Segment-to-segment algorithm for finite volume mechanical contact simulations

Batistić, Ivan

Doctoral thesis / Disertacija

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:395801

Rights / Prava: In copyright/Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-02-03**

University of Zagreb

Faculty of Mechanical Engineering and Naval Architecture

Ivan Batistić

Segment-to-Segment Algorithm for Finite Volume Mechanical Contact Simulations

DOCTORAL THESIS

Zagreb, 2022

University of Zagreb

Faculty of Mechanical Engineering and Naval Architecture

Ivan Batistić

Segment-to-Segment Algorithm for Finite Volume Mechanical Contact Simulations

DOCTORAL THESIS

Supervisor: Prof. Željko Tuković, PhD

Zagreb, 2022

Sveučilište u Zagrebu

Fakultet strojarstva i brodogradnje

Ivan Batistić

Segmentni postupak za računanje mehaničkoga kontakta primjenom metode kontrolnih volumena

DOKTORSKI RAD

Mentor: prof. dr. sc. Željko Tuković

Zagreb, 2022.

BIBLIOGRAPHY DATA

Dedicated to my family

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Željko Tuković for his support and guidance. His expertise, as well as personal qualities, enabled me to prosper professionally as well as personally. I am thankful for every discussion we had, they taught me how to approach the problem, which is the most valuable thing that this journey provided to me.

Sincere thanks to Prof. Philip Cardiff for continuous support, interest and valuable comments. Many thanks to Damjan Čakmak for many fruitful discussions and helpful advice.

Many thanks to all of my friends and colleagues — all of them contributed to my research journey in some way and being afraid I might forget someone, thank you all.

Last but not least, thanks to my family for their unconditional support throughout my education. Coming home and spending time with my family after a long period in Zagreb was truly refreshing and fulfilling. Special thanks to my fiance Antonija for her unwavering support, understanding and patience.

Abstract

The highly nonlinear nature of the mechanical contact interaction, commonly coupled with other sources of nonlinearities, presents a challenging task for numerical modelling. With the increasing need for modelling multiphysics problems involving contact phenomena, the Finite Volume Method has shown to be capable of solving frictional contact problems with material and geometric nonlinearities. Since contact interaction plays an important role in many applications, the development of more accurate and efficient contact treatment procedures is essential.

The principal objective of this thesis's is to develop a finite volume version of the segment-to-segment contact-force calculation algorithm in which normal contact pressure, governed by a penalty law, is integrated across the non-conformal contact interface. Compared to the currently available pointwise contact algorithm, it is shown that such an approach allows for higher accuracy while removing or reducing some drawbacks of the pointwise calculation approach. Various aspects of the penalty-based Neumann-Neumann coupling are discussed and a suitable contact detection algorithm is proposed. To improve efficiency, an implicit version of the Neumann-Neumann coupling is derived. Based on the presented results, such an approach has shown to be a promising direction for future development. Additionally, potential benefits of surface smoothing are analysed by incorporating the Nagata surface interpolation within the pointwise contact algorithm.

The proposed contact algorithm is extensively tested using various benchmark problems as well as industrial grade metal forming problems. The reported results show a good agreement with available analytical or numerical results.

The thesis relies on the advance finite volume structural solver capable of describing hyperelastic and hyperelastoplastic deformation of bodies in contact. The description of such a class of structural solvers is given in detail. The implementation of the proposed procedures is conducted within the `foam-extend` software package, a community-driven fork of the open source OpenFOAM software.

Keywords:

Finite Volume Method, Numerical Analysis, Mechanical Contact, Contact Algorithm, Segment-to-Segment Algorithm, OpenFOAM

Prošireni sažetak

Ovaj rad predstavlja numerički model za računanje problema mehaničkoga kontakta, primjenom metode kontrolnih volumena. Razmatra se izotermalni kontakt s trenjem uz pretpostavku velikih pomaka i deformacija te velikoga međusobnog klizanja kontaktnih površina.

I Uvod

Problem mehaničkoga kontakta dvaju ili više deformabilnih tijela predmet je istraživanja kontaktne mehanike, koja predstavlja zasebnu disciplinu teorije elastičnosti. Zbog visoke složenosti kontakta, razvijena analitička rješenja kontaktne mehanike primjenjiva su samo za jednostavnije probleme kontakta, usprkos velikim naporima uloženima od strane mnogih istraživača. Iz navedenoga razloga numeričke metode predstavljaju alat visoke praktične važnosti jer omogućavaju rješavanje izrazito složenih problema kontakta uz potpuni uvid u polja naprezanja i deformacija.

Primjena metode kontrolnih volumena u području mehanike deformabilnih tijela započela je prije tri desetljeća. Danas se metoda kontrolnih volumena uspješno primjenjuje na razne linearne i nelinearne probleme mehanike deformabilnih tijela, te je u tome području prepoznata kao valjana alternativa znatno razvijenijoj metodi konačnih elemenata. Razvoju metode kontrolnih volumena uvelike su pomogla primijenjena znanja iz problematike mehanike fluida, a zbog sve većih zahtjeva za modeliranjem zahtjevnih multifizičkih procesa, očekuje se daljnji razvoj i njihova primjena na sve složenije probleme mehanike deformabilnih tijela. Prema tome, razvoj numeričkih postupaka za točno i učinkovito modeliranje mehaničkoga kontakta deformabilnih tijela od presudne je važnosti.

Prva primjena metode kontrolnih volumena na probleme kontakta osnivala se na eksplicitnoj Dirichelt-Neumannovoj sprezi kontaktnih granica, kakva se koristi u problemima interakcije fluida i deformabilnih tijela. Zbog određenih nedostataka takvoga pristupa, daljnji razvoj je usmjeren na eksplicitnu Neumann-Neumannovu spregu koja se osniva na metodi penalizacije. Takav pristup je korišten u [1, 2] gdje se predlaže postupak, koji se oslanja na GGI (eng. Generalised Grid Inter-

face) interpolaciju [3] i izračun kontaktnog tlaka u čvorovima (vrhovima) mreže. S obzirom na to da se pokazalo da postupak temeljen na Neumann-Neumannovoj sprezi može pružiti zadovoljavajuću točnost i robusnost prilikom rješavanja zahtjevnih kontaktnih problema, isti je prihvaćen u ovome radu s ciljem daljnjega unaprjeđenja robusnosti, točnosti i učinkovitosti putem razvoja naprednijih postupaka za obradu kontaktnih granica.

II   Matematički model

Matematička formulacija problema kontakta zasniva se na načelima klasične mehanike kontinuuma. Za opis deformacija i gibanja tijela u kontaktu koriste se relacije kinematike kontinuuma. Kao posljedica kontaktne interakcije pojavljuju se naprezanja na kontaktnim plohama i u unutrašnjosti tijela. U radu je dat pregled osnovnih veličina mehanike kontinuuma, koje se koriste i za Lagrangeovu inkrementalnu formulaciju integralnoga oblika zakona održanja količine gibanja. Veza naprezanja i deformacija opisana je pomoću hiperelastičnoga i hiperelastično-plastičnog materijalnoga modela. Nadalje, dat je opis kontaktnih kinematičkih veličina, koje se koriste za definiciju kontaktnih uvjeta. Interakcija tijela u kontaktu opisana je kontaktnim uvjetima, koji su podijeljeni na normalne i tangencijalne. Normalni kontaktni uvjeti, poznati kao Hertz-Signorini-Moreauovi ili Karush-Kuhn-Tuckerovi uvjeti, impliciraju tri temeljne zakonitosti koje moraju biti ispunjene, a to su nemogućnost prodiranja tijela i ostvarivanja vlačnoga kontaktnog opterećenja te komplementarnost opterećenja i kontaktnoga statusa.

Karush-Kuhn-Tuckerovi uvjeti koriste se i za formulaciju tangencijalnih uvjeta prema odabranom zakonu trenja. U radu je korišten Coulombov zakon trenja za određivanje smjera i iznosa sile trenja prema stanju prijanjanja i klizanja.

III   Numerički model

Diskretizacija metodom kontrolnih volumena omogućava tretman kontaktne granice pomoću nelinearnoga Neumannova ili Dirichletova graničnog uvjeta, ne oviseći pritom direktno o odabranome materijalnom modelu ili diskretizacijskim shemama. To je omogućilo korištenje naprednoga strukturnog rješavača, predloženog u [2], namijenjenog za opisivanje velikih elastičnih i elastoplastičnih deformacija. U radu je detaljno opisana takva klasa rješavača, koja se temelji na poliedarskoj metodi

kontrolnih volumena drugoga reda točnosti i na odvojenome postupku rješavanja. Time su pokrivene ostale moguće inačice rješavača, za koje razvijene kontaktne procedure mogu biti primijenjene. Diskretizacija prostornih integrala jednadžbe količine gibanja, prikazana je pojedinačno te je pojašnjena implementacija osnovnih graničnih uvjeta. Posebna pozornost je data postupku rješavanja i postupku pomicanja čvorova mreže. Kako bi se ukazalo na moguće nedostatke, dat je kratak osvrt na alternativne metode diskretizacije i postupke rješavanja.

U sklopu eksplicitne Neumann-Neumannove sprege predložen je segmentni postupak za računanje kontaktnih sila na temelju metode penalizacije. Segmentni se postupak temelji na integraciji kontaktnoga tlaka na nekomfornome kontaktnom sučelju. Naspram čvornoga postupka pokazano je da takav segmentni pristup rezultira većom točnošću, manjim oscilacijama kontaktne sile i simetričnim tretmanom kontakta. Kako bi se povećala efikasnost postupka rješavanja, razmotrena je implicitna Neumann-Neumannova sprega, koja je implementirana pomoću novorazvijenoga segmentnog postupka. Rezultati pokazuju kako implicitna sprega može značajno poboljšati efikasnost postupka rješavanja — uz očuvanje iste razine točnosti. Međutim, potreban je daljnji razvoj implicitne sprege kako bi ona imala istu robusnost kao i eksplicitna izvedba. Razmotrena je i mogućnost zaglađivanja kontaktne granice te je predložena metoda koja se osniva na Nagatovoj interpolaciji, koja po svojim karakteristikama najviše odgovara korištenju u sklopu metode kontrolnih volumena. Zaglađivanje je implementirano u sklopu čvornoga postupka te su pokazane moguće prednosti korištenja takvoga pristupa.

IV   Rezultati

U svrhu detaljne analize robusnosti i točnosti predloženoga kontaktnog algoritma, algoritam je testiran na brojnim verifikacijskim primjerima. Dobiveni rezultati uspoređeni su s analitičkim i referentnim numeričkim rezultatima dostupnima u obrađenoj literaturi. Naglasak je dat na rezoluciju proračunske mreže koja u slučaju niske rezolucije može imati negativan utjecaj na robusnost i točnost kontaktnoga algoritma. Iz navedenoga razloga, mnogi verifikacijski primjeri riješeni su s niskom rezolucijom proračunske mreže i s različitim rezolucijama mreže. U prvome setu verifikacijskih primjera razmatraju se problemi s malim deformacijama te je pokazano vrlo dobro slaganje s dostupnim analitičkim rješenjima. U drugome setu verifikacijskih primjera, razmotreni su primjeri iz literature koji su posebno osmi-

šljeni na način da je za njihovo uspješno rješavanje potreban kontaktni algoritam visoke robusnosti i točnosti. Dobiveni rezultati ukazuju da je primjere moguće riješiti uz zadovoljavajuće odstupanje od referentnih vrijednosti dobivenih metodom konačnih elemenata. Konačno, treći skup verifikacijskih primjera uključuje zahtjevne probleme obrade metala deformiranjem, uključujući vučenje žice, valjanje žice i kompaktiranje snopa žica.

V   Zaključak

Izrazito nelinearna priroda mehaničnoga kontakta, koja često dolazi u sklopu s materijalnim i geometrijskim nelinearnostima, predstavlja izrazito zahtjevan zadatak za numeričku analizu bilo koje vrste. U ovome je radu detaljno prikazano kako metoda kontrolnih volumena može uspješno riješiti zahtjevne probleme mehaničkoga kontakta, pružajući pritom zadovoljavajuću razinu točnosti i robusnosti. Razvojem novih procedura za obradu kontaktnih granica ostvaren je napredak u pogledu točnosti, međutim treba naglasiti kako je daljnji razvoj nužan i kako prostora za napredak ima. Također, kako bi se povećala primjenjivost metode, potrebno je razmotriti teme poput trošenja, samokontakta ili termo-mehaničkog kontakta, koje još uvijek nisu obrađene i zahtijevaju posebnu pažnju u daljnjemu razvoju.

Ključne riječi:
metoda kontrolnih volumena, numerička analiza, mehanički kontakt, segmentni algoritam, OpenFOAM

Contents

Contents

List of Figures

LIST OF FIGURES

List of Tables

Nomenclature

Throughout the thesis, vector quantities are represented with lowercase boldface symbols, second-order tensor quantities with uppercase boldface symbols and scalar quantities with normal symbols. Should be noted that mentioned convection is not strictly followed for some quantities.

