section. The setting is described in Section 7.9.1.

### 7.9.1 Model example

For the two numerical examples in this Section, the same test setting is considered, see Figure 7.1. The coarse domain is  $\Omega_c = [0, 50] \times [0, 50] \times [0, 1]$  and  $\Omega_f = [2, 8] \times [2, 8] \times [0, 1]$ . The material parameters and the tool geometry are the same as in the chapters before. The mesh sizes are again  $H = 1.0$  and  $h = 0.5$ . Only the first load step where the tool indents the workpiece by 0.01 is considered.

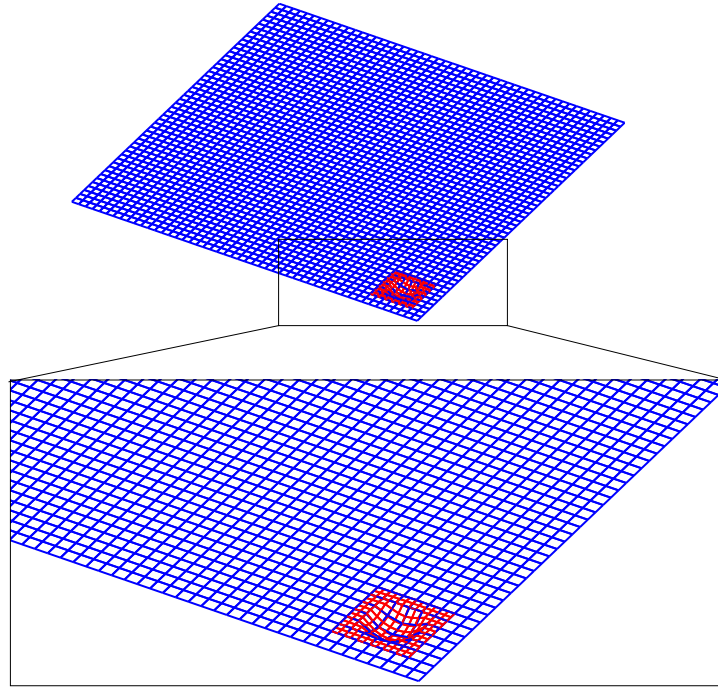


Figure 7.1: Test setting.

### 7.9.2 Schur complement method

In Figure 7.2, the residual  $\|\mathbf{F}\|$  is plotted over the Newton iterations. One can see that superlinear convergence is lost. Nevertheless the convergence speed is at least for this example still acceptable. The solution of (7.23) - (7.25) can be viewed again from the perspective of inexact Newton methods, since it is an approximate solution of the global system

$$\mathbf{K}^{(j-1)} \Delta \mathbf{X}^{(j)} = -\mathbf{F}^{(j-1)}$$

However, it is harder to control than standard inexact Newton methods, since (7.23) - (7.25) define only one solution step and no further correction steps.

### 7.9.3 Block Gauss Seidel

In this section, an example for the application of Algorithm 6 is given. The Newton residual  $\|\mathbf{F}\|$  (dashed line) and the block GS residual  $\|\mathbf{F} + \mathbf{K}\Delta \mathbf{X}\|$  (solid line) are plotted versus the GS steps in Figure 7.3. A first computation is an exact one, where the block GS converges down to  $10^{-12}$  in each Newton step. In a second computation, the inexact strategy in Algorithm 6 is used. Most of the oversolving can be avoided and most of the time, the two residuals stick together, indicating that the iterative solver does not leave the nonlinear solution path very far and does not waste too much effort. Some oversolving in the beginning of the Newton iteration is not unusual, see [37]. In the exact



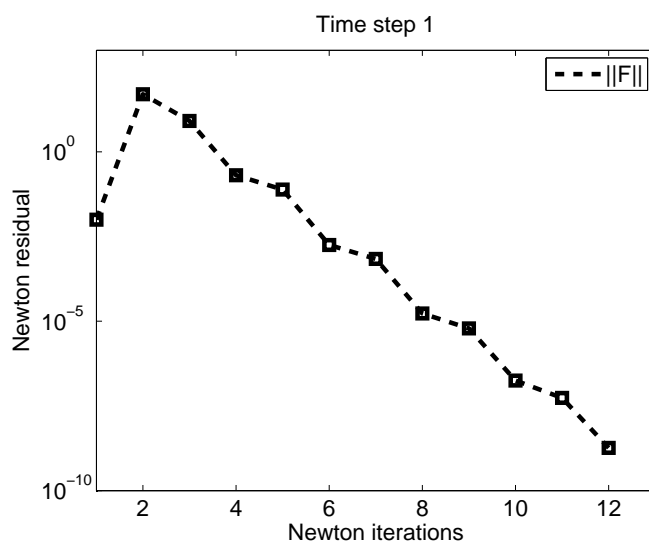


Figure 7.2: Convergence of the Schur complement method.

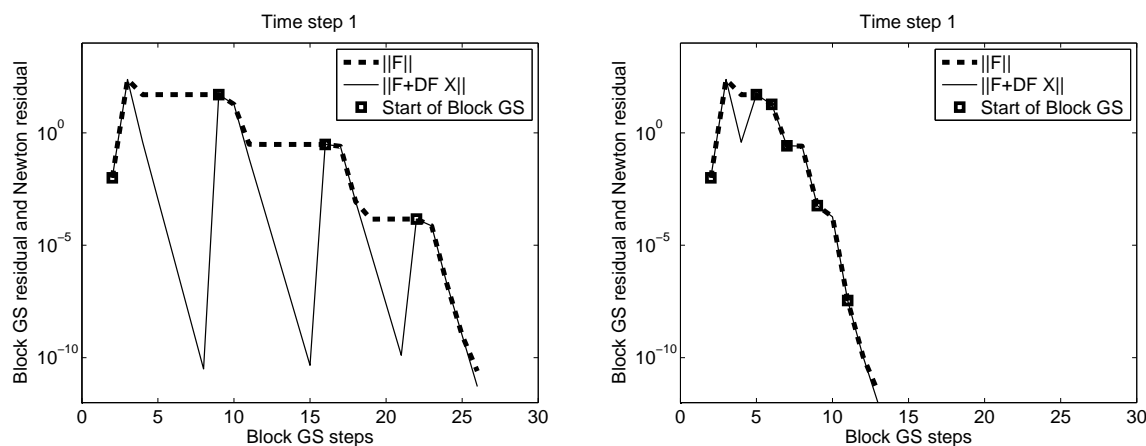


Figure 7.3: Exact and inexact block Newton scheme.

computation,  $\|\mathbf{F}\|$  is stagnant most of the time. Nearly 50 % of the overall number of GS steps are saved, although the inexact computation needs some more Newton iterations.

*Remark:* The solid line in the upper part of Figure 7.3 does not completely go down to the GS tolerance  $10^{-12}$  since the residuals are recorded *before* solving [(7.11),(7.12)].

## 7.10 Summary

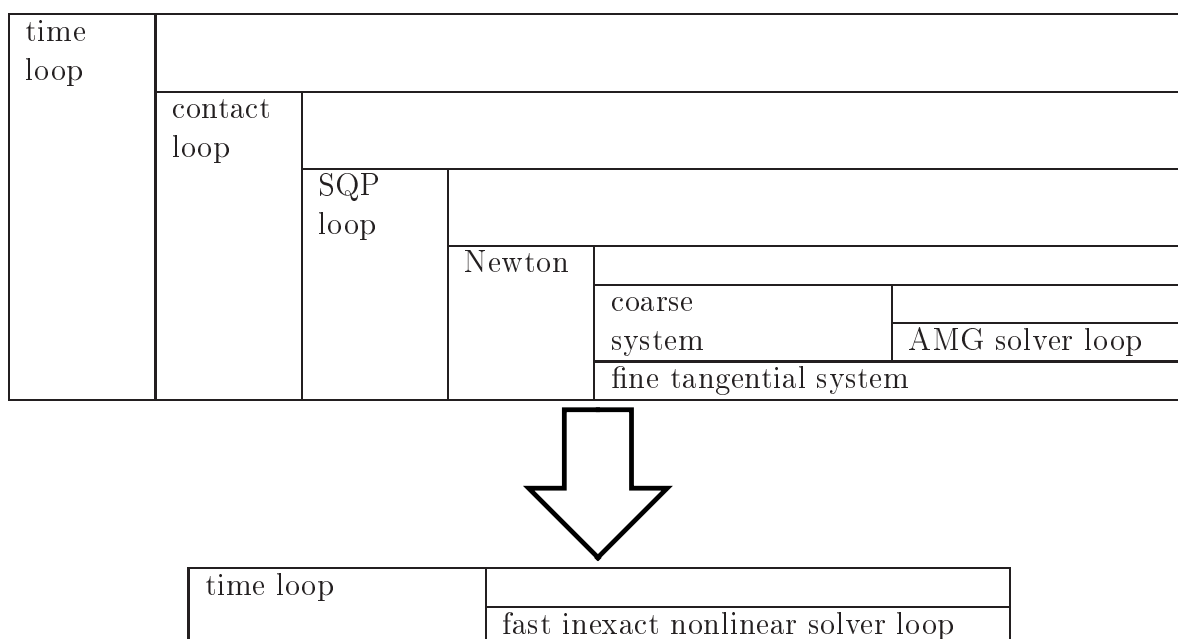
Some concluding remarks are made:

- It turns out that the ODDM defines in a very natural way a partitioned solu-

tion algorithm which shows in combination with classical inexact Newton schemes satisfying convergence.

- The algorithmic interface between coarse and fine is still small.
- The use of inexact Newton methods in this context was insofar interesting as their use is not so commonplace as in fluid mechanics or in fluid-structure interaction.
- In view of Figure 7.3, one can see again that iterative solvers are especially useful in nonlinear problems since they can be used in inexact Newton methods to find a very short nonlinear solution part, which is in this sense not possible when a direct solver is used. So even if a certain iterative solver is for a material linear problem slower than a direct solver (maybe due to problems in the present computer architecture), it gains further efficiency if the problem becomes nonlinear.

As future development is planned: The combination of the contact active set loop and the SQP iteration would be an interesting task. The inexact organization of the SQP loop for the fulfillment of the yield condition and the inner Newton loop for the mechanical equilibrium was discussed in [105]. Finally, with the results of this chapter and ongoing work on the integration of the AMG solver, one arrives at the following chart:



# 8 Large deformations

In this chapter, an outlook is given of what has to be done to extend the ODDM method to the regime of large deformations. The description follows the notation in [21].

## 8.1 Definitions

First of all, some definitions have to be made.

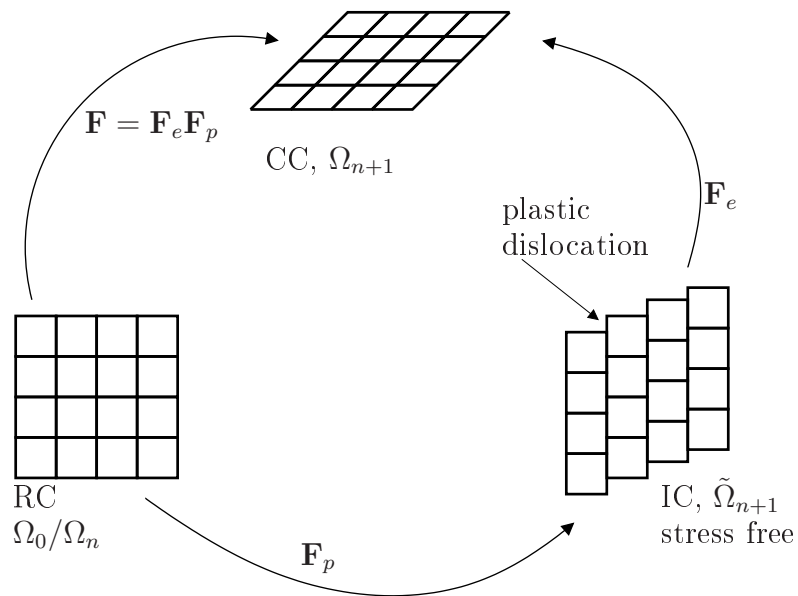


Figure 8.1: Configurations, the magnitude of the plotted elements is beneath the magnitude of the finite elements.

Table 8.1: Configurations.

abbreviation	configuration
RC	reference configuration
IC	intermediate configuration
CC	current configuration

**Deformation gradient** Let the vector field  $\mathbf{x}_t$  be the coordinates of the material points in configuration  $\Omega_t$ . Let

$$\mathbf{F}_0^t := \frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_0} = \mathbf{1} + \nabla_0(\mathbf{u}_t) = \mathbf{1} + \frac{\partial \mathbf{u}_t}{\partial \mathbf{x}_0}$$

be the deformation gradient for the deformation from  $\Omega_0$  to  $\Omega_t$ . With  $J_0^t := \det(\mathbf{F}_0^t)$  the determinant of the deformation gradient  $\mathbf{F}_0^t$  is defined.

### 8.1.1 Stress and strain tensors

The first and the second Piola-Kirchhoff stress tensor are defined by

$$\mathbf{P}_0^t := J_0^t(\mathbf{F}_0^t)^{-1}\boldsymbol{\sigma}_t \quad (8.1)$$

$$\mathbf{T}_0^t := \mathbf{P}_0^t(\mathbf{F}_0^t)^{-1} \quad (8.2)$$

The upper index is the time index and the lower index indicates with respect to *which configuration* the strain or stress is meant. Despite this, strains or stresses marked with an overline are meant *with respect to and at* the current time. In this section, these indices are omitted, and it is assumed that every stress or strain or deformation gradient which is not marked is meant to be  $[\cdot]_0^t$ . The following tensor representations are defined:

$$\text{Right Cauchy Green: } \mathbf{C} := (\mathbf{F})^\top(\mathbf{F}) \quad (8.3)$$

$$\text{Left Cauchy Green: } \overline{\mathbf{C}} := \mathbf{F}^{-\top}\mathbf{F}^{-1} \quad (8.4)$$

$$\text{Green's strain: } \mathbf{E} := \frac{1}{2}(\mathbf{C} - \mathbf{1}) \quad (8.5)$$

$$\text{Almansi strain: } \overline{\mathbf{E}} := \frac{1}{2}(\mathbf{1} - \overline{\mathbf{C}}) \quad (8.6)$$

As in [81], define by

$$\begin{aligned} \phi_\star(\cdot) &:= \mathbf{F}^{-\top}(\cdot)\mathbf{F}^{-1} \\ \phi^\star(\cdot) &:= \mathbf{F}^\top(\cdot)\mathbf{F} \end{aligned} \quad (8.7)$$

the push-forward and the pull-back operator. For instance, one can see immediately that

$$\mathbf{C} = \phi^\star(\mathbf{1}) \quad \text{and} \quad \overline{\mathbf{C}} = \phi_\star(\mathbf{1})$$

So the right Cauchy-Green tensor is the pull-back of the identity in the current configuration (CC) into the reference configuration (RC), and the left Cauchy-Green tensor is the push-forward of the identity in RC into CC.

### 8.1.2 Multiplicative decomposition

Following for example [78], it can be assumed that a multiplicative decomposition

$$\mathbf{F} = \mathbf{F}_e\mathbf{F}_p$$

is valid with the local intermediate, stress-free placement  $\mathbf{F}_p$  and the consecutive elastic deformation  $\mathbf{F}_e$ , see Figure 8.1. In general, neither  $\mathbf{F}_e$  nor  $\mathbf{F}_p$  are deformation gradients, so for example, there exists no  $\mathbf{u}_p \in [H_0^1(\Omega)]^d$  such that

$$\mathbf{F}_p(\mathbf{x}_0) = \nabla \mathbf{u}_p(\mathbf{x}_0) + \mathbf{1}, \quad \mathbf{x}_0 \in \Omega$$

### 8.1.3 Applications of pull-back and push-forward

The Green strain tensor  $\mathbf{E}$  with respect to RC is the pull back of Almansi strain tensor  $\bar{\mathbf{E}}$ :

$$\phi^*(\bar{\mathbf{E}}) = \mathbf{F}^\top(\bar{\mathbf{E}})\mathbf{F} = \mathbf{F}^\top \frac{1}{2}(\mathbf{1} - \mathbf{F}^{-\top}\mathbf{F}^{-1})\mathbf{F} = \mathbf{E}$$

The definition (8.7) can be generalized such that a push-back and a push-forward with respect to a factor of the deformation gradient  $\mathbf{F}$  can be defined:

$$\begin{aligned} \phi_{\star}^{\bullet}(\cdot) &:= \mathbf{F}_{\bullet}^{-\top}(\cdot)\mathbf{F}_{\bullet}^{-1} && \text{push-forward} \\ \phi^{\star\bullet}(\cdot) &:= \mathbf{F}_{\bullet}^{\top}(\cdot)\mathbf{F}_{\bullet} && \text{pull-back} \end{aligned} \quad (8.8)$$

Herein,  $\bullet \in \{e, p\}$  indicates whether the elastic or the plastic part of  $\mathbf{F}$  is used. Another example: There are two ways to compute  $\tilde{\mathbf{C}}^p$ : either as push-forward of the identity into the intermediate configuration (IC) in accordance with (8.3)

$$\tilde{\mathbf{C}}^p = \mathbf{F}^{p-\top}\mathbf{F}^{p-1} \quad (8.9)$$

or one can easily derive  $\tilde{\mathbf{C}}^p$  as pull-back (now with  $\mathbf{F}^e$ , not  $\mathbf{F}$ ) of  $\bar{\mathbf{C}}$  from CC to IC:

$$\tilde{\mathbf{C}}^p = \phi^{\star e}(\bar{\mathbf{C}}) = \mathbf{F}^{e\top}\mathbf{F}^{-\top}\mathbf{F}^{-1}\mathbf{F}^e = \mathbf{F}^{p-\top}\mathbf{F}^{p-1}$$

It is not possible to speak about  $\tilde{\mathbf{C}}$ . This can be seen from the calculation of  $\tilde{\mathbf{C}}^e$  as pull-back of the identity on CC into IC.

$$\tilde{\mathbf{C}}^e = \phi^{\star e}(\mathbf{1}) = \mathbf{F}^{e\top}\mathbf{F}^e \quad (8.10)$$

### 8.1.4 Additive decomposition of Green's strain

According to Equations (8.5) and (8.10), it can be derived:

$$\mathbf{E}^p = \frac{1}{2}(\mathbf{C}^p - \mathbf{1}) = \frac{1}{2}(\mathbf{F}^{p\top}\mathbf{F}^p - \mathbf{1}) \quad (8.11)$$

$$\tilde{\mathbf{E}}^e = \frac{1}{2}(\tilde{\mathbf{C}}^e - \mathbf{1}) = \frac{1}{2}(\mathbf{F}^{e\top}\mathbf{F}^e - \mathbf{1}) \quad (8.12)$$

With the aid of (8.11) and (8.9),  $\tilde{\mathbf{E}}^p$  is now expressed as the push-forward of  $\mathbf{E}^p$  into the intermediate configuration:

$$\tilde{\mathbf{E}}^p = \phi_{\star}^p(\mathbf{E}^p) = \mathbf{F}^{p-\top}\mathbf{E}^p\mathbf{F}^{p-1} = \frac{1}{2}(\mathbf{I} - \mathbf{F}^{p-\top}\mathbf{F}^{-p}) = \frac{1}{2}(\mathbf{1} - \tilde{\mathbf{C}}^p)$$

Furthermore,  $\mathbf{E}^e$  can be understood with the aid of (8.11) and (8.12) as pull-back of  $\tilde{\mathbf{E}}^e$  from IC to RC.

$$\mathbf{E}^e = \phi^{*p}(\tilde{\mathbf{E}}^e) = \mathbf{F}^{p\top} \tilde{\mathbf{E}}^e \mathbf{F}^p = \frac{1}{2}(\mathbf{F}^{p\top} \mathbf{F}^{e\top} \mathbf{F}^e \mathbf{F}^p - \mathbf{F}^{p\top} \mathbf{F}^p) = \frac{1}{2}(\mathbf{C} - \mathbf{C}^p) \quad (8.13)$$

*Remark:* It is not possible to compute  $\mathbf{E}^e$  in terms of (8.5) as  $\frac{1}{2}(\mathbf{F}_e^\top \mathbf{F}_e - \mathbf{I})$ , since  $\mathbf{E}^e$  is valid with respect to RC whereas  $\mathbf{F}_e$  is living on RC.