Latin letters

b	Contact width
$\mathbf{b}$	Source term or left Cauchy-Green deformation tensor
$\mathbf{c}$	Coefficient vector
$\mathbf{C}$	Right Cauchy-Green deformation tensor
e	Extrusion distance
E	Young's modulus
E_c	Contact Young's modulus
$\mathbf{e}$	Eulerian-Almansi strain tensor
$\mathbf{E}$	Green-Lagrange strain tensor
$\mathbf{f}$	Relative deformation gradient
$\mathbf{F}$	Deformation gradient
$\mathbf{F^e}$	Elastic deformation gradient
$\mathbf{F^p}$	Plastic deformation gradient
f^{scale}	Scale factor
G	Shear modulus
g_n	Normal gap distance
$\mathbf{g}_t$	Tangential relative sliding
$\mathbf{I}$	Second-order identity tensor
j	Relative deformation gradient determinant
J	Deformation gradient determinant
K	Bulk modulus
$\mathbf{k}$	non-orthogonal vector component in orthogonal correction
$\mathbf{n}$	Outward pointing unit normal
$\mathbf{P}$	First Piola-Kirchhoff stress tensor

p_0	Peak contact pressure
p_n	Normal contact pressure
$\mathbf{p}_t$	Tangential contact pressure
$\mathbf{R}$	Rotational tensor
$\mathbf{r}$	Position vector
R_r	Relative radius of curvature
$\mathbf{S}$	Second Piola-Kirchhoff stress tensor
t	Time
$\mathbf{T}$	Torque vector
$\mathbf{t}$	Cauchy traction vector
$\mathbf{u}$	Displacement vector
$\mathbf{U}$	Right stretch tensor
$\mathbf{v}$	Velocity vector or left stretch tensor
$\mathbf{x}$	Position vector
$\mathcal{X}$	Configuration mapping

Greek letters

α	Under-relaxation factor
β	Relative area in contact
Δ	Quantity increment
$\boldsymbol{\Delta}$	Orthogonal vector component in orthogonal correction
$\boldsymbol{\Gamma}$	Surface area vector
γ	Consistency parameter (plastic multiplier)
λ	Lame's first parameter
μ	Shear modulus or coefficient of friction
ν	Poisson's ratio
ω	Weighting factor
Ω	Volume
ρ	Density
$\boldsymbol{\sigma}$	Cauchy stress tensor
$\sum$	Summation operator
$\boldsymbol{\tau}$	Kirchhoff stress tensor
θ	Angle
ε_n	Normal penalty parameter
ε_t	Tangential penalty parameter

Superscripts

$\phi^{(i)}$ Body index

ϕ^{-1} Inverse of a second order tensor

$\tilde{\phi}$ Normalised quantity

ϕ^* Previous iteration value

$\dot{\phi}$ First time derivative

$\phi^{(m)}$ Time level

ϕ^{T} Transpose operator

Subscripts

ϕ_b Value at boundary

ϕ_D Diagonal tensor components

ϕ_{LU} Off-diagonal tensor components   -

ϕ_m Master side quantity

ϕ_n Normal component

ϕ_o Quantity referred to original configuration

ϕ_s Slave side quantity

ϕ_t Tangential component

ϕ_u Quantity referred to updated configuration

Other

$|\phi|$ Absolute value of a scalar

$||\phi||$ Euclidean norm of a vector

$\det(\phi)$ Determinant of a tensor

$\phi_a \cdot \phi_b$ Scalar (dot) product

$\phi_a : \phi_b$ Double dot product

$\langle \phi \rangle$ Macaulay's brackets

∇ Hamilton operator (nabla operator)

$\mathrm{tr}(\phi)$ Trace of a tensor

$\phi_a \times \phi_b$ Vector (cross) product

1. Introduction

The application of the implicit cell-centred (collocated) finite volume method
(FVM) on the solid mechanics' problems began in the late 1980s and early 1990s
with the seminal works [11,12]. At that point, the FVM was already well established
and dominant method in the area of computational fluid dynamics. This fact was
crucial for the subsequent development in the area of solid mechanics due to the
acquired knowledge in solving highly non-linear fluid dynamics problems. In the
area of solid mechanics, the FVM does not enjoy the popularity as the widespread
finite element method (FEM); however, lately it has been attracting more attention
and it is being recognised as a viable alternative. Moreover, due to its underlying
simplicity and strong conservative nature, it is a very appealing numerical method
for engineers.

While entering the area of solid mechanics, the complexity of considered prob-
lems was gradually increasing. Consequently, the first encounter with mechanical
contact problems came ten years after the seminal application. Subsequent develop-
ment shows that the FVM can handle large deformation frictional contact problems.
Nevertheless, the development of the FVM contact algorithms is still at its incep-
tion and many modelling aspects are still not tackled. Furthermore, accurately
describing the highly nonlinear nature of contact phenomena, commonly coupled
with other sources of nonlinearity, is a challenging task that necessitates continuous
development and significant effort from the community. The best example is the
FEM, in which the search for more robust, efficient and accurate algorithms lasts
for over five decades.

Contact problems can be analysed through different space and time scales, de-
pending on the context and the related area of research. Due to its multiscale and
multiphysics nature, contact may be accompanied by different interface phenomena
such as lubrication, friction, wear, adhesion and heat or electric transfer. These
complex phenomena and their mutual interaction are the subject of research in the
interdisciplinary branch called tribology. Generally, solid mechanics considers the
mechanical aspects of contact interaction using the macroscopic perspective and
classical continuum formulations.

Numerical modelling of contact problems is of great relevance since the majority

of engineering problems include some type of contact interaction. Some examples in which contact analysis plays a significant role in the design of optimised, safer or lighter products are metal forming processes, gearboxes and bearings, braking systems, tire-road and wheel-rail contact, crash tests, etc. The analytical solutions of contact mechanics, built upon the original work of Hertz [13], are restricted to problems with simple geometry and linear-elastic contacting bodies [14]. Therefore, for better understanding of interactions in such complex physics systems numerical and experimental methods are the only tools available for engineers. As the applicability of experimental procedures is significantly less practical, the numerical method remains the only tool available. Hence, the development of contact algorithms for the FVM is of great importance as these algorithms will lead to further extensions and applications of the method. Moreover, with increasing capabilities in contact modelling, the method can strengthen the assigned status of a viable alternative method for solid mechanics and contact mechanics problems [15].

1.1. Previous and Related Studies

Heretofore the implicit cell-centred FVM for unstructured polyhedral grids has been successfully applied to a wide variety of solid mechanics and multi-physics problems, including linear-elasticity [12, 16–18], incompressible materials [19, 20], anisotropic materials [21–23], multi-materials [24], plasticity [2, 25], fluid-solid interaction [26–28], fracture mechanics [29–33] and others; see [4] for a detailed and comprehensive survey. Apart from the cell-centred grid arrangement, it should be noted that other grid arrangements (face-centred, vertex-centred, staggered-grid) have also been successfully applied to solid mechanics problems [34–41]. The majority of FVM applications on solid mechanics problems rely on the memory-efficient segregated solution procedure coupled with a deferred correction algorithm. Some recent development shows that the block-coupled solution procedure offers impressive improvements in efficiency [42, 43]; however, it has been developed only for linear-elastic solids and efforts are being made to broaden its application to nonlinear material models too [44]. Similarly, efficiency improvement can be achieved by employing the geometric multigrid procedure [21, 29, 45] or Aitken acceleration [46, 47]. Additionally, instead of the second-order, [48] proposed the fourth-order accurate equation discretion on 2D structured Cartesian grids.

Concerning contact modelling, the first developments took place with the need to

model adhesive joints [49]. These developments laid an initial foundation for further progress which will emerge from the need to simulate biomechanics problems [50,51]. In the overall development so far, three different approaches can be distinguished:

- Inspired by interface coupling procedures applied in dealing with fluid-solid interaction problems, the first contact treatment method was implemented using the Dirichlet-Neumann (DN) coupling procedure [49]. Coupling is achieved by treating one of the contact boundaries using the DN (mixed) boundary condition, while the adjacent boundary is treated using the Neumann (fixed gradient) boundary condition. The mixed boundary condition specifies the value of the normal component of the displacement, whereas the displacement gradient is defined for the tangential direction. The direction split is achieved using the local value of the boundary normal whereas active or inactive contact status is distinguished using a value fraction. The data transfer for the non-conformal interface discretisation is achieved using the two-step weighted inverse distance interpolation. The DN coupling procedure was initially developed for quasi-static 2D frictionless contact problems with linear-elastic solids [49,52], and it was later extended to consider inertial effects [53,54] and 3D problems [55]. Although it was a very efficient procedure, the lack of robustness and accuracy in solving 3D frictional contact problems redirected further development towards the Neumann-Neumann coupling procedure, described in the next paragraph.

- The penalty based Neumann-Neumann (NN) coupling procedure was proposed by [1,50]. Initially, the procedure was developed for 3D frictionless contact problems with linear-elastic solids. Subsequently, it was extended to consider large deformation and large sliding contact problems with Coulomb friction [2]. Moreover, in conjunction with the Finite Area Method (FAM), it was successfully used to model the lubricated contact [56]. The NN coupling procedure relies on the generalised grid interface (GGI) interpolation procedure [3] to transmit contact stresses, calculated on the slave side, to the master side. The normal component of the contact traction is calculated in a pointwise manner and subsequently interpolated at face centres. The tangential (frictional) component is calculated directly at face centres using transmitted (interpolated) values of the displacement field from the master surface. To ensure convergence, the explicitly calculated contact traction is under-relaxed before the update, performed in a deferred correction manner. Penalty based calculation of the contact traction ensures the robustness of the overall procedure; therefore, this is currently the only procedure proven to

be capable of solving complex contact problems with a high degree of nonlinearities involved. Moreover, it should be mentioned that it is publicly available as a part of the OpenFOAM based toolbox `solids4Foam` [46, 57].

- The segregated solution procedure allows for great flexibility in the nonlinearity treatment, thus the contact coupling is explicitly updated during the iterative procedure. As a consequence of the explicit update, the overall efficiency can be impaired. In other words, the resolution of the inter-equation coupling is faster than the resolution of the contact coupling. This led to further developments towards the implicit treatment of contact coupling. In [58] an implicit treatment of the contact boundary is derived using the methodology borrowed from the multi-material interface treatment [24]. The contact traction is decomposed, and the normal component is implicitly treated via an equation derived using the equality of normal displacement and action-reaction principle at the contact interface. The tangential component is explicitly calculated using Coulomb's law. The handling of the non-conformal discretisation is achieved using arbitrary mesh interpolation (AMI) [59]. Treatment of faces in partial contact is achieved using a blending coefficient defined by sigmoid function. From the application to the numerical example, it is shown that the procedure is capable solving force-loading contact problems and thus offering a significant improvement in efficiency compared to penalty-based explicit coupling. However, only contact problems with linear elastic bodies (material and geometric linear) are considered. Further investigation into the applicability to large deformation contact problems remains to be carried out.

The above-listed approaches of contact treatment are intended for the cell-centred FV method. In addition, some limited work has been done regarding contact treatment within the vertex-centred FVM [60, 61] and within the Godunov-type cell-centred FVM [62]. Altogether, it can be noticed that some limited work has been done regarding the contact modelling in the FVM. Still, each of the proposed approaches has a fairly limited number of publications. In most cases, contact modelling is only a part of the proposed discretisation and solution procedure, thus limited attention is given to the description of the applied contact algorithm.

Contrary to the FVM literature, the FEM literature regarding the contact modelling is very abundant and significant efforts are continuously being made in this field. As a result, the whole branch of the computational mechanics named computational contact mechanics is established using the FEM [5, 63, 64]. Generally,

in the implicit FEM contact analysis, we can distinguish two main aspects which are responsible for the enforcement of contact constraints and the transfer of loads from one contact surface to another. For the enforcement of the contact constraints, mainly three methods are used: the penalty method, the Lagrangian method and the augmented Lagrangian method. The second aspect deals with the linkage between contact surfaces by applying contact discretisation. The result of the contact discretisation is the construction of abstract contact elements, for which three different approaches are mostly used. The simplest approach is the node-to-node (NTN) approach, which is suitable only for problems with conformal discretisation, therefore its application is restricted to problems with small deformations. The second approach is the node-to-segment (NTS) approach [65, 66]. It is the first approach applicable for finite deformation and large sliding problems; therefore, it is widely adopted in many codes, despite of its stability problems and punctuality problems as regards the calculation of the transfer of the load at non-conformal interfaces [67]. The third approach, the most complex and advanced one, is the segment-to-segment (STS) approach [68, 69], which is also applicable to finite deformation and large sliding problems. In terms of accuracy and robustness, the STS approach coupled with the mortar method is currently the most advanced FEM contact procedure [70–72]. As stated at the beginning of the paragraph, the literature is extremely broad and this brief overview is only scratching the surface of the years of development which is out of the scope of this thesis.