If one tries now to find a meaningful additive decomposition of Green's strain tensor, one gets with (8.11) and (8.13) the following result:

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{1}) = \frac{1}{2}(\mathbf{C} - \mathbf{C}^p) + \frac{1}{2}(\mathbf{C}^p - \mathbf{1}) = \mathbf{E}^e + \mathbf{E}^p \quad (8.14)$$

## 8.2 Equilibrium, Total Lagrange

With the results gathered in the last sections, the Total Lagrange (TL) equilibrium can be stated. At first, some notations have to be clarified. Let  $\Omega_n = \Omega_t$  be the last known configuration and let  $\Omega_{n+1} = \Omega_{t+\Delta t}$  be the unknown configuration. By  $\mathbf{E}(\mathbf{u}; \mathbf{v})$ , the directional derivative of  $\mathbf{E}$  at  $\mathbf{u}$  in direction of  $\mathbf{v}$  is denoted:

$$\begin{aligned} \mathbf{E}(\mathbf{u}; \mathbf{v}) &:= \lim_{k \rightarrow 0} \frac{1}{k} (\mathbf{E}(\mathbf{u} + k\mathbf{v}) - \mathbf{E}(\mathbf{u})) \\ &= \lim_{k \rightarrow 0} \frac{1}{2k} (\nabla(k\mathbf{v}) + \nabla(k\mathbf{v})^\top + \nabla(k\mathbf{v})\nabla(k\mathbf{v})^\top + \nabla(\mathbf{u})\nabla(k\mathbf{v})^\top + \nabla(k\mathbf{v})\nabla(\mathbf{u})^\top) \\ &= \text{sym}(\nabla(\mathbf{v})^\top + \nabla(\mathbf{u})\nabla(\mathbf{v})^\top) \end{aligned} \quad (8.15)$$

with

$$\text{sym}(\mathbf{A}) := \frac{1}{2}(\mathbf{A} + \mathbf{A}^\top)$$

The main problem in the regime of large deformations is that the domain, where the solution of the PDE has to be found, is solution dependent itself. This means that the weak form is stated with respect to the unknown configuration  $\Omega_{t+\Delta t}$ . At the moment, only the internal forces are considered. Using

$$d\Omega_t = J_0^t d\Omega_0 \quad (8.16)$$

$$\mathbf{A} : \mathbf{B} = \text{tr}(\mathbf{A}\mathbf{B}) \quad (8.17)$$

$$\nabla_t(\cdot) = (\mathbf{F}_0^t)^{-\top} \nabla_0(\cdot) \quad (8.18)$$

and (8.1) as well as (8.2), it can be derived:

$$\begin{aligned}
 & f^{int}(\mathbf{u}_{n+1}, \mathbf{v}) \tag{8.19} \\
 &= \int_{\Omega_{n+1}} \boldsymbol{\sigma}_{n+1} : \nabla_{n+1}(\mathbf{v})^\top d\Omega_{n+1} \\
 &= \int_{\Omega_0} J_0^{n+1} \boldsymbol{\sigma}_{n+1} : \nabla_{n+1}(\mathbf{v})^\top d\Omega_0 \\
 &= \int_{\Omega_0} (\mathbf{F}_0^{n+1} \mathbf{P}_0^{n+1}) : \nabla_0(\mathbf{v})^\top (\mathbf{F}_0^{n+1})^{-1} d\Omega_0 \tag{8.20} \\
 &= \int_{\Omega_0} \mathbf{P}_0^{n+1} : \nabla_0(\mathbf{v})^\top d\Omega_0 \\
 &= \int_{\Omega_0} \mathbf{T}_0^{n+1} : (\mathbf{F}_0^{n+1})^\top \nabla_0(\mathbf{v})^\top d\Omega_0 \\
 &= \int_{\Omega_0} \mathbf{T}_0^{n+1} : (\mathbf{1} + \nabla_0(\mathbf{u}_{n+1})) \nabla_0(\mathbf{v})^\top d\Omega_0 \\
 &= \int_{\Omega_0} \mathbf{T}_0^{n+1} : \mathbf{E}(\mathbf{u}_{n+1}; \mathbf{v}) d\Omega_0 \tag{8.21}
 \end{aligned}$$

The last equality holds because the second Piola Kirchhoff (PK) stress is a symmetric matrix. The gradient  $\nabla_{n+1}$  comes from the strong form of the mechanical problem, where a divergence operator is acting with respect to the current configuration. It is important to note, that if one writes  $\nabla_{n+1}(\mathbf{v})$  it is indeed meant  $\nabla_{n+1}(\mathbf{v} \circ \boldsymbol{\varphi}^{-1})$  since the test function  $\mathbf{v}$  is defined on the reference domain  $\Omega_0$ . With this notation, Equation (8.20) becomes apparent ( $\boldsymbol{\varphi}^{-1}(\mathbf{x}_{n+1}) = \mathbf{x}_0$ ):

$$\nabla_{n+1}(\mathbf{v}(\boldsymbol{\varphi}^{-1}(\mathbf{x}_{n+1}))) = \nabla_0(\mathbf{v}) \frac{d\boldsymbol{\varphi}^{-1}}{d\mathbf{x}_{n+1}} = \nabla_0(\mathbf{v})(\mathbf{F}_0^{n+1})^{-1}$$

In the following, this notation is omitted most of the time. In view of (8.21), it is investigated if the TL formulation is suitable for the ODDM. Therefore, the plastic state variable  $\mathbf{F}^p$  is assumed to be known in this section. If the St.Venant-Kirchhoff material law  $\mathbf{T} = \mathbf{C}^{el} \mathbf{E}^e$  is applicable, then the global equilibrium can be stated in this way:

$$\begin{aligned}
 & \int_{\Omega_0} \mathbf{E}(\mathbf{u}_{n+1}) : \mathbf{C}^{el} : \mathbf{E}(\mathbf{u}_{n+1}; \mathbf{v}) d\Omega_0 = \int_{\Omega_0} \mathbf{E}^p : \mathbf{C}^{el} : \mathbf{E}(\mathbf{u}_{n+1}; \mathbf{v}) d\Omega_0 + \text{ext. forces} \\
 \Leftrightarrow & \int_{\Omega_0} \mathbf{E}(\mathbf{u}_{n+1}) : \mathbf{C}^{el} : \mathbf{E}(\mathbf{u}_{n+1}; \mathbf{v}) - \mathbf{E}^p : \mathbf{C}^{el} : \nabla_0(\mathbf{u}_{n+1}) \nabla_0(\mathbf{v})^\top d\Omega_0 \\
 &= \int_{\Omega_0} \mathbf{E}^p : \mathbf{C}^{el} : \nabla_0(\mathbf{v})^\top d\Omega_0 + \text{ext. forces}
 \end{aligned}$$

Therefore, one finally arrives at a linear material (for known  $\mathbf{E}^p$ ), but still the following difficulties arise:

1. A geometric nonlinear problem (expensive reassemblies of the coarse system in each Newton step).

Table 8.2: Green strains.

configuration	
RC	$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{1})$
IC	$\tilde{\mathbf{E}} = \frac{1}{2}(\tilde{\mathbf{C}}^e - \mathbf{C}^p)$
CC	$\bar{\mathbf{E}} = \frac{1}{2}(\mathbf{1} - \bar{\mathbf{C}})$

2. In addition, the plastic strain  $\mathbf{E}^p$  does not only appear as load correction term on the right hand side but also on the left hand side in a geometric stiffness.
3. Moreover, the St.Venant-Kirchhoff law is only valid for large displacements and not for large deformations. In the case of a general hyperelastic relation ( $\Psi^e$  is the elastic potential )

$$\mathbf{T} = \rho_0 \frac{\partial \Psi^e}{\partial \mathbf{E}^e} \quad (8.22)$$

between stress and elastic strain, the situation gets even more complicated. Then  $\mathbf{E}^e$  has to be computed, hence  $\mathbf{F}^p$  enters even the coarse material stiffness and not only the right hand side.

Thus, it must be summarized, that the additive split (8.14) does not lead to a formulation which makes the application of the domain decomposition feasible. Nevertheless a workaround is sought at least for point 3: Using (8.9) one gets

$$\begin{aligned} \tilde{\mathbf{E}} &= \phi_{\star}^p(\mathbf{E}) \\ &= \frac{1}{2}(\mathbf{F}^{p-\top} \mathbf{C} \mathbf{F}^{-p} - \mathbf{F}^{p-\top} \mathbf{F}^{-p}) \\ &= \frac{1}{2}(\mathbf{F}^{e\top} \mathbf{F}^e - \mathbf{F}^{p-\top} \mathbf{F}^{-p}) \\ &= \frac{1}{2}(\tilde{\mathbf{C}}^e - \tilde{\mathbf{C}}^p) \\ &= \tilde{\mathbf{E}}^e + \tilde{\mathbf{E}}^p \end{aligned}$$

In  $\tilde{\mathbf{E}}^e$  and  $\tilde{\mathbf{E}}^p$ , the elastic and the plastic part of  $\mathbf{F}$  are clearly separated, but the problem is that one cannot work in the IC since it is nonconforming. So this is not an appropriate workaround. To give a better overview, the three most important representations of  $\mathbf{E}$  are given in Table 8.2.

### 8.3 Constitutive law

This section follows the notation in [46]. The hyperelastic material tensor is defined by

$$\mathbf{C}^{hyp} := 4 \frac{\partial^2 \Psi_e}{\partial \mathbf{C} \partial \mathbf{C}^\top}$$



Without deriving the constitutive law, the four equations for the four unknown variables  $\mathbf{T}$ ,  $\boldsymbol{\alpha}$ ,  $E^p$  and  $\lambda$  are given:

$$\dot{\mathbf{T}} + \lambda \mathbf{C}^{hyp} : \frac{\partial \Phi}{\partial \mathbf{T}} - \frac{1}{2} \mathbf{C}^{hyp} : \dot{\mathbf{C}} = 0 \quad (8.23)$$

$$\dot{\boldsymbol{\alpha}} + \lambda \mathbf{Q}_1(\mathbf{T}, \boldsymbol{\alpha}) = 0 \quad (8.24)$$

$$\dot{E}^p + \lambda Q_2(\mathbf{T}, \boldsymbol{\alpha}) = 0 \quad (8.25)$$

$$\Phi(E^p, \mathbf{T}, \boldsymbol{\alpha}) = 0 \quad (8.26)$$

with  $\mathbf{Q}_1$  and  $Q_2$  denoting general constitutive functions for the evolution of the backstress  $\boldsymbol{\alpha}$  and the plastic arc length  $E^p$ . The function  $Q_2$  is assumed to be negative valued. All other variables have been defined before. The KKT conditions have already entered (8.23)- (8.26). Of special interest is (8.23), since it is an extra evolution equation for the stress which does not appear in common plasticity formulations. It is based on [51] and enhanced in [46]. The advantage of this extra equation is that there is no longer a need for an explicit back projection step which necessitates in many cases an awkward linearization. With this method, only a full linearization of system (8.23) – (8.26) is necessary to compute a stress which fulfills the yield condition.