Unfortunately, the direct incorporation of the FEM contact algorithms into the finite volume framework is not possible as they do not share discretisation and solution techniques. In short, the FVM starts from the strong form of the governing equations and the discretisation procedure which is independent of the cell shape result in second-order accurate and locally conservative discretisation. The resulting system of equations, convenient for the efficient application of iterative solvers, is solved using the segregated solution algorithm which resolves nonlinearities with the linear convergence rate using outer Picard/Fixed-Point iterations. In contrast, the FEM starts from the weak form of the governing equations and the discretisation procedure which depends on the cell shape result in globally conservative discretisation. The discretisation procedure results in a densely populated coefficient matrix requiring the application of direct linear solvers. Further, the resolution of nonlinearities is possible with the use of quadratic convergence rate by employing the full or modified Newton-Raphson loop. Finally, contact is treated through an additional

contact term in the energy functional, whereas in the FVM it is treated as a special class of the nonlinear boundary condition. Nonetheless, both numerical methods resolve contact via incremental and iterative contact algorithms, which may share the same fundamental idea.

1.2. Present Contributions

This thesis contributes to the field of numerical contact modelling using the finite volume method. Specifically, the scientific contribution can be summarised as:

- A novel penalty-based contact algorithm, applicable to finite deformation and to large sliding contact problems, is introduced. Contact algorithm, named the segment-to-segment algorithm, is based on the integration of contact pressure across a non-conformal contact interface, thus sharing the same fundamental idea as the method of the same name in the FEM [68–72]. The main advantages of the proposed algorithm over the currently available algorithm are improved accuracy, elimination of the surface-to-surface interpolation and reduction in oscillations caused by piecewise linear surface discretisation.

- The efficiency of the overall solution procedure can be degraded because of the explicit contact resolution, especially in the case of poorly adjusted user-defined factors. To improve the efficiency, an implicit coupling procedure is proposed using the newly introduced contact algorithm. The boundary contact traction is linearised using the Picard linearisation method and the implicit equation for boundary gradient and boundary displacement are derived. Firstly, the simple 1D contact problems are considered, and the procedure is gradually extended to account for non-conformal discretisation, boundary non-orthogonality and material nonlinearities.

- The piecewise linear discretisation of the contact boundary is only an approximation of the initially smooth contact boundary. Accordingly, the coarse discretisation of the curved contact boundary results in a less accurate distribution of contact pressure. Also, the non-smoothness of the contact boundary may cause unphysical oscillations of the contact pressure, and it can negatively affect the convergence rate. To remedy this, smoothing techniques are considered, and a surface smoothing technique using the Nagata interpolation [73,74] is implemented within the pointwise contact algorithm.

- The contact detection within the framework of the finite volume segregated solution procedure is discussed and a contact detection algorithm based on a object-oriented bounding box and the advancing front technique is proposed.

The numerical implementation is conducted in a metal forming software package based on the open-source C++ library `foam-extend` which is a community-driven fork of the OpenFOAM library for general computational continuum mechanics [75]. Consequently, existing features as well as the advanced structural solver previously published in [2], are reused within this thesis and supplemented with new contact modules proposed in this thesis.

1.3. Thesis Outline

The remainder of the thesis is organised into interrelated chapters which are designed to be more or less self-sufficient:

- Chapter 2 gives a continuum description of the large deformation frictional contact problem by summarising relevant quantities of the nonlinear solid mechanics and contact mechanics. The governing and constitutive equations are stated, followed by the formulation of nonlinear contact conditions. Contact conditions, formulated per fundamental contact laws, are represented as a set of constraints for normal and tangential directions.

- Chapter 3 describes the FV framework for which contact procedures developed in this thesis are intended. The framework is based on the implicit cell-centred unstructured FV method with a segregated solution procedure coupled with a deferred correction algorithm. The equation discretisation starts from the strong integral form of the governing equations and contact is treated as a special case of nonlinear boundary condition, wherefore it is discussed separately in Chapter 4. The resulting discretisation coefficients, used in this thesis, are given, while the equation discretisation is presented in a general manner. In order to show the limitations and drawbacks, alternative solution procedures as well as discretisation strategies are discussed.

- Chapter 4 is devoted to contact modelling, thereby it represents the core of this thesis. Firstly, the difficulties regarding contact modelling are discussed and the penalty regularisation of contact conditions is explained. The point-wise contact algorithm from [1,2] is presented and possible modifications are

considered. Next, the developed segment-to-segment algorithm is introduced, followed by an outline of the numerical implementation and a description of its features and advantages. Further, to improve the overall efficiency, the implicit implementation of the proposed contact algorithm is presented. Also, as a very convenient feature, surface smoothing is considered, and smoothing using the Nagata interpolation is proposed in combination with the point-wise contact algorithm. Lastly, contact detection within the FV framework is discussed and a contact detection algorithm is proposed.

- Chapter 5 is intended to test the robustness and accuracy of the developed contact procedure by using some of the popular benchmark problems from the computational contact mechanics literature. The chapter is divided into three sections. The first section deals with small deformation contact problems, thus results are compared with the available analytical solutions. The second section refers to contact problems with material and geometric nonlinearities for which results from the literature are used for comparison. In the last section, the applicability to industrial grade problems is shown by solving metal forming problems. Generally, the capabilities of the FV method in solving contact problems have not been thoroughly investigated in the current literature, thus this chapter intends to fulfil this gap. Moreover, by giving a suitable description of the benchmark problems, this chapter can be especially useful to new researchers and further development of FV contact algorithms.

- Chapter 6 gives a summary of the thesis and provides possible directions for further work.

- Appendix A lists some of the analytical solutions of the contact mechanics used to verify numerical results of benchmark problems.

2. Continuum Description of Contact Problem

This chapter is devoted to the mathematical formulation of the mechanical contact problem. The first section deals with the fundaments of non-linear continuum solid mechanics; the issues related to kinematics, stress and governing equations are briefly presented. The second section presents the mathematical formulation of the contact problem in the framework of the continuum mechanics. The non-linear behaviour of contact is described by introducing contact kinematic and static quantities used to assemble the contact conditions. Lastly, in the penultimate section, an overview and concluding remarks are given.

2.1. Continuum Solid Mechanics

This section gives a brief overview of general concepts and principles of non-linear continuum solid mechanics. The presented kinematics, strain and stress relations together with governing equations are except for numerical aspects also required for the mathematical modelling of the contact problem, presented in the next section. Further details and a thorough introduction to the continuum solid mechanics can be found, for example, in the following textbooks [76–79].

2.1.1. Kinematics

In the continuum mechanics, the motion of a continuum body irrespective of its cause is described using kinematic quantities. Continuum body, defined as a set of continuously distributed points/particles in the three-dimensional Euclidean space, undergoes a continuous set of configurations in a given time interval. The configuration at $t = 0$ is called reference or material configuration, whereas any subsequent configuration at time t is called a current or spatial configuration, see Fig. 2.1.

When defining kinematic quantities we can distinguish two main descriptions of motion: the Lagrangian (material) description and the Eulerian (spatial) description. Although they are mathematically equivalent, from the viewpoint of the

2. Continuum Description of Contact Problem

computational solid mechanics, the logical/suitable choice is the Lagrangian description in which the current (spatial) configuration is described via configuration mapping $\mathcal{X}$ defined using reference (material) quantities.
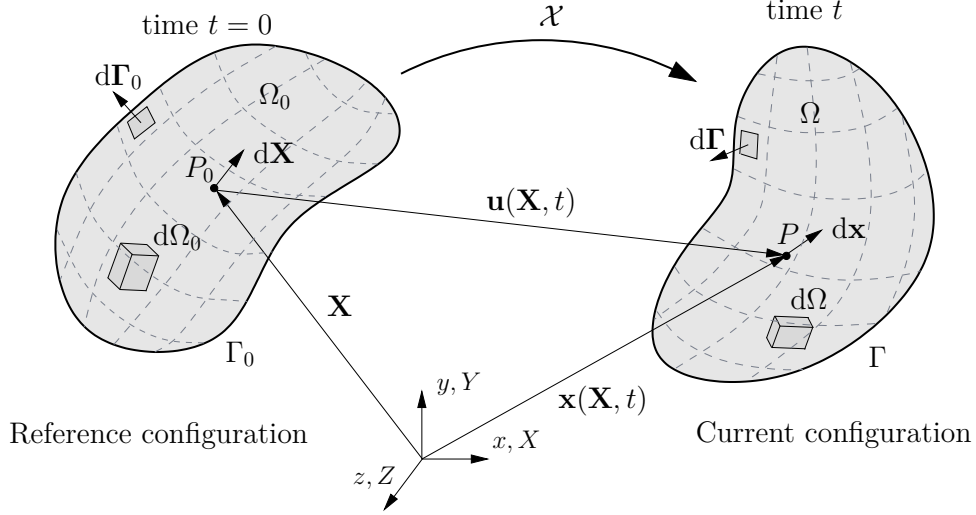
Figure 2.1. Reference and current configuration of the continuum body in motion

The displacement of material particle P_0 is derived as the difference between its associated vectors $\mathbf{x}$ and $\mathbf{X}$ for the current and initial configuration:

$$\mathbf{u}(\mathbf{X}, t) = \mathbf{x}(\mathbf{X}, t) - \mathbf{X}. \tag{2.1}$$

The displacement field does not include information regarding volume change, rotation or shape change, therefore the gradient of the deformation map $\mathcal{X}$ is introduced:

$$\mathbf{F}(\mathbf{X}, t) = \frac{\partial \mathcal{X}(\mathbf{X}, t)}{\partial \mathbf{X}} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}, \tag{2.2}$$

where $\mathbf{F}(\mathbf{X}, t)$ is the deformation gradient which maps line element $d\mathbf{X}$ to corresponding line element $d\mathbf{x}$, for the reference configuration:

$$d\mathbf{x} = \mathbf{F} \cdot d\mathbf{X}. \tag{2.3}$$

Using Eq. (2.1) the deformation gradient can be expressed in terms of displacement field:

$$\mathbf{F} = \frac{\partial \mathbf{u}(\mathbf{X}, t)}{\partial \mathbf{X}} + \frac{\partial \mathbf{X}}{\partial \mathbf{X}} = \frac{\partial \mathbf{u}(\mathbf{X}, t)}{\partial \mathbf{X}} + \mathbf{I} = (\nabla_0 \mathbf{u})^{\mathrm{T}} + \mathbf{I}, \tag{2.4}$$

where $\mathbf{I}$ is the second order identity tensor and ∇_0 indicates the gradient with respect

to material configuration. Apart from the line element, it is also possible to map other quantities, such as reference volume element $d\Omega_0$:

$$d\Omega = J \, d\Omega_0, \tag{2.5}$$

and reference area element $d\boldsymbol{\Gamma}_0$ by using Nanson's relation:

$$d\boldsymbol{\Gamma} = J \, \mathbf{F}^{-\mathrm{T}} d\boldsymbol{\Gamma}_0, \tag{2.6}$$

where J denotes the determinant of the deformation gradient $J = \det(\mathbf{F})$. This quantity (also called Jacobian) should be always positive to satisfy the impenetrability condition of body particles.

By employing the left or right polar decomposition, the deformation gradient can be decomposed into rotational (orthogonal) tensor and stretch (symmetric and positive definite) tensor:

$$\mathbf{F} = \mathbf{R} \cdot \mathbf{U} = \mathbf{v} \cdot \mathbf{R}, \tag{2.7}$$

where $\mathbf{R}$ is a rotational tensor and $\mathbf{U}$ and $\mathbf{v}$ are right and left stretch tensors, respectively. Pure rotation, represented with the rotational tensor, does not contribute to stresses; therefore, stretch tensors are used when defining strain tensors.

The finite strain tensors defined either in the material or the spatial configuration are defined as the difference of squared length of line element:

$$d\mathbf{x} \cdot d\mathbf{x} - d\mathbf{X} \cdot d\mathbf{X} = 2d\mathbf{X} \cdot \mathbf{E} \cdot d\mathbf{X} = 2d\mathbf{x} \cdot \mathbf{e} \cdot d\mathbf{x} \tag{2.8}$$

where $\mathbf{E}$ and $\mathbf{e}$ are the Green-Lagrange and Eulerian-Almansi strain tensors, respectively. For the Lagrangian description of motion, the Green-Lagrange strain tensor is a logical choice, however, it should be noted that a variety of other strain tensors exist in the literature. Using Eq. (2.3) the square length of a line element in the reference configuration is:

$$d\mathbf{x} \cdot d\mathbf{x} = d\mathbf{X} \cdot \left(\mathbf{F}^{\mathrm{T}} \cdot \mathbf{F}\right) \cdot d\mathbf{X} = d\mathbf{X} \cdot \mathbf{C} \cdot d\mathbf{X}, \tag{2.9}$$

and in the current configuration:

$$d\mathbf{X} \cdot d\mathbf{X} = d\mathbf{x} \cdot \left(\mathbf{F} \cdot \mathbf{F}^{\mathrm{T}}\right)^{-1} \cdot d\mathbf{x} = d\mathbf{x} \cdot \mathbf{b}^{-1} \cdot d\mathbf{x}, \tag{2.10}$$

where $\mathbf{C}$ and $\mathbf{b}$ are right and left Cauchy-Green tensors, respectively. By employing Eq. (2.7) it is possible to define these tensors via stretch tensors:

$$\mathbf{C} = \mathbf{F}^{\mathrm{T}} \cdot \mathbf{F} = \mathbf{U}^2, \qquad \mathbf{b} = \mathbf{F} \cdot \mathbf{F}^{\mathrm{T}} = \mathbf{v}^2. \tag{2.11}$$

Finally, using Eqs. (2.8), (2.9) and (2.10) it is possible to write strain tensors $\mathbf{E}$ and $\mathbf{e}$ as:

$$\mathbf{E} = \frac{1}{2}\left(\mathbf{C} - \mathbf{I}\right), \qquad \mathbf{e} = \frac{1}{2}\left(\mathbf{I} - \mathbf{b}^{-1}\right). \tag{2.12}$$

2.1.2. Stress

In the previously presented kinematic relations, the motion of the body is caused by the internal and boundary forces acting on the body. In the case that motion includes deformation, internal and boundary stresses arise.

For the arbitrary chosen differential area $\mathrm{d}\boldsymbol{\Gamma} = \mathbf{n}\,\mathrm{d}\Gamma$ in the current configuration, the differential resultant force $\mathrm{d}\mathbf{f}$ acting on it is:

$$\mathrm{d}\mathbf{f} = \mathbf{t}\,\mathrm{d}\Gamma, \tag{2.13}$$

where $\mathbf{t}$ is the Cauchy traction vector, which is according to the Cauchy theorem linearly mapped as:

$$\mathbf{t}\,\mathrm{d}\Gamma = \boldsymbol{\sigma} \cdot \mathbf{n}\,\mathrm{d}\Gamma = \boldsymbol{\sigma} \cdot \mathrm{d}\boldsymbol{\Gamma}, \qquad \mathbf{t} = \boldsymbol{\sigma} \cdot \mathbf{n}. \tag{2.14}$$

Tensor $\boldsymbol{\sigma}$, called the Cauchy stress tensor, is a symmetric tensor that represents the stress state in the current configuration using spatial quantities. To define its counterpart for the reference configuration, Nanson's relation (Eq. (2.6)) can be used:

$$\mathbf{t}\,\mathrm{d}\Gamma = J\,\boldsymbol{\sigma} \cdot \mathbf{F}^{-T} \cdot \mathrm{d}\boldsymbol{\Gamma}_0 = \mathbf{P} \cdot \mathrm{d}\boldsymbol{\Gamma}_0, \qquad \mathbf{P} = J\,\boldsymbol{\sigma} \cdot \mathbf{F}^{-T}, \tag{2.15}$$

where $\mathbf{P}$ is a two-point nonsymmetric tensor also referred as the first Piola-Kirchhoff stress tensor. The first Piola-Kirchhoff tensor represents a spatial resultant force with material quantities. To represent stress using solely material quantities, the differential resultant force is pulled back to material configuration:

$$\mathbf{F}^{-1} \cdot \mathrm{d}\mathbf{f} = \mathbf{F}^{-1} \cdot \mathbf{P} \cdot \mathrm{d}\boldsymbol{\Gamma}_0 = \mathbf{S} \cdot \mathrm{d}\boldsymbol{\Gamma}_0, \qquad \mathbf{S} = J\,\mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-T}, \tag{2.16}$$

where $\mathbf{S}$ is the symmetric tensor also referred as the second Piola-Kirchhoff stress

tensor. An alternative to the Cauchy stress tensor is the Kirchhoff stress tensor which coincides with it in the case of incompressible deformations as follows:

$$\boldsymbol{\tau} = J\boldsymbol{\sigma}. \tag{2.17}$$

Apart from the presented stress tensors, it should be noted that other stress tensors exist in continuum mechanics literature.

2.1.3. Governing Equations

To describe motion and deformation of the continuum body upon applied loads the conservation or balance laws are used. There are five fundamental conservation laws that refer to the conservation of mass, linear momentum, angular momentum, energy, and entropy inequality. Since this thesis considers purely mechanical contact problems, the conservation of the energy and entropy inequality principle are not considered. In fact, for such a problem, the only fundamental law of interest is the conservation of linear momentum; the conservation of angular momentum proves the symmetry of the Cauchy tensor whereas mass conservation proves that motion and state of the body do not affect its mass. For the Lagrangian description of motion, the strong integral form of the conservation of the linear momentum, formulated with respect to the current configuration, can be written as:

$$\int_{\Omega} \frac{\partial}{\partial t}\left(\rho\mathbf{v}\right) d\Omega = \oint_{\Gamma} \boldsymbol{\sigma} \cdot \mathbf{n} \ d\Gamma + \int_{\Omega} \rho \ \mathbf{b}_{\Omega} \ d\Omega, \tag{2.18}$$

where $\mathbf{v}$ is the velocity vector, ρ is the density and $\mathbf{b}_{\Omega}$ are volume forces per unit mass. The term on the left-hand side presents inertial forces, whereas the first and the second term on the right-hand side present surface and body forces — their equality is in fact generalisation of Newton's second law of motion. Since displacement $\mathbf{u}$ is commonly taken as a solution variable in numerical procedures, it is more convenient to rewrite the left-hand side of the Eq. (2.18) as follows:

$$\int_{\Omega} \frac{\partial}{\partial t}\left(\rho\frac{\partial\mathbf{u}}{\partial t}\right) d\Omega = \oint_{\Gamma} \boldsymbol{\sigma} \cdot \mathbf{n} \ d\Gamma + \int_{\Omega} \rho \ \mathbf{b}_{\Omega} \ d\Omega. \tag{2.19}$$

The momentum balance derived with respect to the current configuration can be used under the assumption of infinitesimally small displacement, i.e. when current and reference configuration coincide. For large deformation problems current config-

uration is unknown therefore it is more convenient to formulate momentum balance with respect to the known material configuration:

$$\int_{\Omega_o} \rho_o \frac{\partial^2 \mathbf{u}}{\partial t^2} \, d\Omega_o = \oint_{\Gamma_o} (J\mathbf{F}^{-T} \cdot \mathbf{n}_o) \cdot \boldsymbol{\sigma} \, d\Gamma_o + \int_{\Omega_o} \rho_o \mathbf{b}_\Omega \, d\Omega_o, \qquad (2.20)$$

where subscript o is used to denote the material configuration (also called original configuration). Equivalently, the momentum balance can be reformulated with respect to any intermediate, i.e. updated configuration, denoted with subscript u:

$$\int_{\Omega_u} \frac{\partial}{\partial t} \left(\rho_u \frac{\partial \mathbf{u}}{\partial t} \right) \, d\Omega_u = \oint_{\Gamma_u} (j\mathbf{f}^{-T} \cdot \mathbf{n}_u) \cdot \boldsymbol{\sigma} \, d\Gamma_u + \int_{\Omega_u} \rho_u \mathbf{b}_\Omega \, d\Omega_u, \qquad (2.21)$$

where $\mathbf{f}$ is the relative deformation gradient and j is the relative Jacobian, $j = \det(\mathbf{f})$. The integration over the initial configuration is referred to as the Total Lagrangian (TL) approach whereas the integration over the updated configuration is referred to as the Updated Lagrangian (UL) approach. Mathematically, both approaches are equivalent — they differ in which configuration static and kinematic variables correspond to [80].

From the aspect of numerical procedures, the usage of the UL or TL approach differs in terms of numerical efficiency, i.e. suitability to provide a more accurate/robust numerical procedure. It is a common practice to use them in incremental forms in which the displacement field is decomposed using displacement increment $\Delta\mathbf{u}$ and displacement field $\mathbf{u}_u$ from the last calculated configuration, see Fig. 2.2.
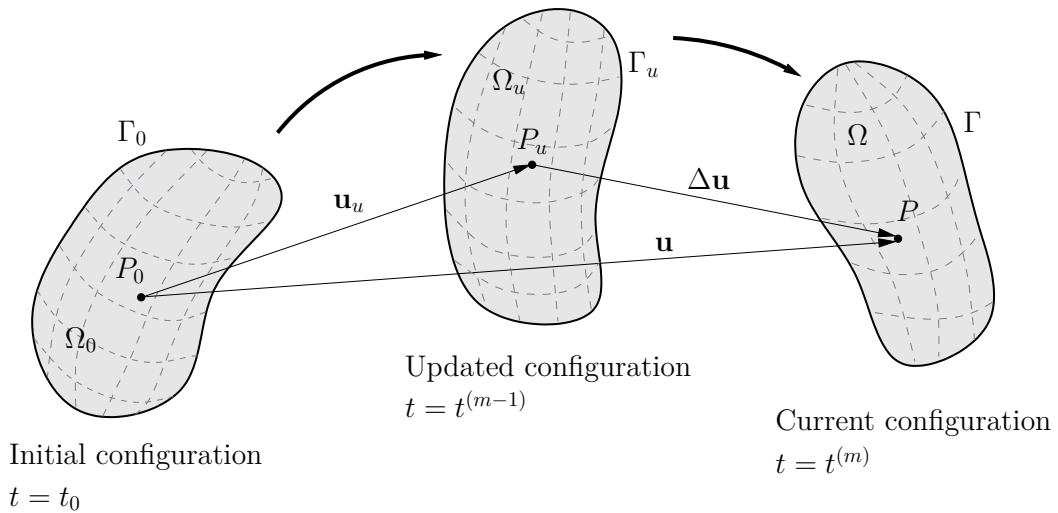
Figure 2.2. Reference and current configuration of the continuum body in motion

In this thesis the incremental UL approach is employed, i.e. momentum balance is formulated as:

$$\int_{\Omega_u} \frac{\partial}{\partial t} \left(\rho_u \frac{\partial (\mathbf{u}^{(m-1)} + \Delta \mathbf{u})}{\partial t} \right) \, d\Omega_u = \oint_{\Gamma_u} (j\mathbf{f}^{-\mathrm{T}} \cdot \mathbf{n}_u) \cdot \boldsymbol{\sigma} \, d\Gamma_u + \int_{\Omega_u} \rho_u \mathbf{b}_\Omega \, d\Omega_u, \quad (2.22)$$

where superscript m is used to denote the time-increment. Accordingly, superscript $m-1$ denotes the time-increment of the updated configuration, i.e. $\mathbf{u}_u = \mathbf{u}^{(m-1)}$. The relative deformation gradient, calculated with respect to the updated configuration, is:

$$\mathbf{f} = [\nabla_u(\Delta \mathbf{u})]^{\mathrm{T}} + \mathbf{I} \quad (2.23)$$

where ∇_u denotes the gradient with respect to the updated configuration.

2.1.4. Constitutive Equations

The constitutive equations or the so-called equations of state are linking equations between the previously presented stress and the strain quantities. In other words, for the considered conditions, the constitutive relations are used to approximate the physical behaviour of the analysed material. Accordingly, they depend on the considered material properties. Generally, constitutive relations are based on phenomenological constitutive models which rely on the experimental investigation of the material behaviour.

This thesis employs a rate-independent isotropic hyperelastoplastic constitutive relation based on the Kröner-Lee multiplicative decomposition of the deformation gradient $\mathbf{F} = \mathbf{F}^e \cdot \mathbf{F}^p$ together with the deviatoric-volumetric split of the strain energy function $W = W(\mathbf{F})$. For the strain energy function, expressed as a function of the deformation gradient, the Kirchhoff stress tensor is defined as:

$$\boldsymbol{\tau} = J\boldsymbol{\sigma} = 2\mathbf{F}^e \frac{\partial W}{\partial \mathbf{C}^e} \mathbf{F}^{e\mathrm{T}}. \quad (2.24)$$

Using the uncoupled strain energy function of the compressible neo-Hookean material [76], Eq. (2.24) yields:

$$\boldsymbol{\tau} = J \, \frac{K}{2} \left(\frac{J^2 - 1}{J} \right) \mathbf{I} + \mu \, \mathrm{dev}(\bar{\mathbf{b}}^e), \quad (2.25)$$

where the first and the second term on the right-hand side represent volumetric and deviatoric component, respectively. The $\bar{\mathbf{b}}^e$ is the volume preserving component of

15

2. Continuum Description of Contact Problem

the elastic left Cauchy-Green tensor, defined as:

$$\bar{\mathbf{b}}^e = J^{-\frac{2}{3}}\mathbf{b}^e = J^{-\frac{2}{3}}\mathbf{F}^e \cdot \mathbf{F}^{e\mathrm{T}}. \tag{2.26}$$