## 8.4 Consistent linearization of the constitutive law

After time discretization, two options are possible. Both options lead finally to the same linear tangential stiffnesses:

1. First linearize the continuous nonlinear bivariate form  $f^{int}(\mathbf{u}, \mathbf{v})$  in direction of a Newton increment  $\Delta \mathbf{u}$ , and then discretize the resulting bilinear and linear forms spatially.
2. Perform the spatial discretization *before* performing the Newton linearizations.

In this chapter, the first way is pursued, since it leads to a compact notation and it supersedes the necessity to write down the spatial discretizations. Using (8.15) and the fact that the directional derivative of  $\nabla_0(\mathbf{u})$  in direction of  $\Delta \mathbf{u}$  is  $\nabla_0(\Delta \mathbf{u})$  since the gradient is a linear functional, an auxiliary calculation gives:

$$\mathbf{E}(\mathbf{u}_{n+1}; \Delta \mathbf{u}, \mathbf{v}) = \text{sym}(\nabla_0(\Delta \mathbf{u}) \nabla_0(\mathbf{v})^\top) \quad (8.27)$$

For computing the Jacobian of  $f^{int}(\mathbf{u}_{n+1}; \mathbf{v})$  in Equation (8.19), one has to compute the directional derivative in direction of the Newton increment  $\Delta \mathbf{u}$ :

$$\begin{aligned} & f^{int}(\mathbf{u}_{n+1}, \mathbf{v}; \Delta \mathbf{u}) \\ &= \int_{\Omega_0} \mathbf{T}(\mathbf{u}_{n+1}; \Delta \mathbf{u}) : \mathbf{E}(\mathbf{u}_{n+1}; \mathbf{v}) + \mathbf{T}(\mathbf{u}_{n+1}) : \mathbf{E}(\mathbf{u}_{n+1}; \mathbf{v}, \Delta \mathbf{u}) \, d\Omega_0 \\ &= \int_{\Omega_0} \mathbf{E}(\mathbf{u}_{n+1}; \Delta \mathbf{u}) : \mathbf{C}^{ep} : \mathbf{E}(\mathbf{u}_{n+1}; \mathbf{v}) + \mathbf{T}(\mathbf{u}_{n+1}) : \mathbf{E}(\mathbf{u}_{n+1}; \mathbf{v}, \Delta \mathbf{u}) \, d\Omega_0 \\ &= \int_{\Omega_0} \mathbf{E}(\mathbf{u}_{n+1}; \Delta \mathbf{u}) : \mathbf{C}^{ep} : \mathbf{E}(\mathbf{u}_{n+1}; \mathbf{v}) + \mathbf{T}(\mathbf{u}_{n+1}) : [\nabla_0(\Delta \mathbf{u}) \nabla_0(\mathbf{v})^\top] \, d\Omega_0 \end{aligned} \quad (8.28)$$

Above, (8.27) was used, as well as the definition

$$\mathbf{C}^{ep} := \frac{d\mathbf{T}(\mathbf{E})}{d\mathbf{E}}$$

which is the subject of the next section. The first summand of (8.28) is the material stiffness and the second summand is the geometric stiffness.

## 8.5 A Gauss point section: Generalized projection method

It is explained in this last Gauss point section, how the elastoplastic tangent in (8.28) can be computed for a very general plasticity model. It can be expected that the method, already mentioned in Section 8.3 and which shall be named here as ‘generalized projection’ method, will gain new interest in the FE simulation of incremental forming processes: The main disadvantage of the method is that a Newton iteration has to be performed for the stress tensor and for the plastic data at each integration point. Since the zone, where plastic state variables evolve, is relatively small in incremental processes, this disadvantage is outbalanced by the advantage of the less complex linearizations to be performed.

Some notations: Let  $\mathcal{G}$  be the nonlinear functional defined by the time discretized version of (8.23)-(8.26) and let

$$\mathbf{z} := (\mathbf{T}, \boldsymbol{\alpha}, E^p, \lambda)_{n+1}^\top \in \mathbb{R}^{14}$$

be the vector <sup>1</sup> of unknowns. So the task is to find some  $\mathbf{z}^*$  such that

$$\mathcal{G}(\mathbf{z}^*(\mathbf{E}_{n+1}^*), \mathbf{E}_{n+1}^*) = \mathbf{0} \quad (8.29)$$

see [51] and the references therein. Herein,  $\mathbf{E}_{n+1}^*$  has the meaning of the total Green strain in the current Newton step. The implicit function theorem states that there exist open sets  $\mathbb{U}_0 \supset \{\mathbf{E}_{n+1}^*\}$  and  $\mathbb{V}_0 \supset \{\mathbf{z}^*(\mathbf{E}_{n+1}^*)\}$  and an injective  $\mathcal{C}^1$  mapping

$$\mathbf{z} : \mathbb{U}_0 \rightarrow \mathbb{V}_0$$

such that

$$\mathcal{G}(\mathbf{z}(\mathbf{E}_{n+1}), \mathbf{E}_{n+1}) = \mathbf{0}, \quad \mathbf{E}_{n+1} \in \mathbb{U}_0$$

Thus

$$\begin{aligned} \frac{d\mathcal{G}}{d\mathbf{E}_{n+1}} \Big|_{\mathbf{E}_{n+1}=\mathbf{E}_{n+1}^*} &= \mathbf{0} \\ \Rightarrow \nabla_{\mathbf{z}} \mathcal{G} \frac{d\mathbf{z}}{d\mathbf{E}_{n+1}} \Big|_{\mathbf{E}_{n+1}=\mathbf{E}_{n+1}^*} &= - \frac{\partial \mathcal{G}}{\partial \mathbf{E}_{n+1}} \Big|_{\mathbf{E}_{n+1}=\mathbf{E}_{n+1}^*} \end{aligned} \quad (8.30)$$

---

<sup>1</sup>It is not explicitly stated in this chapter when tensor values variables are meant to be in Voigt notation

**Algorithm**  $[\mathbf{T}_{n+1}^{(j)}, \mathbf{C}_{n+1}^{ep,(j)}, \boldsymbol{\alpha}_{n+1}^{(j)}] = \text{GENERALIZED\_PROJEC}(\mathbf{E}_{n+1}^{(j-1)}, \mathbf{E}_n, \boldsymbol{\alpha}_n, \Phi, \mathbf{C}^{hyp})$

1. Compute trial values

$$\mathbf{T}_{n+1}^{tr} := \mathbf{T}_n + \mathbf{C}^{hyp}(\mathbf{E}_{n+1}^{(j-1)} - \mathbf{E}_n), \mathbf{C}_{n+1}^{ep,tr} := \mathbf{C}^{hyp}, \boldsymbol{\alpha}_{n+1}^{tr} := \boldsymbol{\alpha}_n$$

2. Check yield condition

$$\Phi_{n+1}^{tr} := \Phi(\mathbf{T}_{n+1}^{tr}, \boldsymbol{\alpha}_n)$$

IF  $\Phi_{n+1}^{tr} \leq 0$  THEN:

$$\text{SET } (\bullet)_{n+1}^{(j)} := (\bullet)_{n+1}^{tr} \text{ AND EXIT}$$

ENDIF

3. Solve problem (8.29) for  $[\mathbf{T}_{n+1}^{(j)}, \boldsymbol{\alpha}_{n+1}^{(j)}]$

4. Solve the 6 linear equation systems in (8.30) to get the consistent elastoplastic tangent module as part of

$$\frac{d\mathbf{z}_{n+1}^{(j)}}{d\mathbf{E}_{n+1}^{(j)}} = [\mathbf{C}_{n+1}^{ep,(j)}, \dots]^\top$$

## 8.6 Updated Lagrange

For a more detailed description of the updated Lagrange (UL) technique, see textbooks like for instance [66] and the references therein. Let  $\Delta \mathbf{u}_{n+1} := \mathbf{u}_{n+1} - \mathbf{u}_n$  be the incremental displacement from  $\Omega_n$  to  $\Omega_{n+1}$ . The main assumption of UL is that the configurations  $\Omega_n$  and  $\Omega_{n+1}$  are similar in the sense that in the load step  $t_n \rightarrow t_{n+1}$  the deformation is assumed to be small:

$$\int_{\Omega_{n+1}} \Delta \boldsymbol{\sigma}_{n+1} : \nabla_{n+1}(\mathbf{v})^\top d\Omega_{n+1} \approx \int_{\Omega_n} \Delta \boldsymbol{\sigma}_{n+1} : \nabla_n(\mathbf{v})^\top d\Omega_n \quad (8.31)$$

It is important to note here, that  $\Delta \boldsymbol{\sigma}_{n+1}$  on the left hand side is indeed related to the current configuration  $\Omega_{n+1}$ , actually it is the push-forward of  $\Delta \mathbf{T}_n^{n+1}$ , whereas  $\Delta \boldsymbol{\sigma}_{n+1}$  on the right hand side is related to  $\Omega_n$  and is meant in a small deformation manner. In this sense it should be made clear that  $\Delta \boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_{n+1} - \boldsymbol{\sigma}_n$  is meaningless, since two stresses are subtracted here which are measured with respect to two different configurations. Equation (8.31) shall be called the assumption of small time steps. It means that the stress increment  $\Delta \boldsymbol{\sigma}_{n+1}$  which would be only valid with respect to  $\Omega_{n+1}$  in the context of large deformations between  $\Omega_n$  and  $\Omega_{n+1}$ , is assumed to be valid with respect to  $\Omega_n$ . The assumption of small load steps can be justified, since otherwise the tool path is not resolved properly, see Figure 8.2. If the tool path is not resolved adequately then even the most sophisticated implicit time integration of the constitutive law and full linearization of all bivariate forms will not reduce the error in the boundary conditions. On the other

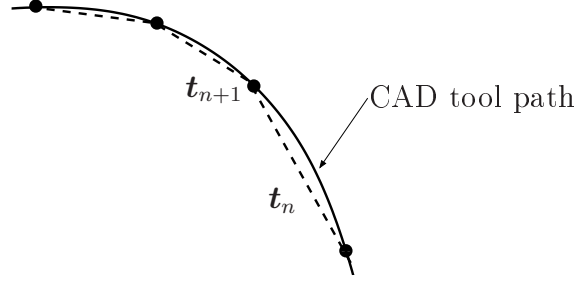


Figure 8.2: CAD tool path and approximation for the FE computation.

hand, the assumption of small time steps does not render full linearization unnecessary. On the contrary: The feature of a Newton method with a consistent Jacobi matrix to converge superlinearly or even quadratic in the proximity of the root, becomes particularly interesting if the load/time steps are small and therefore the starting solutions are good. Again, only the left hand side with the internal forces is considered. Using the same ideas as in (8.19) and

$$\mathbf{E}_n(\Delta \mathbf{u}_{n+1}; \mathbf{v}) = \text{sym}(\nabla_n(\mathbf{v})^\top + \nabla_n(\Delta \mathbf{u}_{n+1})\nabla_n(\mathbf{v})^\top)$$

according to (8.15), it can be derived:

$$\begin{aligned} & \int_{\Omega_{n+1}} \boldsymbol{\sigma}_{n+1} : \nabla_{n+1}(\mathbf{v})^\top d\Omega_{n+1} \\ = & \int_{\Omega_n} \mathbf{T}_n^{n+1} : \mathbf{E}_n(\Delta \mathbf{u}_{n+1}; \mathbf{v}) d\Omega_n \\ = & \int_{\Omega_n} \Delta \mathbf{T}_n^{n+1} : \mathbf{E}_n(\Delta \mathbf{u}_{n+1}; \mathbf{v}) d\Omega_n + \int_{\Omega_n} \boldsymbol{\sigma}_n : \nabla_n(\Delta \mathbf{u}_{n+1})\nabla_n(\mathbf{v})^\top d\Omega_n \\ & + \int_{\Omega_n} \boldsymbol{\sigma}_n : \nabla_n(\mathbf{v})^\top d\Omega_n \end{aligned} \quad (8.32)$$

Here, the additive decomposition of  $\mathbf{T}_n^{n+1}$  into  $\boldsymbol{\sigma}_n$  and

$$\Delta \mathbf{T}_n^{n+1} = \mathbf{T}_n^{n+1} - \boldsymbol{\sigma}_n$$

was used. Considering that  $\Delta \boldsymbol{\sigma}_{n+1}$  is the push-forward of  $\Delta \mathbf{T}_n^{n+1}$ , one can proceed:

$$\begin{aligned} & \int_{\Omega_{n+1}} \boldsymbol{\sigma}_{n+1} : \nabla_{n+1}(\mathbf{v})^\top d\Omega_{n+1} \\ = & \int_{\Omega_{n+1}} \Delta \boldsymbol{\sigma}_{n+1} : \nabla_{n+1}(\mathbf{v})^\top d\Omega_{n+1} + \int_{\Omega_n} \boldsymbol{\sigma}_n : \nabla_n(\Delta \mathbf{u}_{n+1})\nabla_n(\mathbf{v})^\top d\Omega_n \\ & + \int_{\Omega_n} \boldsymbol{\sigma}_n : \nabla_n(\mathbf{v})^\top d\Omega_n \end{aligned}$$

With the assumption (8.31), one arrives at:

$$\begin{aligned} & \int_{\Omega_{n+1}} \boldsymbol{\sigma}_{n+1} : \nabla_{n+1}(\mathbf{v})^\top d\Omega_{n+1} \\ & \approx \int_{\Omega_n} \Delta\boldsymbol{\sigma}_{n+1} : \nabla_n(\mathbf{v})^\top + \boldsymbol{\sigma}_n : \nabla_n(\Delta\mathbf{u}_{n+1})\nabla_n(\mathbf{v})^\top d\Omega_n \\ & \quad + \int_{\Omega_n} \boldsymbol{\sigma}_n : \nabla_n(\mathbf{v})^\top d\Omega_n \end{aligned} \quad (8.33)$$

$$=: f_n^{int}(\Delta\mathbf{u}_{n+1}; \mathbf{v}) + f_n^g(\Delta\mathbf{u}_{n+1}; \mathbf{v}) + l_n^g(\mathbf{v}) \quad (8.34)$$

The advantage of the formulation of the equilibrium in an updated Lagrange manner

$$f_n^{int}(\Delta\mathbf{u}_{n+1}; \mathbf{v}) + f_n^g(\Delta\mathbf{u}_{n+1}; \mathbf{v}) = -l_n^g(\mathbf{v}) + \text{contact forces}$$

is that on the left hand side, there is a problem, which is material nonlinear only in  $\Delta\mathbf{u}$ . With the aid of the ODDM, this can again be brought to the right hand side of the global coarse problem as load correction term. The effective stiffness of the global problem has to be updated in every load step due to the geometric stiffness, which is still on the left hand side, containing  $\boldsymbol{\sigma}_n$ . But this is nevertheless far more advantageous than the TL formulation, since that would demand a global reassembly in each Newton step. One has to consider that the geometric nonlinearity is always a global one.

### 8.6.1 Some iteration for UL

One can easily observe how large the error of the small-time-step assumption is, by checking the norm of the residual  $\mathbf{R}^{UL}$  which is defined as the standard FE assembly of

$$\int_{\Omega_n} \Delta\boldsymbol{\sigma}_{n+1} : \nabla_n(\mathbf{v}) - \int_{\Omega_{n+1}^{(1)}} \Delta\boldsymbol{\sigma}_{n+1} : \nabla_{n+1}(\mathbf{v})$$

with

$$\Omega_{n+1}^{(1)} := \Omega_n + \Delta\mathbf{u}_{n+1}$$

Obviously,  $\Omega_{n+1}^{(1)}$  is in general not equal to  $\Omega_{n+1}$  since the incremental solution  $\Delta\mathbf{u}_{n+1}$  is only an approximation because the domain, where it was computed, was  $\Omega_n$  and not the correct  $\Omega_{n+1}$ . A further development would be to perform an extra loop above the structural nonlinear solution process and make the residual  $\mathbf{R}^{UL,(l)}$ , which is the standard FE assembly of

$$\int_{\Omega_{n+1}^{(l-1)}} \Delta\boldsymbol{\sigma}_{n+1}^{(l)} : \nabla_{n+1,l-1}(\mathbf{v}) - \int_{\Omega_{n+1}^{(l)}} \Delta\boldsymbol{\sigma}_{n+1}^{(l)} : \nabla_{n+1,l}(\mathbf{v})$$

vanish for  $l \rightarrow \infty$ . Here,  $\Omega_{n+1}^{(l)}$  is defined by

$$\Omega_{n+1}^{(0)} := \Omega_n \quad (8.35)$$

$$\Omega_{n+1}^{(l)} := \Omega_n + \Delta\mathbf{u}_{n+1}^{(l)}, \quad l = 1, \dots \quad (8.36)$$

Proceeding like this, the domain is updated in every step of the extra UL loop. Bathe [8] states that it is possible to unroll the extra UL loop and the Newton loop such the update (8.36) is performed in each Newton step. Of course in this case, superlinear convergence cannot be guaranteed anymore and for an acceptable convergence rate, the assumption (8.31) is again needed.

## 8.7 Contact Total Lagrange

The main issue of this section is the transmission of the contact stresses from fine to coarse in the large deformation regime. Due to the large deformations and rigid body motions of the workpiece, the following question arises: In which configuration do the mortar operator  $\mathbf{D}$  and  $\mathbf{M}$  need to be assembled? This question is answered exemplarily for  $\mathbf{M}$ , since  $\mathbf{D}$  depends only on the fine contact interface which means that a change of the configuration would not cause severe problems. However it is much more convenient to set up  $\mathbf{M}$  in the reference configuration for the following reason: For many important examples for the ISF process, the unmachined sheet can be assumed to be planar, such that fine grid and coarse grid fit very well in the reference configuration and translational movement of the fine grid with the tool causes no problem. This nice property is lost, as soon as the coarse grid deforms due to the UL formulation. Then the biorthogonality relation changes due to the fact that a projection has to be performed to compute the entries of  $\mathbf{M}$  like it has to be done in the two-body contact with curved interfaces. In the next section, it will be shown how the important integrals for  $\mathbf{M}$  can be pulled back into the reference configuration.

### 8.7.1 Transformation of the contact stresses

Let  $\bar{\boldsymbol{\lambda}}_{n+1}$  be current contact stress with respect to time  $t_{n+1}$  and  $\boldsymbol{\varphi}$  is the mapping  $\Omega_0 \rightarrow \Omega_{n+1}$ . Then proceed in this way:

$$\begin{aligned}
& \int_{\Gamma_{n+1}^{con}} \bar{\boldsymbol{\lambda}}_{n+1}(\boldsymbol{\varphi}^{-1}(\mathbf{x}_{n+1})) \cdot \mathbf{v}(\boldsymbol{\varphi}^{-1}(\mathbf{x}_{n+1})) d\Gamma_{n+1}^{con} \\
&= \int_{\Gamma_{n+1}^{con}} [\boldsymbol{\sigma}_{n+1} \boldsymbol{\nu}_{n+1}] (\boldsymbol{\varphi}^{-1}(\mathbf{x}_{n+1})) \cdot \mathbf{v}(\boldsymbol{\varphi}^{-1}(\mathbf{x}_{n+1})) d\Gamma_{n+1}^{con} \\
&= \int_{\Gamma_0^{con}} [\boldsymbol{\sigma}_{n+1} J_0^{n+1} (\mathbf{F}_0^{n+1})^{-1} \boldsymbol{\nu}_0] (\mathbf{x}_0) \cdot \mathbf{v}(\mathbf{x}_0) d\Gamma_0^{con} \\
&= \int_{\Gamma_0^{con}} [\mathbf{P}_0^{n+1} \boldsymbol{\nu}_0] (\mathbf{x}_0) \cdot \mathbf{v}(\mathbf{x}_0) d\Gamma_0^{con} \\
&= \int_{\Gamma_0^{con}} \boldsymbol{\lambda}_0^{n+1} (\mathbf{x}_0) \cdot \mathbf{v}(\mathbf{x}_0) d\Gamma_0^{con} \tag{8.37}
\end{aligned}$$

Again, Nanson's formula was used here in the form

$$\boldsymbol{\nu}_{n+1} d\Gamma_{n+1}^{con} = J_0^{n+1} (\mathbf{F}_0^{n+1})^{-1} \boldsymbol{\nu}_0 d\Gamma_0^{con}$$

Equation (8.37) takes care of the transformation of the contact normal and of the Jacobian  $J_0^{n+1}$  and is therefore the reason, why the virtual contact work has the same structure in CC and RC. A remark has to be made to the notation:  $\bar{\lambda}_{n+1}$  is a stress *with respect to* the current configuration

$$\bar{\Gamma}^{con} = \Gamma_{n+1}^{con} = \varphi(\Gamma_0^{con})$$

but it is *defined* with respect to its domain  $\Gamma_0^{con}$ . Due to the bijectivity of  $\varphi^{-1}$ , it is in principle possible to define a  $\bar{\lambda}_{n+1}^*(\mathbf{x}_{n+1})$  with

$$\bar{\lambda}_{n+1}^* := \bar{\lambda}_{n+1} \circ \varphi^{-1}$$

but to  $\bar{\lambda}_{n+1}^*$  the integral transformation

$$\int_{\varphi(\Omega)} g(y)f(y)dy = \int_{\Omega} g(\varphi(x))f(\varphi(x))|detD_x(\varphi(x))|dx \quad (8.38)$$

is not directly applicable.