Note that for the incompressible (isochoric) plastic deformation $J^e = J$ thus super-script e is omitted. It is possible to express bulk modulus K and shear modulus μ in Eq. (2.25) using more common material parameters, Young modulus E and Poisson's ratio ν:

$$\mu, G = \frac{E}{2(1+\nu)}, \qquad K = \frac{E}{3(1-2\nu)}. \tag{2.27}$$

Eq. (2.25) is also used to model purely elastic response (compressible hyperelastic material). Moreover, it is suitable in the case of a small strain regime since its reduced form corresponds to the Hookean constitutive relation. The plasticity is modelled using the classical Huber-von Mises J_2 yield function, formulated as:

$$f(\boldsymbol{\tau}, \varepsilon^{p,eq}) = ||\mathrm{dev}(\boldsymbol{\tau})|| - \sqrt{\frac{2}{3}}\sigma_Y, \tag{2.28}$$

where $\sigma_Y = \sigma_Y(\varepsilon^{p\ eq})$ is the yield stress and $\varepsilon^{p\ eq}$ is the equivalent plastic strain. The associative flow rule is formulated as:

$$\frac{\partial}{\partial t}(\bar{\mathbf{F}}^{-1} \cdot \bar{\mathbf{b}}^e \bar{\mathbf{F}}^{-\mathrm{T}}) = -\frac{2}{3}\,\gamma\,\mathrm{tr}(\bar{\mathbf{b}}^e)\,\bar{\mathbf{F}}^{-1} \cdot \mathbf{N} \cdot \bar{\mathbf{F}}^{-\mathrm{T}}, \qquad \mathbf{N} = \frac{\mathrm{dev}(\boldsymbol{\tau})}{||\boldsymbol{\tau}||}, \tag{2.29}$$

where γ is the consistency parameter. The rate of change of the hardening is given by:

$$\dot{\varepsilon}^{p\ eq} = \sqrt{\frac{2}{3}}\gamma, \tag{2.30}$$

To complete the model, the consistency parameter and yield function are subjected to the Karush-Kuhn-Tucker (KKT) conditions together with consistency condition:

$$\gamma \geqslant 0, \qquad f(\varepsilon^{p\ eq}) \leqslant 0, \qquad \gamma f(\varepsilon^{p\ eq}) = 0, \qquad \gamma \dot{f}(\varepsilon^{p\ eq}) = 0. \tag{2.31}$$

It should be noted that the mathematical formulation of the contact problem is independent of the constitutive relations applied to bodies in contact [63]. Since the main focus of this thesis is contact modelling, further details regarding consti-tutive equations can be found in [76, 78, 81]. More insights regarding the numerical implementation of the presented constitutive equations in the context of the finite volume method can be found in [2].

2.2. Continuum Contact Mechanics

In the nonlinear solid mechanics, the previously presented equations are combined to assemble the boundary value problem, in which portions of the boundary are treated using the Dirichlet or Neumann boundary conditions. These portions of the boundary are known in advance, as well as their prescribed time-dependent distribution of the external traction or displacement. In contrast to this, the contact interaction results in the boundary with unknown distribution of displacement and contact traction. Besides, the contact boundary is a priori unknown, and it can change in time. Due to these reasons, the contact boundary needs special attention, i.e. mathematical treatment, in order to describe its highly nonlinear behaviour.

The intention of this section is to provide the contact quantities needed to formulate nonlinear contact conditions, which will be used to develop the numerical treatment of the contact boundary. From the continuum standpoint, contact treatment is independent of the applied numerical method; therefore, more details about mathematical modelling of contact can be found in the contact mechanics literature [14, 82], or in the computational contact mechanics literature [5, 63, 64].

2.2.1. Contact Kinematics

Before introducing fundamental contact quantities, consideration of two deformable bodies under contact at the current configuration is needed, as shown in Fig. 2.3. To identify bodies, superscript $i = \{1, 2\}$ is introduced. Note that the contact between two deformable bodies is a general scenario, thus the contact between a deformable body and a rigid obstacle, the self-contact and the contact between multiple bodies are in fact special cases covered by the description herein. Boundary $\Gamma^{(i)}$ of each body $\Omega^{(i)}$ is divided into the following non-intersecting portions:

$$\Gamma_{\mathbf{u}}^{(i)} \cup \Gamma_{\boldsymbol{\sigma}}^{(i)} \cup \Gamma_c^{(i)} = \Gamma^{(i)},$$
$$\Gamma_{\mathbf{u}}^{(i)} \cap \Gamma_{\boldsymbol{\sigma}}^{(i)} = \Gamma_{\boldsymbol{\sigma}}^{(i)} \cap \Gamma_c^{(i)} = \Gamma_{\mathbf{u}}^{(i)} \cap \Gamma_c^{(i)} = \emptyset, \tag{2.32}$$

where $\Gamma_{\mathbf{u}}^{(i)}$ and $\Gamma_{\boldsymbol{\sigma}}^{(i)}$ denote the portions of the boundary where displacement $\mathbf{u}_b^{(i)}$ and traction $\mathbf{t}_b^{(i)}$ are prescribed, respectively. The portion of the boundary where contact occurs is denoted with $\Gamma_c^{(i)}$. Further, we can distinguish two parts of the contact boundary: active and inactive. For the current configuration, active and

inactive parts are a priori unknown and they are part of the non-linear solution process for the current configuration. Moreover, they continuously evolve with time which means that for each time instance the potential contact boundary must be assumed in advance.
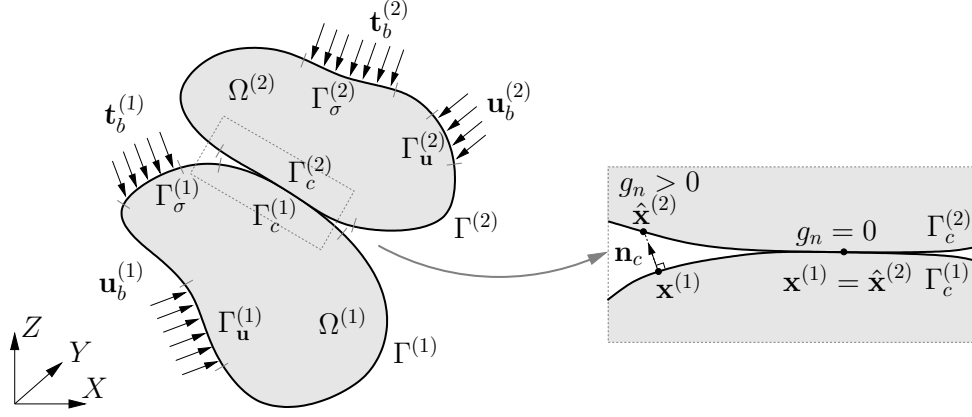
Figure 2.3. Current configuration of two deformable bodies in contact

As stated, for the current configuration, the portion of the boundary in contact is unknown and must be determined. Because of such type of non-linearity, contact related quantities are derived with respect to the current configuration while the momentum balance is derived for the initial or the updated configuration.

To measure the proximity of bodies in contact, i.e. the distance between them, the fundamental contact quantity, called gap function, is defined:

$$g_n(\mathbf{x}^{(1)}, \Gamma_c^{(2)}) = \left[\mathbf{x}^{(1)} - \hat{\mathbf{x}}^{(2)}(\mathbf{x}^{(1)}, \Gamma_c^{(2)})\right] \cdot \mathbf{n}_c \tag{2.33}$$

where $\mathbf{x}^{(1)}$ denotes point on the contact surface $\Gamma_c^{(1)}$ and $\hat{\mathbf{x}}^{(2)}$ denotes its projection on surface $\Gamma_c^{(2)}$. There are multiple approaches to the determination of projected point $\hat{\mathbf{x}}^{(2)}$ and corresponding contact normal vector $\mathbf{n}_c$. The most common approach defines projection as the minimum distance between point $\mathbf{x}^{(1)}$ and surface $\Gamma^{(2)}$:

$$\hat{\mathbf{x}}^{(2)} = \arg \min_{\mathbf{x}^{(2)} \in \Gamma_c^{(2)}} ||\mathbf{x}^{(1)} - \mathbf{x}^{(2)}||. \tag{2.34}$$

Eq. (2.34) represents the minimisation problem, which under centrain conditions — results in an orthogonal projection, as shown in Fig. 2.3. Next, the definition of the normal gap using Eq. (2.34) results in the asymmetric treatment of contact, i.e. the projected points do not have the same value of the gap function, see Fig 2.4 a). Futhermore, Fig. 2.4 b) and c) shows scenarios in which such an approach does not

have a solution or results in multiple solutions [5, 83]. These problems are further emphasized for the discretised boundary; therefore, this thesis employs a slightly different approach in which contact normal, $\mathbf{n}_c$, at point $\mathbf{x}^{(1)}$ is used for projection, as proposed in [84–86]. Such an approach, essentially based on the ray-tracing projection [87], offers some practical advantages over the minimum distance approach; however, it should be noted that other possibilities exist in the literature, for example, the shadow projection method [88]. Since from the continuum viewpoint, the active part of the contact boundary coincides and has zero value of the normal gap, irrespectively of its definition; this will be discussed in more detail in Chapter 4.
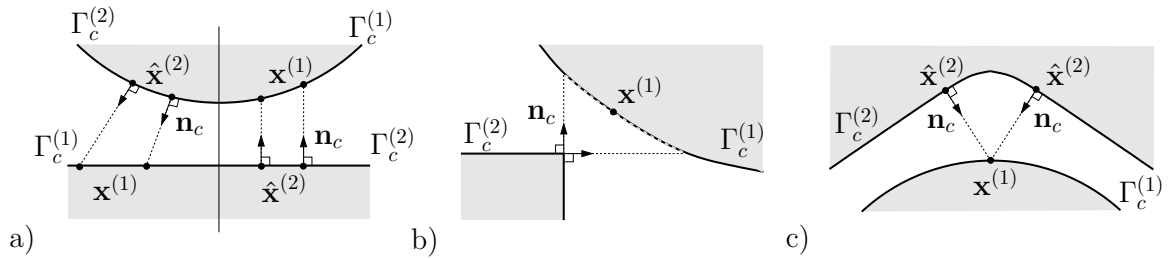
Figure 2.4. Minimum distance projection: a) asymmetry in gap function, b) non-existence of solution and c) existence of multiple solutions

The second kinematic quantity, relative tangential velocity, is used to describe the frictional contact response by means of the relative motion of surfaces in contact. The relative tangential velocity is defined as the change between point $\mathbf{x}^{(1)}$ and its projection:

$$\dot{\mathbf{g}}_t(\mathbf{x}^{(1)}) = (\mathbf{I} - \mathbf{n}_c\mathbf{n}_c) \bullet \left[\dot{\mathbf{x}}^{(1)}(\mathbf{x}^{(1)}) - \dot{\hat{\mathbf{x}}}^{(2)}(\mathbf{x}^{(1)})\right]. \tag{2.35}$$

where $(\mathbf{I} - \mathbf{n}_c\mathbf{n}_c)$ is the projection tensor [64] and the dot superscript is used to denote the time derivative. Although $\dot{\mathbf{g}}_t$ is a common notation in the literature, due to the ambiguity, it is more convenient to denote the relative tangential velocity as $\mathbf{v}_t$ [5]. Alternatively, incremental slip $\Delta\mathbf{g}_t$ can be used instead of the relative tangential velocity [70, 85, 89]:

$$\Delta\mathbf{g}_t(\mathbf{x}^{(1)}) = (\mathbf{I} - \mathbf{n}_c\mathbf{n}_c) \bullet \left[\mathring{\mathbf{x}}^{(1)}(\mathbf{x}^{(1)}) - \mathring{\hat{\mathbf{x}}}^{(2)}(\mathbf{x}^{(1)})\right], \tag{2.36}$$

where superscript $\circ$ is defined as:

$$(\mathring{\phi}) = \frac{\mathrm{d}\phi}{\mathrm{d}t}\,\Delta t = \phi^{(m)} - \phi^{(m-1)}, \tag{2.37}$$

where ϕ is a generic quantity and Δt is a arbitrary time increment. Equations for the incremental slip and the relative tangential velocity are exact in the case of continuum formulation — for perfect sliding and persistent contact ($g_n = \dot{g}_n = 0$). Otherwise, they are not frame-indifferent and they do not include the change of the closest point during sliding. Nevertheless, they are typically employed, although the prerequisites are not exactly met; the interested reader can find further details on this topic in [5, 63].

2.2.2. Contact Traction

On the active part of the contact boundary, contact stress arises and momentum balance, i.e. the action-reaction principle, must be satisfied at each contact point:

$$\mathbf{t}^{(1)} = -\mathbf{t}^{(2)}, \tag{2.38}$$

where $\mathbf{t}$ here denotes the contact traction (contact stress vector). In the absence of additional tractional forces, the inactive part of the contact boundary is treated as a stress-free boundary:

$$\boldsymbol{\sigma}^{(i)} \cdot \mathbf{n}^{(i)} = 0. \tag{2.39}$$

Next, the contact traction is decomposed into the normal and tangential components:

$$\mathbf{t}^{(i)} = \mathbf{t}_n^{(i)} + \mathbf{t}_t^{(i)}. \tag{2.40}$$

The normal component is further defined using a scalar variable, normal contact pressure, and outward-pointing surface normal vector:

$$\mathbf{t}_n^{(i)} = (\mathbf{t}^{(i)} \cdot \mathbf{n}^{(i)}) \cdot \mathbf{n}^{(i)} = p_n \mathbf{n}^{(i)}. \tag{2.41}$$

The tangential component, commonly denoted as tangential contact pressure $\mathbf{p}_t$ is obtained using projection tensor $\mathbb{P} = (\mathbf{I} - \mathbf{n}^{(i)}\mathbf{n}^{(i)})$:

$$\mathbf{t}_t^{(i)} = (\mathbf{I} - \mathbf{n}^{(i)}\mathbf{n}^{(i)}) \cdot \mathbf{t}^{(i)} = \mathbb{P} \cdot \mathbf{t}^{(i)}. \tag{2.42}$$

The unknown distribution of the contact traction components interrelates with the contact conditions explained in the next section.

2.2.3. Contact Conditions

At the contact surface, the unknown distribution of the contact traction cannot be uniquely defined, therefore this section introduces the concept of contact conditions. Contact conditions are in accordance with fundamental contact laws derived using the previously presented contact quantities. The non-linear contact behaviour in the normal and tangential directions obey the contact conditions for these directions. Consequently, the conditions for the normal and tangential directions are represented with non-linear functions which are interdependent and non-smooth (not-differentiable).

Normal contact conditions

In accordance with the unilateral contact law the contact behaviour in the normal direction is formulated using normal contact pressure p_n and gap function value g_n:

$$p_n \leqslant 0, \qquad g_n \geqslant 0, \qquad p_n g_n = 0. \tag{2.43}$$

The first condition implies that only compressive interaction exists (no adhesion), whereas the second term represents the geometric condition of the impenetrability of bodies in contact. The third term is a complementary condition that forces zero-gap value in the case of contact or zero value of contact pressure if there is no contact ($g_n > 0$). In the frictionless contact mechanics, these conditions are called the Hertz-Signorini-Moreau conditions, while in the optimisation theory they are known as the KKT conditions [90].

The graphical representation of the normal contact conditions is shown in Fig. 2.5 — the admissible combinations of g_n and p_n are coloured in red. As can be seen, the value of the contact pressure cannot be defined using the contact conditions. Moreover, the dependence of pressure and the normal gap is represented with a function that is non-linear and non-differentiable. As a result, the contact pressure can take an infinite number of negative values in the case of contact.
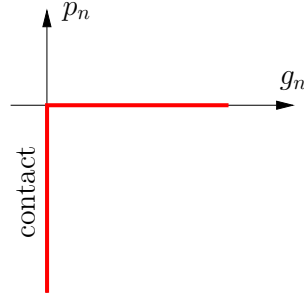
Figure 2.5. Graphical representation of normal contact conditions

Tangential contact conditions

This thesis employs the classical Coulomb friction law to describe the contact with dry friction response. The same as for normal conditions, the tangential (frictional) conditions are also represented using the KKT conditions:

$$\Phi \leqslant 0, \qquad \zeta \geqslant 0, \qquad \Phi\zeta = 0,$$
$$\Phi = ||\mathbf{t}_t|| - \mu|p_n|, \tag{2.44}$$
$$\dot{\mathbf{g}}_t + \zeta \frac{\mathbf{t}_t}{||\mathbf{t}_t||} = 0,$$

where ζ is the slip factor, Φ is the slip function and μ is the constant coefficient of friction, chosen with respect to material properties of bodies in contact. The first inequality condition, commonly called the Coulomb friction condition, mandates that the value of the tangential traction cannot exceed the threshold value defined using the normal contact pressure and the coefficient of friction. The second condition defines that the tangential traction is collinear and in the opposite direction of its slip direction. The last condition is the complementary condition which states that the slip is zero if the tangential traction is not equal to its threshold value. Alternatively, the existence of slip mandates that the tangential traction have its threshold value. In accordance with the last condition, we can distinguish two contact statuses: slip ($\zeta > 0$) and stick ($\zeta = 0$, no relative motion of the points in contact).

The graphical representation of tangential contact conditions is shown in Fig. 2.6 — the left-hand side represents the dependence of tangential traction and tangential sliding whereas the right-hand side represents the relation between the normal contact pressure and the component of the tangential traction (Coulomb's frictional

cone). Similarly to the normal direction, the friction law is represented with a non-differentiable Heaviside function. Moreover, the tangential conditions are related to the normal conditions and only the threshold value of the tangential traction is defined, while the tangential traction in the case of the stick status is not defined.
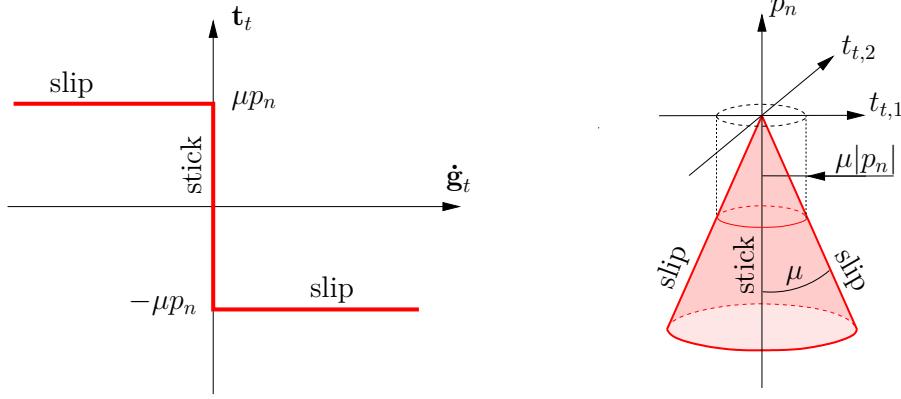
Figure 2.6. Graphical representation of tangential contact conditions

Due to the interrelation of the contact conditions, it is useful to summarise possible scenarios for each point on the potential contact boundary $\Gamma_c^{(i)}$:

$$
\begin{cases}
\text{contact active} \ \ p_n < 0, \ \ g_n = 0 \begin{cases} \text{stick status} \ \ \dot{\mathbf{g}}_\mathbf{t} = 0, \ \ ||\mathbf{t}_t|| < \mu|p_n| \\[2mm] \text{slip status} \ \ \dot{\mathbf{g}}_\mathbf{t} > 0, \ \ ||\mathbf{t}_t|| = \mu|p_n| \end{cases} , \\[6mm]
\text{contact inactive} \ \ g_n > 0, \ \ p_n = 0, \ \ \mathbf{t}_t = 0 \ .
\end{cases}
\tag{2.45}
$$

It should be noted that the presented normal and tangential contact conditions purely rely on the macroscopic observation of contact. Albeit such an approach is widely used in the computational contact mechanics, it is possible to take into account the micromechanical behaviour of contact surfaces by using statistically and experimentally developed constitutive relations. Nonetheless, the inclusion of such effects is straightforward if the numerical procedure based on macroscopic observation is already developed. From the perspective of the computational contact mechanics, more details on the micromechanical modelling of contact phenomena can be found in [5, 90].

2.3. Summary

This chapter provides a continuum-based formulation of the frictional contact problem between deformable solids. Bodies can undergo material as well as geometric nonlinearities, while the contact is considered as isothermal and non-adhesive.

In the first part, a review of the general principles of nonlinear solid mechanics is given. Fundamental kinematic, stress and strain quantities are introduced and used to present governing and constitutive equations. The second part is devoted to contact modelling. Using the presented contact quantities, the contact conditions for normal and tangential directions are formulated in accordance with fundamental contact laws: the impenetrability law and the Coulomb friction law.

The presented governing equation together with the constitutive equation and contact conditions, form the required set of equations needed to describe the class of contact problems dealt with in this thesis. Since these equations are well-known and their background can be discussed extensively, the presentation is kept brief and yet sufficient to cover all details necessary for the numerical discretisation using the finite volume method.

3. Finite Volume Structural Solvers

3.1. Introduction

The goal of this chapter is to provide an overview of one dominant class of structural solvers, for which the contact procedures, presented below, are developed. Since the finite volume (FV) discretisation starts with the strong integral form of the governing equations, the contact treatment can be analysed separately as a special class of the nonlinear boundary condition. Accordingly, the contact boundary conditions are discussed separately in Chapter 3.

The presented class of the FV structural solvers is based on the implicit cell-centred unstructured FV method with second-order spatial discretisation using compact computational molecules. The resulting system of equations (sparse and diagonally dominant) is solved employing a memory-efficient segregated solution procedure coupled with a deferred correction algorithm. Such a procedure is proven to be capable of solving highly nonlinear problems of solid mechanics and the majority of the developed structural solvers relies on it. It should be point out that some recent developments show that the coupled solution technique is a promising direction for further developments. However, the segregated solution procedure is still the only procedure capable of solving a broad spectrum of challenging nonlinear solid mechanics problems.

In the following sections, an overview of the cell-centred unstructured FV discretisation procedure is given. First, the discretisation of the solution domain is presented, followed by the discretisation of the governing equations. The equation discretisation is given rather in a general way by discretising the general form of momentum balance which is valid both for UL and TL formulation. The next sections present the treatment of the three main boundary conditions and characteristics of the resulting system of equations. Finally, the solution procedure is summarised and discussed. In the end, an overview of the available alternative solution or discretisation procedures is shown for the cell-centred FV method.

3.2. Discretisation

The finite volume discretisation procedure transforms the strong integral form of the governing equations into a corresponding system of linear algebraic equations. These equations represent a discrete approximation of the exact integral equations. The description of the discretisation procedure is separated into two distinct parts: solution domain discretisation and equation discretisation.

3.2.1. Solution Domain Discretisation

The discretisation of the solution domain is divided into space and time discretisation. The time discretisation consists of a division of the total simulation time interval into finite number of constant or varying time increments Δt and solving governing equations in a time-marching manner. The domain space is discretised by dividing it into the finite number of arbitrary convex polyhedral cells (control volumes) with computational nodes located at their centroids. The polyhedral cells do not overlap and fill space completely. The equation discretisation is not dependent on cell shape, therefore the domain space can consist of different cell shapes arranged in an either structured or unstructured manner.

An arbitrary polyhedral cell with volume Ω_P and computational node P located at $\mathbf{r}_P$ is shown in Fig. 3.1. The neighbouring cell with computational node N is connected to P with vector $\mathbf{d}_f$ and it shares the same face f with area vector $\mathbf{\Gamma}_f$.
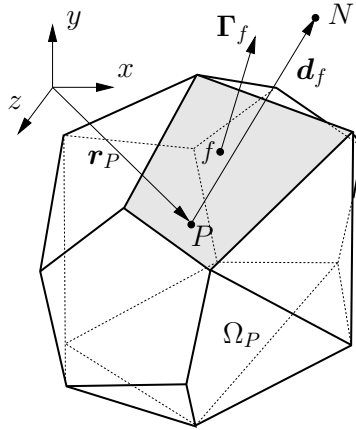
Figure 3.1. Arbitrary convex polyhedral cell [4]

3.2.2. Equation Discretisation

Since the equation discretisation is interdependent on the solution methodology, the discretisation presented here is designed for the segregated solution procedure in which the resulting system of equations is temporarily decoupled and solved sequentially per each component. To account for the inter-component coupling, the outer Fixed-point/Picard iterations are performed.

We can distinguish two different treatments of the equation terms: implicit and explicit. The implicit treatment refers to the discretisation using an unknown solution variable, whereas the explicit treatment refers to the calculation using the available value of the solution variable. Accordingly, the explicit treatment will contribute to the source vector of the resulting system of the equations, whereas the implicit treatment will contribute to the resulting left-hand side coefficient matrix.

In the case of an isothermal Hookean solid (small strain and rotations) the term which represents the surface forces has the following form:

$$\oint_{\Gamma_o} \mathbf{n}_o \cdot \boldsymbol{\sigma} \, d\Gamma_o = \oint_{\Gamma_o} \mathbf{n}_o \cdot \left[\mu \boldsymbol{\nabla} \mathbf{u} + \mu \boldsymbol{\nabla} \mathbf{u}^T + \lambda \mathrm{tr}(\boldsymbol{\nabla} \mathbf{u}) \mathbf{I} \right] \, d\Gamma_o, \qquad (3.1)$$

where only the first term $\mu \mathbf{n}_o \cdot \boldsymbol{\nabla} \mathbf{u}$ (surface displacement Laplacian) can have implicit treatment. The remaining terms (inter-equation coupling terms) are explicitly discretised and updated in a lagged correction manner. The second and the third term (Laplacian transpose term and trace term) are inter-equation coupling terms, which are explicitly discretised and updated in a lagged correction manner. In [18] it is shown that approximating the Laplacian transpose term and trace term using the displacement Laplacian can significantly improve the convergence rate:

$$\oint_{\Gamma_o} (2\mu + \lambda) \, \mathbf{n}_o \cdot \boldsymbol{\nabla} \mathbf{u} \, d\Gamma_o + \oint_{\Gamma_o} \mathbf{n}_o \cdot \left[\mu \boldsymbol{\nabla} \mathbf{u}^T + \lambda \mathrm{tr}(\boldsymbol{\nabla} \mathbf{u}) \mathbf{I} - (\mu + \lambda) \boldsymbol{\nabla} \mathbf{u} \right] \, d\Gamma_o. \quad (3.2)$$

Although this modification allows for an optimal convergence rate, in the case of strong inter-equation coupling, a drastically lower convergence rate can be still expected [18, 24].