### 8.7.2 Mortar coupling fine to coarse in the large deformation regime

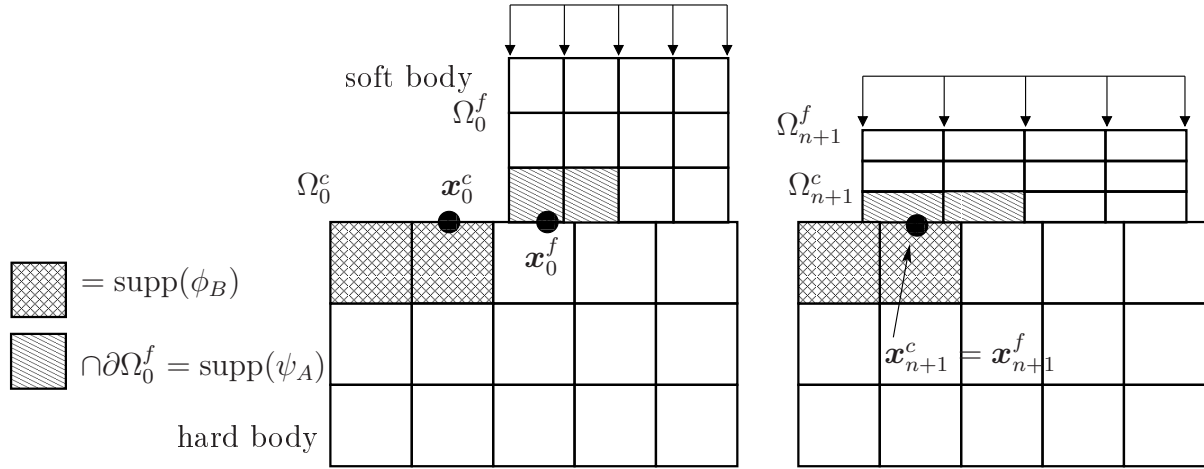


Figure 8.3: Large deformation two body contact.

The equilibrium stated on the global domain is:

$$\int_{\Omega_{n+1}^c} \sigma_{n+1} : \nabla_{n+1}(\mathbf{v}(\varphi_c^{-1}(\mathbf{x}_{n+1}^c))) + \int_{\Gamma_{n+1}^c} \bar{\lambda}_{n+1}^f(\varphi_f^{-1}(\mathbf{x}_{n+1}^f)) \cdot \mathbf{v}_c(\varphi_c^{-1}(\mathbf{x}_{n+1}^c)) = 0 \quad (8.39)$$

By  $\mathbf{x}^f$ , the total fine coordinates are meant, including the coarse placement. If one is in the situation of two-body contact problems with large relative motions, depicted in

Figure 8.3, then it is impossible to apply (8.38) to the second summand and to pull it back into the reference configuration, since  $\boldsymbol{\varphi}_f^{-1} \neq \boldsymbol{\varphi}_c^{-1}$ . For example in publications like [115], the operator  $\mathbf{M}$  has to be set up in  $\Gamma_{n+1}^c$

$$\mathbf{M}[A, B] = \int_{\Gamma_{n+1}^c} \psi_B(\boldsymbol{\varphi}_f^{-1}(\mathbf{x}_{n+1}^f)) \phi_A(\boldsymbol{\varphi}_c^{-1}(\mathbf{x}_{n+1}^c))$$

which means that it is solution dependent with all the awkward implications concerning consistent linearization. In the context of this work, large relative motions of the two grids are impossible since the fine grid solution does not substantially influence the coarse grid solution. Therefore, the second summand of Problem (8.39) can be pulled back with the aid of (8.37) into  $\Gamma_0^c$ , and can be discretized there. The vector  $\boldsymbol{\Lambda}$  is then keeping the discretized contact stresses of  $\boldsymbol{\lambda}_0^{n+1}$ , which does not play a role for the evaluation of  $\mathbf{F}^{con}$ , as  $\boldsymbol{\lambda}_0^{n+1}$  and  $\bar{\boldsymbol{\lambda}}_{n+1}$  differ only in a positive (even though spatially not constant) factor. With  $\boldsymbol{\Lambda}$  being meant in this way, one has to take care that the mortar operator  $\mathbf{D}$  which discretizes the coupling

$$\int_{\Gamma_{n+1}^f} \bar{\boldsymbol{\lambda}}_{n+1}^f(\boldsymbol{\varphi}_f^{-1}(\mathbf{x}_{n+1}^f)) \cdot \mathbf{v}_f(\boldsymbol{\varphi}_f^{-1}(\mathbf{x}_{n+1}^f))$$

has to be pulled back into RC, too. Otherwise the two mortar couplings do not fit together. On the other hand, the weak nonpenetration condition has to be set up on  $\Omega_n$ , because the incremental displacements are discretized there.

## 8.8 Conclusions

It can be concluded that in the regime of large deformations, the ODDM is in principle possible with a fine balanced combination of Total Lagrange and Updated Lagrange methods, but both the volumetric and the surface coupling of the fine and the coarse grid are difficult.

- In principle, the mortar coupling of the contact stress between fine and coarse grid is possible if
  1. The workpiece is planar and the assemblation is done in the reference configuration or:
  2. Alternatively, one works with complicated projections.
- Due to the deformation of the coarse grid and the fine grid, also the construction of the data mapping operators  $\mathbf{P}^{GP}$  becomes very difficult.
- The ODDM together with a purely total Lagrange approach is not efficient since the nonlinearities are not separated from the coarse grid computation as it can be achieved in the small deformation regime.



## 9 Conclusions

This thesis is devoted to the Finite Element simulation of incremental metal forming processes. Several methods have been proposed to accelerate the Finite Element simulation which is very important to make it an attractive alternative to trying out forming strategies ‘by hand’.

A large part has been dedicated to the improvement of commonly used contact formulations, another one to the development of a partitioned solution approach with a fine Finite Element grid in the small and mobile forming zone and a coarse mesh for the global workpiece. In detail:

- The importance of correct linearization of the elastoplastic constitutive law in the context of metal forming processes has been underlined.
- Several possibilities to solve the constrained optimization problem of plasticity have been shown and compared.
- The use of the primal dual active set strategy with dual Lagrange multipliers has been studied, and its advantage over standard penalty methods has been shown in numerical examples.
- The contact algorithm allows to apply fast iterative solvers on nonlinear problems in a very efficient way.
- It can be combined with energy conservative schemes in the implicit computation of dynamical problems and it has been applied to a velocity driven, rigid plastic problem.
- The above mentioned divide-and-conquer approach has been introduced to reduce the number of unknowns in the Finite Element computation.
- The nonlinearities of the simulation, namely the contact and the elastoplastic problem have been hidden from the global computation.
- Due to clear data structures and a small interface, it turns out, that the fine grid and the coarse grid computation can be distributed to two different FE codes.
- The domain decomposition method is a dynamic one, in the sense that the fine grid is travelling with the forming zone.

- It defines in a very natural way a partitioned solution scheme which shows in combination with inexact Newton schemes good convergence and finds a very short nonlinear solution path.

Based on the results of this work, several future developments are desirable.

- The energy consistent treatment of plasticity in the regime of large deformations should be taken into account.
- It should be intended to incorporate frictional contact into the proposed contact algorithms.
- A numerical example in the context of an elastic problem showed that the fine and coarse grid are in principle independent. This result has to be extended to the elastoplastic case by improving the data transfer operators.
- It would be strongly desirable that the work is extended to the regime of large deformations.
- The proposed two-grid method should be compared to adaptive methods with hanging nodes in terms of efficiency and accuracy. Nevertheless, the ODDM should not be regarded as a method which is competing against more classical adaptive schemes. On the contrary: It should be investigated, if both methods can complete one another.

Overall, a significant speed increase has been achieved, by using several iterative solution strategies and domain decomposition methods. The proposed methods turn out to be promising and worth to be tested in the framework of a commercial FEM package to validate them for industrial applications.

First steps in this direction have already been taken, for this thesis has been also intended to make the nowadays still large gap between existing advanced numerical schemes and the needs of today's FEM software development a little bit smaller.

# Bibliography

- [1] M. Ainsworth and J.T. Oden. *A posteriori error estimation in finite element analysis*. Chichester: Wiley, 2000. 73
- [2] P. Alart and A. Curnier. A mixed formulation for frictional contact problems prone to Newton like solution methods. *Comput. Methods Appl. Mech. Engrg.*, 92:353–375, 1991. 26, 27, 35, 43
- [3] H. Aretz. *Modellierung des anisotropen Materialverhaltens von Blechen mit Hilfe der Finite-Elemente-Methode*. PhD thesis, Fakultät für Bergbau, Hüttenwesen und Geowissenschaften, RWTH Aachen, 2003. 8, 14, 16, 17
- [4] J.H. Argyris and J.S. Doltsinis. A primer on superplasticity in natural formulation. *Comput. Methods Appl. Mech. Engrg.*, 46:83–131, 1984. 65
- [5] F. Armero and E. Petocz. A new class of conserving algorithms for dynamic contact problems. *Comput. Methods Appl. Mech. Engrg.*, 158:269–300, 1998. 54
- [6] I. Babuška and T. Strouboulis. *The finite element method and its reliability*. Oxford: Clarendon Press, 2001. 73
- [7] F. Barlat and J. Lian. Plastic behaviour and stretchibility of sheet metals. Part I: A yield function for orthotropic sheets under plane stress condition. *International Journal of Plasticity*, 95:15–29, 1989. 8
- [8] K.J. Bathe. *Finite Elemente Methoden*. Springer, 1986. 116
- [9] F. Ben Belgacem and Y. Renard. Hybrid finite element methods for the Signorini problem. *Mathematics of Computation*, 72(243):1117–1145, 2003. 43
- [10] T. Belytschko, W.K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. John Wiley & Sons, 2000. 7, 36
- [11] F. Ben Belgacem, P. Hild, and P. Laborde. Approximation of the unilateral contact problem by the mortar finite element method. *C. R. Acad. Sci. Paris*, 324:123–127, 1997. 43
- [12] F. Ben Belgacem, P. Hild, and P. Laborde. Extension of the mortar finite element method to a variational inequality modeling unilateral contact. *Mathematical Models and Methods in Applied Sciences*, 9:287–303, 1999. 43