By further rearrangement, Eq. (3.2) can be rewritten as:

$$\oint_{\Gamma_o} (2\mu + \lambda) \, \mathbf{n}_o \cdot \boldsymbol{\nabla} \mathbf{u} \, d\Gamma_o + \oint_{\Gamma_o} \mathbf{n}_o \cdot \boldsymbol{\sigma} \, d\Gamma_o - \oint_{\Gamma_o} (2\mu + \lambda) \, \mathbf{n}_o \cdot \boldsymbol{\nabla} \mathbf{u} \, d\Gamma_o, \qquad (3.3)$$

where only the first term is treated implicitly and the remaining terms are treated explicitly. It can seen that the modified equation is mathematically equivalent to

original Eq. (3.1) since the first and the third term cancel out.

Using Eq. (3.3) the conservation of linear momentum (Eq. (2.19)) has the following form:

$$\int_{\Omega_o} \rho_o \frac{\partial^2 \mathbf{u}}{\partial t^2} \, \mathrm{d}\Omega_o = \oint_{\Gamma_o} \mathbf{t}_{\boldsymbol{\sigma}} \, \mathrm{d}\Gamma_o + \oint_{\Gamma_o} \mathbf{n}_o \boldsymbol{\cdot} \boldsymbol{\sigma} \, \mathrm{d}\Gamma_o - \oint_{\Gamma_o} \mathbf{t}_{\boldsymbol{\sigma}} \, \mathrm{d}\Gamma_o + \int_{\Omega_o} \rho_o \mathbf{b}_{\Omega} \, \mathrm{d}\Omega_o, \quad (3.4)$$

where traction field $\mathbf{t}_{\boldsymbol{\sigma}} = (2\mu + \lambda) \, \mathbf{n}_o \boldsymbol{\cdot} \boldsymbol{\nabla} \mathbf{u}$ is used to denote the approximation of the surface forces in terms of displacement field. The diffusion coefficient, $(2\mu + \lambda)$, is also called implicit stiffness because its value matches the first three diagonal components of the fourth-order stiffness (constitutive) tensor for linear elasticity [57]. In the case of a non-linear constitutive relation, the choice of optimal traction field $\mathbf{t}_{\boldsymbol{\sigma}}$, i.e. implicit stiffness is less trivial; however, the approximation used for the linear-elastic solid can be applied without drastically affecting the overall efficiency [57]. Accordingly, the conservation of the linear momentum in the case of the TL and UL formulation (Eq. (2.20) and Eq. (2.21)) can be rewritten as:

$$\int_{\Omega_o} \rho_o \frac{\partial^2 \mathbf{u}}{\partial t^2} \, \mathrm{d}\Omega_o =$$
$$= \oint_{\Gamma_o} \mathbf{t}_{\boldsymbol{\sigma}} \, \mathrm{d}\Gamma_o + \oint_{\Gamma_o} (J\mathbf{F}^{-\mathrm{T}} \boldsymbol{\cdot} \mathbf{n}_o) \boldsymbol{\cdot} \boldsymbol{\sigma} \, \mathrm{d}\Gamma_o - \oint_{\Gamma_o} \mathbf{t}_{\boldsymbol{\sigma}} \, \mathrm{d}\Gamma_o + \int_{\Omega_o} \rho_o \mathbf{b}_{\Omega} \, \mathrm{d}\Omega_o, \quad (3.5)$$

$$\int_{\Omega_u} \rho_u \frac{\partial^2 \mathbf{u}}{\partial t^2} \, \mathrm{d}\Omega_u =$$
$$= \oint_{\Gamma_u} \mathbf{t}_{\boldsymbol{\sigma}} \, \mathrm{d}\Gamma_u + \oint_{\Gamma_u} (j\mathbf{f}^{-\mathrm{T}} \boldsymbol{\cdot} \mathbf{n}_u) \boldsymbol{\cdot} \boldsymbol{\sigma} \, \mathrm{d}\Gamma_u - \oint_{\Gamma_u} \mathbf{t}_{\boldsymbol{\sigma}} \, \mathrm{d}\Gamma_u + \int_{\Omega_u} \rho_u \mathbf{b}_{\Omega} \, \mathrm{d}\Omega_u. \quad (3.6)$$

In the case of the incremental UL or TL form, dependent variable $\mathbf{u}$ is replaced by $\mathbf{u}^{(m-1)} + \Delta\mathbf{u}$. It is possible to arrange Eq. (3.4), (3.5) and (3.6) in a general form for which discretisation of each component will be given separately:

$$\underbrace{\int_{\Omega} \rho \frac{\partial^2 \boldsymbol{\phi}}{\partial t^2} \, \mathrm{d}\Omega}_{\text{Temporal term}} = \underbrace{\oint_{\Gamma} (2\mu + \lambda)\mathbf{n} \boldsymbol{\cdot} \boldsymbol{\nabla}\boldsymbol{\phi} \, \mathrm{d}\Gamma}_{\substack{\text{Implicit} \\ \text{diffusion term}}} - \underbrace{\oint_{\Gamma} (2\mu + \lambda)\mathbf{n} \boldsymbol{\cdot} \boldsymbol{\nabla}\boldsymbol{\phi} \, \mathrm{d}\Gamma}_{\substack{\text{Explicit} \\ \text{diffusion term}}}$$
$$+ \underbrace{\oint_{\Gamma} \mathbf{n} \boldsymbol{\cdot} \hat{\boldsymbol{\sigma}} \, \mathrm{d}\Gamma}_{\substack{\text{Explicit} \\ \text{surface source term}}} + \underbrace{\int_{\Omega} \rho \mathbf{b}_{\Omega} \, \mathrm{d}\Omega}_{\text{Volume source term}}, \quad (3.7)$$

where dependent variable $\boldsymbol{\phi}$ can be either in total or incremental form. In the case of the UL formulation, all values are referred to the updated configuration, whereas in the case of the TL formulation the values are referred to the initial undeformed

configuration.

Approximation of the surface and volume integrals

By assuming the linear variation of dependent variable ϕ across the CV interior and surfaces:

$$
\begin{aligned}
\phi(\mathbf{r}) &= \phi_P + (\mathbf{r} - \mathbf{r}_P) \cdot (\boldsymbol{\nabla}\phi)_P, \\
\phi(\mathbf{r}) &= \phi_f + (\mathbf{r} - \mathbf{r}_f) \cdot (\boldsymbol{\nabla}\phi)_f,
\end{aligned}
\tag{3.8}
$$

the volume integrals together with surface integrals in Eq. (3.7) are approximated to a second-order accuracy by employing the mid-point rule:

$$
\int_\Omega \phi(\mathbf{r})\, \mathrm{d}\Omega = \phi_P \Omega_P, \qquad \int_\Gamma \phi(\mathbf{r})\, \mathrm{d}\Gamma = \phi_f \Gamma_f.
\tag{3.9}
$$

Temporal term

The time derivative is discretised using the backward Euler finite differencing scheme:

$$
\left(\frac{\partial \phi}{\partial t}\right)^{(m)} \approx \frac{\phi_P^{(m)} - \phi_P^{(m-1)}}{\Delta t^{(m)}},
\tag{3.10}
$$

which is bounded and first-order accurate in time. Superscript m is used to denote the current time instance to which value refers. The first antecedent time instance is denoted with $m-1$, while the second one with $m-2$. Using Eq. (3.10) the temporal term in Eq. (3.7) is discretised as:

$$
\begin{aligned}
\int_\Omega \rho \frac{\partial^2 \phi}{\partial t^2}\, \mathrm{d}\Omega &\approx \frac{1}{\Delta t^{(m)}} \left[\left(\rho_P \Omega_P \frac{\partial \phi}{\partial t}\right)^{(m)} - \left(\rho_P \Omega_P \frac{\partial \phi}{\partial t}\right)^{(m-1)} \right] \\
&\approx \frac{1}{\Delta t^{(m)}} \left[(\rho_P \Omega_P)^{(m)} \left(\frac{\phi_P^{(m)} - \phi_P^{(m-1)}}{\Delta t^{(m)}}\right) - (\rho_P \Omega_P)^{(m-1)} \left(\frac{\phi_P^{(m-1)} - \phi_P^{(m-2)}}{\Delta t^{(m-1)}}\right) \right].
\end{aligned}
\tag{3.11}
$$

Instead of the first-order scheme, it is possible to discretise temporal term using any other second-order scheme [4, 18, 91].

Diffusion term

The implicit diffusion term is decomposed into the orthogonal and non-orthogonal component using the over-relaxed approach [92, 93]. The normal component, calculated using the central differencing scheme is implicitly treated, whereas the non-

orthogonal component is treated explicitly:

$$\oint (2\mu + \lambda)\mathbf{n} \cdot \boldsymbol{\nabla}\boldsymbol{\phi} \ \mathrm{d}\Gamma = \sum_F \int_{\Gamma_f} (2\mu_f + \lambda_f)\mathbf{n}_f \cdot (\boldsymbol{\nabla}\boldsymbol{\phi})_f \ \mathrm{d}\Gamma_f$$

$$\approx \sum_F (2\mu_f + \lambda_f)\Big[||\boldsymbol{\Delta}_f||\frac{\phi_N - \phi_P}{||\mathbf{d}_f||} + \underbrace{\mathbf{k}_f \cdot (\boldsymbol{\nabla}\boldsymbol{\phi})_f^*}_{\text{Non-orthogonal correction}}\Big]||\boldsymbol{\Gamma}_f||, \tag{3.12}$$

where $\boldsymbol{\Delta}_f = \mathbf{d}_f/(\mathbf{d}_f \cdot \mathbf{n}_f)$ and $\mathbf{k}_f = \mathbf{n}_f - \boldsymbol{\Delta}_f$. Superscript $*$ is introduced to denote explicitly calculated values. The non-orthogonal component (correction for cell-face non-orthogonality) is treated in a lagged correction manner with the face-centre gradient calculated using the linear interpolation:

$$(\boldsymbol{\nabla}\boldsymbol{\phi})_f^* \approx \gamma_x(\boldsymbol{\nabla}\boldsymbol{\phi})_P^* + (1 - \gamma_x)(\boldsymbol{\nabla}\boldsymbol{\phi})_{N_f}^*, \tag{3.13}$$

where $\gamma_f = ||\mathbf{r}_{N_f} - \mathbf{r}_f||/||\mathbf{d}_f||$ is the interpolation weight determined by using the inverse distance method. The cell centre gradient is calculated by using the weighted linear least-squares method [16, 18].

The explicit diffusion term is calculated also by using Eq. (3.12) but with the explicit calculation of the normal component. Accordingly, because of the identical discretisation, the explicitly and the implicitly discretised diffusion term cancel out for the final converged solution.

By employing Eq. (3.12) the diffusion term is discretised using a compact computational molecule, therefore oscillations with twice the period of the cell size will not be registered. To reduce the oscillations, i.e. the so-called checker-boarding spatial instabilities (known as zero-energy modes or hourglassing in the FEM) it is possible to introduce additional numerical diffusion by employing the Rhie-Chow stabilisation term [16, 94]. The stabilisation term is calculated by employing linear interpolation (3.13), which consequently results in the usage of a larger computational molecule:

$$\sum_F \int_{\Gamma_f} (2\mu_f + \lambda_f) \ \mathbf{n}_f \cdot (\boldsymbol{\nabla}\boldsymbol{\phi})_f \ \mathrm{d}\Gamma_f =$$
$$\sum_F (2\mu_f + \lambda_f) \ \mathbf{n}_f \cdot \Big[\gamma_x(\boldsymbol{\nabla}\boldsymbol{\phi})_P^* + (1 - \gamma_x)(\boldsymbol{\nabla}\boldsymbol{\phi})_{N_f}^*\Big] ||\boldsymbol{\Gamma}_f||. \tag{3.14}$$

The final form of the stabilisation term is obtained by subtracting Eq. (3.14) from Eq. (3.12) and by multiplying it with a scaling factor — typically set to $\alpha_{RC} = 0.1$.

Similar smoothing effects are obtainable by using alternative approaches, such as the Jameson–Schmidt–Turkel [95] or the Godunov-type [96] approach, see [4] for further details.