- [13] M. Benzi. Preconditioning techniques for large linear systems: a survey. *J. Comput. Phys.*, 182:418–477, 2002. 31
- [14] C. Bernardi, Y. Maday, and A.T. Patera. A new nonconforming approach to domain decomposition: the mortar element method. In H. Brezzi et al., editor, *In: Nonlinear partial differential equations and their applications*, pages 13–51. Paris, 1994. 43
- [15] M. Bischoff and E. Ramm. On the physical significance of higher order kinematic and static variables in a three-dimensional shell formulation. *Internat. J. Solids Structures*, 37:6933–6960, 2000. 53
- [16] W.L. Briggs, V.E. Henson, and S. McCormick. *A Multigrid Tutorial, 2nd Edition*. SIAM, Philadelphia, 2000. 31, 38
- [17] S. Brunssen. Modeling and simulation of elastoplastic forming processes. Technical Report 014, Universität Stuttgart, SPP 1146, 2006. 1, 4
- [18] S. Brunssen, S. Hübner, and B. Wohlmuth. Contact Dynamics with Lagrange Multipliers. In P. Wriggers and U. Nackenhorst, editors, *Computational Methods in Contact Mechanics*, volume 3 of *IUTAM Bookseries*, pages 17–32. Springer, 2007. 1, 4
- [19] S. Brunssen, F. Schmid, M. Schäfer, and B. Wohlmuth. A fast and robust method for contact problems by combining a primal-dual active set strategy and algebraic multigrid. *Internat. J. Numer. Methods Engrg.*, 69:524–543, 2007. 1, 3, 25, 27
- [20] S. Brunssen and B. Wohlmuth. An overlapping domain decomposition method for the simulation of elastoplastic incremental forming processes. Technical report, Universität Stuttgart, SPP 1146, 2007. *Internat. J. Numer. Methods Engrg.*, to appear. 1, 4
- [21] A. Bucher. *Deformationsgesetze für große elastisch-plastische Verzerrungen unter Berücksichtigung einer Substruktur*. PhD thesis, TU Chemnitz, 2001. 105
- [22] N. Büchter and E. Ramm. 3D-extension of nonlinear shell equations based on the enhanced assumed strain concept. In C. Hirsch, editor, *Computational Methods in Applied Sciences*, pages 55–62. Elsevier, 1992. 44, 53
- [23] N. Büchter, E. Ramm, and D. Roehl. Three-dimensional extension of nonlinear shell formulation based on the enhanced assumed strain concept. *Internat. J. Numer. Methods Engrg.*, 37:2551–2568, 1994. 53
- [24] G.T. Camacho and M. Ortiz. Adaptive Lagrangian modelling of ballistic penetration of metallic targets. *Comput. Methods Appl. Mech. Engrg.*, 142:3–47, 1997. 89

- 
- [25] E. Ceretti, C. Giardini, and A. Attanasio. Sheet incremental forming on CNC machines. In *Proceedings of the SheMet 2003*, pages 49–56, April 2003. 2
- [26] P.W. Christensen. A nonsmooth Newton method for elastoplastic problems. *Comput. Methods Appl. Mech. Engrg.*, 191:1189–1219, 2002. 35
- [27] P.W. Christensen and J.S. Pang. Frictional contact algorithms based on semismooth Newton methods. *Internat. J. Numer. Methods Engrg.*, 42:145–173, 1998. 35, 43
- [28] J. Chung and G.M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: The Generalized- $\alpha$  method. *J. Appl. Mech.*, 60:371–375, 1993. 44, 50
- [29] W.J. Chung, J.W. Cho, and T. Belytschko. On the dynamic effects of explicit FEM in sheet metal forming analysis. *Computational Mechanics*, 15(6):750–776, 1998. 3
- [30] T. Cichosz. Simulation von inkrementellen Blechumformprozessen. Master’s thesis, Institute of Structural Mechanics, Stuttgart, 2006. 57
- [31] A.J. Cleary, R.D. Falgout, V.E. Henson, J.E. Jones, T.A. Manteuffel, S.F. McCormick, G.N. Miranda, and J.W. Ruge. Robustness and scalability of algebraic multigrid. *SIAM J. Sci. Comput.*, 21(5):1886–1908, 2000. 27, 31
- [32] R.S. Dembo, S.C. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM J. Numer. Anal.*, 19:400–408, 1982. 99
- [33] R. Diez, U. Hindenlang, and A. Kurz. LARSTRAN user’s manual – LARSTRAN documentation. Technical report, Lasso GmbH, Stuttgart, 1996. 7, 36
- [34] Z. Dostal, A. Friedlander, and S.A. Santos. Solution of coercive and semicoercive contact problems by FETI domain decomposition. *Contemporary Mathematics*, 218, 1998. 26
- [35] A. Düster, E. Rank, G. Steinl, and W. Wunderlich. A combination of an h- and a p-version of the finite element method for elastic-plastic problems. In *Proceedings of the ECCM 1999*, September 1999. 73
- [36] S. Eckert, H. Baaser, D. Gross, and O. Scherf. A BDF2 integration method with step size control for elasto-plasticity. *Comput. Mech.*, 34:377–386, 2004. 9
- [37] S.C. Eisenstat and H.F. Walker. Choosing the forcing terms in an inexact Newton method. *SIAM J. Sci. Comput.*, 17(1):16–32, 1996. 96, 99, 102
- [38] K.A. Fischer and P. Wriggers. Frictionless 2D contact formulations for finite deformations based on the mortar method. *Comput. Mech.*, 36:226–244, 2005. 43, 76

- [39] J. Fish. The s-version of the finite element method. *Computers and Structures*, 43:539–547, 1992. 73
- [40] J. Fish and S. Markolefas. Adaptive s-method for linear elastostatics. *Comput. Methods Appl. Mech. Engrg.*, 104:363–396, 1993. 73
- [41] B. Flemisch. *Non-matching Triangulations of Curvilinear Interfaces Applied to Electro-Mechanics and Elasto-Acoustics*. PhD thesis, Universität Stuttgart, Shaker Verlag, Aachen, 2007. 32
- [42] B. Flemisch, J.M. Melenk, and B. Wohlmuth. Mortar methods with curved interfaces. *Appl. Numer. Math.*, 54(3-4):339–361, 2005. 83
- [43] B. Flemisch and B. Wohlmuth. Stable Lagrange multipliers for quadrilateral meshes of curved interfaces in 3D. *Comput. Methods Appl. Mech. Engrg.*, 196(8):1589–1602, 2007. 46
- [44] M. Gee, W.A. Wall, and E. Ramm. Parallel multilevel solution of nonlinear shell structures. *Comput. Methods Appl. Mech. Engrg.*, 194:2513–2533, 2005. 53
- [45] R. Glowinski, J. He, A. Lozinski, J. Rappaz, and J. Wagner. Finite element approximations of multi-scale elliptic problems using patches of elements. *Numer. Math.*, 101:663–687, 2005. 74, 92
- [46] U.J. Görke, A. Bucher, Kreißig, and D. Michael. Ein Beitrag zur Lösung von Anfangs-Randwert-Problemen einschließlich der Materialmodellierung bei finiten elastisch-plastischen Verzerrungen mit Hilfe der FEM. Technical report, Preprint 00-09 SFB 393, TU Chemnitz, 2000. 110, 111
- [47] A.M. Habraken and S. Cescotto. An automatic remeshing technique for finite element simulation of forming processes. *Internat. J. Numer. Methods Engrg.*, 30:1503–1525, 1990. 89
- [48] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer, 1985. 31, 38
- [49] W. Han and B.D. Reddy. *Plasticity, Mathematical Theory and Numerical Analysis*. Springer, 1999. 7
- [50] S. Hartmann, S. Brunssen, E. Ramm, and B. Wohlmuth. A primal-dual active set strategy for unilateral non-linear dynamic contact problems of thin-walled structures. *Internat. J. Numer. Methods Engrg.*, 70:883–912, 2007. 1, 4, 32
- [51] S. Hartmann, G. Lühns, and P. Haupt. An efficient stress algorithm with applications in viscoplasticity and plasticity. *Internat. J. Numer. Methods Engrg.*, 40:991–1013, 1997. 111, 112
- [52] P. Hauret and P. Le Tallec. Energy-controlling time integration methods for nonlinear elastodynamics and low-velocity impact. *Comput. Methods Appl. Mech. Engrg.*, 195:4890–4916, 2006. 53, 56

- 
- [53] M. Hintermüller, K. Ito, and K. Kunisch. The primal-dual active set strategy as a semi-smooth Newton method. *SIAM Journal on Optimization*, 13(3):865–888, 2003. 26, 27, 35, 43
- [54] G. Hirt, M. Bambach, and S. Junk. Modelling of the incremental CNC sheet metal forming process. In *Proceedings of the SheMet 2003*, pages 495–502, April 2003. 2
- [55] <http://www.ecoroll.de>. 2
- [56] S. Hübner, G. Stadler, and B. Wohlmuth. A primal-dual active set algorithm for three-dimensional contact problems with Coulomb friction. *SIAM J. Sci. Comput.*, 30(2):572–596, 2008. 25
- [57] S. Hübner and B. Wohlmuth. A primal-dual active set strategy for non-linear multibody contact problems. *Comput. Methods Appl. Mech. Engrg.*, 194:3147–3166, 2005. 27, 43, 76
- [58] A. Hund and E. Ramm. Locality constraints within multiscale model for non-linear material behaviour. *Internat. J. Numer. Methods Engrg.*, 70:1613–1632, 2007. 73
- [59] Simo J. and T.A. Laursen. Augmented Lagrangian treatment of contact problems involving friction. *Comput. Struct.*, 42:97–116, 1992. 53
- [60] S. Junk. *Inkrementelle Blechumformung mit CNC Werkzeugmaschinen: Verfahrensgrenzen und Umformstrategien*. PhD thesis, Universität des Saarlandes, 2003. 2
- [61] N. Kikuchi and Y. Joon Song. Penalty/finite-element approximations of a class of unilateral problems in linear elasticity. *Quarterly of Applied Mathematics*, 39:1–22, 1981. 27
- [62] D. Klatte and B. Kummer. *Nonsmooth Equations in Optimization*, volume 60 of *Nonconvex Optimization and Its Applications*. Springer, 2002. 10
- [63] F. Klocke and S. Mader. Fundamentals of the deep rolling of compressor blades for turbo aircraft engines. In *Proceedings of the 9th international conference on shot peening*, pages 125–130, Paris, 2005. 1
- [64] D.A. Knoll and D.E. Keyes. Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *J. Comput. Phys.*, 193:357–397, 2004. 74
- [65] D.P. Koistinen and N.M. Wang. Mechanics of sheet metal forming - material behavior and deformation analysis. *Plenum Press*, 1978. 15
- [66] M. Kojić and K.J. Bathe. *Inelastic Analysis of Solids and Structures*. Springer, 2005. 113
- [67] R. Kornhuber. Monotone multigrid methods for elliptic variational inequalities, I. *Numer. Math.*, 69(2):167–184, 1994. 26

- [68] R. Kornhuber and R. Krause. Adaptive multigrid methods for Signorini's problem in linear elasticity. *CVS*, 4:9–20, 2001. 31
- [69] L. Krabbenhoft, A.V. Lyamin, S.W. Sloan, and P. Wriggers. An interior point algorithm for elastoplasticity. *Internat. J. Numer. Methods Engrg.*, 69:592–626, 2007. 20
- [70] R. Krause. *Monotone Multigrid Methods for Signorini's Problem with Friction*. PhD thesis, Freie Universität Berlin, 2001. 26, 31
- [71] R. Krause and E. Rank. Multiscale computations with a combination of the h- and p-versions of the finite element method. *Comput. Methods Appl. Mech. Engrg.*, 192:3959–3983, 2003. 73
- [72] E. Kröner. Benefits and shortcomings of the continuous theory of dislocations. *Internat. J. Solids Structures*, 38:1115–1134, 2001. 94
- [73] D. Kuhl and M.A. Crisfield. Energy conserving and decaying algorithms in non-linear structural dynamics. *Internat. J. Numer. Methods Engrg.*, 45:569–599, 1999. 50
- [74] D. Kuhl and E. Ramm. Generalized energy-momentum method for non-linear adaptive shell dynamics. *Comput. Methods Appl. Mech. Engrg.*, 178:343–366, 1999. 44, 50
- [75] T.A. Laursen. *Computational Contact and Impact Mechanics*. Springer, 2002. 25, 53
- [76] T.A. Laursen and V. Chawla. Design of energy conserving algorithms for frictionless dynamic contact problems. *Internat. J. Numer. Methods Engrg.*, 40:863–886, 1997. 54
- [77] T.A. Laursen and G.R. Love. Improved implicit integrators for transient impact problems - geometric admissibility within the conserving framework. *Internat. J. Numer. Methods Engrg.*, 53:245–274, 2002. 44, 54, 56, 63
- [78] E.H. Lee. Elastic-plastic deformation at finite strains. *J. Appl. Mech.*, 36:1–6, 1969. 106
- [79] N.S. Lee and K.J. Bathe. Error indicators and adaptive remeshing in large deformation finite element analysis. *Fin. El. in Analysis and Design*, 116:99–139, 1994. 89
- [80] S.H. Lee, J.H. Song, Y.C. Yoon, G. Zi, and T. Belytschko. Combined extended and superimposed finite element method for cracks. *Internat. J. Numer. Methods Engrg.*, 59:1119–1136, 2004. 73



- 
- [81] J.E. Marsden and T.J.R. Hughes. *Mathematical Foundations of Elasticity*. Prentice-Hall International Inc., 1983. 106
- [82] T.W. McDevitt and T.A. Laursen. A mortar finite element formulation for frictional contact problems. *Internat. J. Numer. Methods Engrg.*, 48:1525–1547, 2000. 43, 76
- [83] K. McLoughlin, A. Cognot, and E. Quigley. Dieless manufacturing of sheet metal components with non rigid support. In *Proceedings of the SheMet 2003*, pages 123–130, April 2003. 2
- [84] T. Meinders, A.H. van den Boogaard, and J. Huétink. Improvement of implicit finite element code performance in deep drawing simulations by dynamics contributions. *J. of Materials Processing Technology*, 134:413–420, 2003. 4
- [85] A. Meyer. Projected PCGM for handling hanging nodes in adaptive finite element procedures. Technical report, Preprint 99-25 SFB 393, TU Chemnitz, 1999. 89
- [86] R. Mohr, A. Menzel, and P. Steinmann. Conservation properties of galerkin-based time-stepping schemes for finite elasto-plasto-dynamics. In R. Owen, E. Onate, and B. Suarez, editors, *Computational Plasticity IX*, pages 948–951. Int. Center for Numerical Methods in Engrg., CIMNE, September 2007. 4
- [87] M. Oldenburg and L. Nilsson. The position code algorithm for contact searching. *Internat. J. Numer. Methods Engrg.*, 37:359–386, 1994. 77
- [88] V. Padmanabhan and T.A. Laursen. Surface smoothing procedure for large deformation contact analysis. *Finite Elem. Anal. Des.*, 37:173–198, 2001. 43
- [89] P. Papadopoulos and R. Taylor. A mixed formulation for the finite element solution of contact problems. *Comput. Methods Appl. Mech. Engrg.*, 94:373–389, 1992. 43, 82
- [90] J. Pebreel, P. Gosselet, and C. Rey. Multiscale analysis of structures using overlapping nonlinear patches. In E. Onate, M. Papadrakakis, and B. Schrefler, editors, *Computational Methods for Coupled Problems in Science and Engineering II*, pages 453–456. Int. Center for Numerical Methods in Engrg., CIMNE, May 2007. 74
- [91] M.A. Puso and T.A. Laursen. A 3D contact smoothing method using Gregory patches. *Internat. J. Numer. Methods Engrg.*, 54:1161–1194, 2002. 43
- [92] M.A. Puso and T.A. Laursen. A mortar segment-to-segment contact method for large deformation solid mechanics. *Comput. Methods Appl. Mech. Engrg.*, 193:601–629, 2004. 25, 43, 76
- [93] M.A. Puso and T.A. Laursen. A mortar segment-to-segment frictional contact method for large deformations. *Comput. Methods Appl. Mech. Engrg.*, 193:4891–4913, 2004. 43

- [94] E. Rank, A. Düster, D. Ledentsov, I. Heinle, W. Volk, and M. Wagner. A unified approach to multiscale simulation and dimensional adaptivity: basic principles and industrial applications. In E. Onate, M. Papadrakakis, and B. Schrefler, editors, *Computational Methods for Coupled Problems in Science and Engineering II*, pages 649–652. Int. Center for Numerical Methods in Engrg., CIMNE, May 2007. 73
- [95] J.W. Ruge and K. Stüben. *Multigrid Methods, Frontiers in Applied Mathematics*, volume 3, chapter Algebraic Multigrid, pages 73–130. SIAM, Philadelphia, 1987. 31
- [96] Y. Saad. *Iterative Methods for Sparse Linear Systems, 2nd Edition*. SIAM, Philadelphia, 2003. 94, 97
- [97] J.C. Simo and T.J.R. Hughes. *Computational Inelasticity*. Springer, 1998. 7, 9, 10, 13, 14, 23
- [98] J.C. Simo and M. Ortiz. A unified approach to finite deformation elastoplastic analysis based on the use of hyperelastic constitutive equations. *Comput. Methods Appl. Mech. Engrg.*, 49:221–245, 1985. 14
- [99] J.C. Simo and R.L. Taylor. Consistent tangent operators for rate independent elasto-plasticity. *Comput. Methods Appl. Mech. Engrg.*, 48:101–118, 1985. 9
- [100] M. Stadler and G.A. Holzapfel. Subdivision schemes for smooth contact surfaces of arbitrary mesh topology in 3D. *Internat. J. Numer. Methods Engrg.*, 60:1161–1195, 2004. 43
- [101] E. Stein and M. Rüter. Finite element methods for elasticity with error-controlled discretization and model adaptivity. In E. Stein, R. de Borst, and J.R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 2, chapter 2, pages 5–58. John Wiley & Sons, 2004. 73
- [102] K. Stüben. A review of algebraic multigrid. *J. Comput. Appl. Math.*, 128:281–309, 2001. 27, 31
- [103] M. Tabbara, T. Blacker, and T. Belytschko. Finite element derivative recovery by moving least square interpolants. *Comput. Methods Appl. Mech. Engrg.*, 117:211–223, 1994. 89
- [104] R. Verfürth. *A review of a posteriori error estimation and adaptive mesh-refinement techniques*. Wiley–Teubner Series Advances in Numerical Mathematics. Wiley–Teubner, Chichester, Stuttgart, 1996. 73
- [105] C. Wieners. Nonlinear solution methods for infinitesimal perfect plasticity. *Z. Angew. Math. Mech.*, 87(8):643–660, 2007. 17, 19, 73, 104
- [106] K. Willner. *Kontinuums- und Kontaktmechanik*. Springer, 2003. 25, 39

- [107] B. Wohlmuth. A mortar finite element method using dual spaces for the Lagrange multiplier. *SIAM J. Numer. Anal.*, 38:989–1012, 2000. 43
- [108] B. Wohlmuth. Discretization techniques and iterative solvers based on domain decomposition. *Lecture Notes in Computational Science and Engineering*, 17, 2001. Springer. 27, 32
- [109] P. Wriggers. *Nichtlineare Finite-Element-Methoden*. Springer, 2001. 7
- [110] P. Wriggers. *Computational Contact Mechanics*. John Wiley & Sons, 2002. 25
- [111] P. Wriggers and K. Fischer. Recent new developments in contact mechanics. In *Proceedings of the 4th European LS-DYNA Users Conference, Ulm*, 2003. 25
- [112] P. Wriggers and G. Zavarise. *Encyclopedia of Computational Mechanics*, volume 2, chapter Computational Contact Mechanics. John Wiley & Sons, 2004. 25
- [113] S.P. Xiao and T. Belytschko. A bridging domain method for coupling continua with molecular dynamics. *Comput. Methods Appl. Mech. Engrg.*, 193:1645–1669, 2004. 74
- [114] B. Yang and T.A. Laursen. A contact searching algorithm including bounding volume trees applied to finite sliding mortar formulations. *Computational Mechanics*, 41:189–205, 2008. 77
- [115] B. Yang, T.A. Laursen, and X.N. Meng. Two dimensional mortar contact methods for large deformation frictional sliding. *Internat. J. Numer. Methods Engrg.*, 62:1183–1225, 2005. 29, 43, 118
- [116] Z. Yue and D.H. Robbins. Adaptive superposition of finite element meshes in elastodynamic problems. *Internat. J. Numer. Methods Engrg.*, 63(11):1604–1635, 2005. 73