Surface source term

Using the mid-point rule, the surface diffusion source terms are discretised as follows:

$$\oint_\Gamma \mathbf{n} \boldsymbol{\cdot} \hat{\boldsymbol{\sigma}} \ \mathrm{d}\Gamma = \sum_f \int_{\Gamma_f} \mathbf{n}_f \boldsymbol{\cdot} \hat{\boldsymbol{\sigma}}_f \ \mathrm{d}\Gamma_f \approx \sum_f \boldsymbol{\Gamma}_f \boldsymbol{\cdot} \hat{\boldsymbol{\sigma}}_f. \tag{3.15}$$

The general form of stress tensor $\hat{\boldsymbol{\sigma}}$ in the case of the UL or the TL formulation is:

$$\hat{\boldsymbol{\sigma}}_{f \ UL} = (j\boldsymbol{\sigma} \boldsymbol{\cdot} \mathbf{f}^{-\mathrm{T}})_f, \qquad \hat{\boldsymbol{\sigma}}_{f \ TL} = (J\boldsymbol{\sigma} \boldsymbol{\cdot} \mathbf{F}^{-\mathrm{T}})_f. \tag{3.16}$$

where Cauchy stress tensor $\boldsymbol{\sigma}$ is calculated in accordance with the chosen constitutive equation. In this thesis, the Cauchy stress tensor for hyperelastic and hyperelastoplastic material behaviour is calculated using Eq. (2.25). In the case of an isothermal Hookean solid (small strain and rotations), due to the assumption of infinitesimal deformations, tensor $\hat{\boldsymbol{\sigma}}$ coincides with $\boldsymbol{\sigma}$ and it has the following form:

$$\boldsymbol{\sigma}_f = \mu_f (\boldsymbol{\nabla}\mathbf{u})_f + \mu_f (\boldsymbol{\nabla}\mathbf{u})_f^{\mathrm{T}} + \lambda_f \mathrm{tr}(\boldsymbol{\nabla}\mathbf{u})_f \mathbf{I}. \tag{3.17}$$

Irrespective of the constitutive equation applied, the Cauchy stress tensor, i.e. surface source term is completely explicitly calculated — superscript $*$ is omitted for better clarity.

Volume source term

Similarly to the surface source term, the volume integral of the body forces is discretised as:

$$\int_\Omega \rho \mathbf{b}_\Omega \ \mathrm{d}\Omega \approx \rho_P \mathbf{b}_{\Omega \ P} \Omega_P. \tag{3.18}$$

3.2.3. Boundary and Initial Conditions

In the derived discretisation of the linear momentum, all faces are treated as internal and no distinction is made for faces at the boundary of the discretised solution domain. To account for these faces and their conditions, the boundary conditions are incorporated into the discretised linear momentum equation.

Figure 3.2 shows the boundary face with its centre b and distance vector $\mathbf{d}_b$. To take the boundary non-orthogonality into account, distance vector $\mathbf{d}_b$ is decomposed into the orthogonal $\mathbf{d}_n = (\mathbf{n}_b \cdot \mathbf{d}_b)\mathbf{n}_b$ and the non-orthogonal $\mathbf{k}_b$ component. In solving fluid flow problems it is a standard practice to assume constant distribution of ϕ across the face, therefore there is no need to perform boundary non-orthogonal correction. Previous studies [50, 97] show that introducing such an assumption in solid mechanics results in the unphysical distribution of stresses at the boundary. Because of that, preserving the second-order accuracy using boundary the non-orthogonal correction became a rule for problems in solid mechanics.

Apart from the boundary conditions, initial conditions must also be specified for the whole computational domain. The initial conditions are used only at the first time/load increment and they specify the distribution of ϕ and $\partial\phi/\partial t$.

In the remainder of this subsection, the implementation of three main types of boundary conditions is outlined.
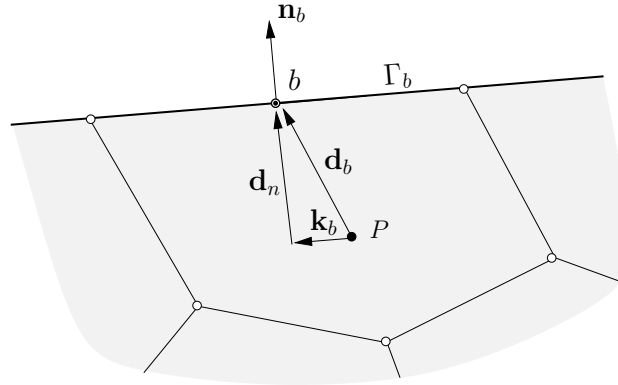
Figure 3.2. Control volume at boundary

Dirichlet/Displacement Boundary Condition

The displacement boundary condition (Dirichlet boundary condition) specifies a constant or time-varying discrete distribution of the dependent variable. For the boundary faces, the implicit surface diffusion term has the following form:

$$\int_{\Gamma_b} (2\mu_b + \lambda_b)\mathbf{n}_b \cdot (\boldsymbol{\nabla}\phi)_b \, \mathrm{d}\Gamma_b \approx (2\mu_b + \lambda_b)\left[\frac{\phi_b - \phi_P}{||\mathbf{d}_n||} + \mathbf{k}_b \cdot (\boldsymbol{\nabla}\phi)_b^*\right]||\boldsymbol{\Gamma}_f||, \quad (3.19)$$

where ϕ_b denotes a specified value at the boundary face centre.

Neumann/Traction Boundary Condition

The traction boundary condition (Neumann boundary condition) prescribes the constant or time-varying discrete distribution of the boundary gradient $\mathbf{g}_b$:

$$\mathbf{g}_b = \mathbf{n}_b \cdot (\boldsymbol{\nabla}\phi)_b, \tag{3.20}$$

which is directly included in the discretised diffusion term:

$$\int_{\Gamma_b} (2\mu_b + \lambda_b)\mathbf{n}_b \cdot (\boldsymbol{\nabla}\phi)_b \, \mathrm{d}\Gamma_b \approx (2\mu_b + \lambda_b)\mathbf{g}_b ||\boldsymbol{\Gamma}_f||, \tag{3.21}$$

The calculation of the normal gradient is conducted using the following decomposition:

$$\mathbf{g}_b = \Delta\mathbf{g}_b + \mathbf{g}_b^* = \Delta\mathbf{g}_b + \mathbf{n}_b(\boldsymbol{\nabla}\phi)_b^*, \tag{3.22}$$

where boundary normal $\mathbf{n}_b$ is in the initial or updated configuration depending on the mathematical formulation used. Gradient increment $\Delta\mathbf{g}_b$ is calculated by using the difference between prescribed/targeted boundary traction $\mathbf{t}_b$ and the currently calculated boundary traction:

$$\Delta\mathbf{g}_b = \frac{1}{(2\mu_b + \lambda_b)}(\mathbf{t}_b - \mathbf{n}_b^* \cdot \boldsymbol{\sigma}_b^*), \tag{3.23}$$

where current boundary normal $\mathbf{n}_b^*$ is calculated by using the deformation gradient from the previous outer iteration. Note that Eq. (3.21) is also used to describe a traction-free boundary simply by setting $\mathbf{g}_b$ to zero vector.

The value of the dependent variable at boundary is necessary to updated kinematics and stress quantities at the end of each outer iteration. The calculation is done by discretising the boundary gradient in Eq. (3.20) using a central differencing scheme with the addition of non-orthogonal correction:

$$\phi_b = \phi_P + ||\mathbf{d_n}||\mathbf{g}_b + \mathbf{k}_b \cdot (\boldsymbol{\nabla}\phi)_P. \tag{3.24}$$

The added non-orthogonal correction uses gradient at the cell centre, therefore assuming the constant distribution of the tangential gradient. An alternative option for the boundary gradient usage in the non-orthogonal correction is to extrapolate it from the interior using the least-square method.

Symmetry Plane

The usage of the symmetry boundary condition (Robin boundary condition) directly implies the symmetry of geometry, material and load. In solid mechanics it is not unusual to satisfy all of these conditions, therefore the symmetry boundary condition is often used to reduce the problem size. Mathematically, the symmetry boundary is described by specifying zero value of the normal component of the dependent variable and by zero value of the normal gradient of the tangential component:

$$(\boldsymbol{\phi}_n)_b = (\mathbf{n}_b \cdot \boldsymbol{\phi}_b)\mathbf{n}_b = 0, \qquad \mathbf{n}_b \cdot \boldsymbol{\nabla}(\boldsymbol{\phi}_t)_b = 0. \tag{3.25}$$

Following the same algebraic manipulation as for Eq. (3.24), the value of the dependent variable at the symmetry boundary is:

$$\boldsymbol{\phi}_b = (\mathbf{I} - \mathbf{n}_b\mathbf{n}_b) \cdot (\boldsymbol{\phi}_P + \mathbf{k}_b \cdot (\boldsymbol{\nabla}\boldsymbol{\phi})_P). \tag{3.26}$$

3.2.4. Resulting System of Equations

For each control volume P, the momentum balance in the final discretised form is written in the form of a linear algebraic equation:

$$a_P\boldsymbol{\phi}_P + \sum_{N_f} a_{N_f}\boldsymbol{\phi}_{N_f} = \mathbf{r}_P, \tag{3.27}$$

where a_P is the central (diagonal) coefficient, a_N is the neighbouring cell coefficient and $\mathbf{r}_P$ is the source term. In the case of the existence of contact or multi-material interface with implicit treatment, a distinction is made and the algebraic equation for control volume P includes the contribution from the neighbouring CV shared at the interface:

$$a_P\boldsymbol{\phi}_P + \sum_{N_f} a_{N_f}\boldsymbol{\phi}_{N_f} + \sum_{N_{fi}} a_{Nfi}\,\boldsymbol{\phi}_{Nfi} = \mathbf{r}_P, \tag{3.28}$$

where a_{Ni} is the interface neighbouring cell coefficient which can be direction dependent. In accordance with the discretisation procedure presented in Section 3.2.2 the coefficients of the linear algebraic equation for the incremental UL formulation

of the momentum balance, used in this thesis, are listed below:

$$
a_P = \frac{\rho_u \Omega_u}{\Delta t^2} + \sum_F (2\mu_f + \lambda_f) \frac{||\mathbf{\Delta}_{uf}||}{||\mathbf{d}_{uf}||} |\mathbf{\Gamma}_{uf}|,
$$

$$
a_N = -(2\mu_f + \lambda_f) \frac{||\mathbf{\Delta}_{uf}||}{||\mathbf{d}_{uf}||} ||\mathbf{\Gamma}_{uf}||,
$$

$$
\begin{aligned}
\mathbf{r}_p =\ & \sum_F (2\mu_f + \lambda_f)[\mathbf{k}_{uf} \boldsymbol{\cdot} (\boldsymbol{\nabla}(\Delta\mathbf{u}))_f^*]||\mathbf{\Gamma}_{uf}|| + \sum_F \mathbf{n}_u \boldsymbol{\cdot} (j\boldsymbol{\sigma}^* \boldsymbol{\cdot} \mathbf{f}^{-\mathrm{T}})_f ||\mathbf{\Gamma}_{uf}|| \\
& - (1 + \alpha_{RC}) \sum_F (2\mu_f + \lambda_f) \left[||\mathbf{\Delta}_{uf}|| \frac{(\Delta\mathbf{u})_N^* - (\Delta\mathbf{u})_P^*}{||\mathbf{d}_{uf}||} + \mathbf{k}_{uf} \boldsymbol{\cdot} (\boldsymbol{\nabla}(\Delta\mathbf{u}))_f^* \right] ||\mathbf{\Gamma}_{uf}|| \quad (3.29) \\
& - \alpha_{RC} \sum_F (2\mu_f + \lambda_f)\, \mathbf{n}_u \boldsymbol{\cdot} (\boldsymbol{\nabla}(\Delta\mathbf{u}))_f^* ||\mathbf{\Gamma}_f|| + \frac{\rho_{uP}\Omega_{uP}}{\Delta t^2}(\Delta\mathbf{u})_P^{(m-1)} + \rho_{uP}\mathbf{b}_{\Omega P}\Omega_{uP} \\
& + \frac{(\rho_{uP}\Omega_{uP})^{(m-1)}}{\Delta t^2} \left((\Delta\mathbf{u})_P^{(m-1)} - (\Delta\mathbf{u})_P^{(m-2)} \right) \\
& - \frac{1}{\Delta t} \left(\left(\rho_P \Omega_P \frac{\partial \mathbf{u}_u}{\partial t} \right)^{(m-2)} - \left(\rho_P \Omega_P \frac{\partial \mathbf{u}_u}{\partial t} \right)^{(m-3)} \right).
\end{aligned}
$$

Note that the coefficients arising from the inertia term discretisation, derived under the assumption of constant time-step Δt, are neglected in the quasi-static analysis. Interface neighbouring coefficients a_{Ni} for the multi-material interface can be found in [24, 50], whereas for the implicit contact interface they are discussed separately in Chapter 4. Assembling Eqs. (3.27) or (3.28) for all CVs results in the system of equations:

$$
[A][\boldsymbol{\phi}] = [\mathbf{r}], \tag{3.30}
$$

where $[A]$ is a square matrix with a_P coefficients on the diagonal and a_N coefficients on the off-diagonal positions. The size of matrix $[A]$ is equal to the number of CVs. Vector $[\boldsymbol{\phi}]$ represents the value of the unknown solution variable ($\mathbf{u}$ or $\Delta\mathbf{u}$) at the cell centres while the right-hand side vector $[\mathbf{r}]$ consists of the source terms.

The number of off-diagonal coefficients in each row depends on the corresponding CV shape. In other words, it depends on the number of CV internal polygonal faces which are shared by the neighbouring CV. For example, the number of off-diagonal coefficients in the case of tetrahedra is 4, in the case of triangular prisms it is 5, in the case of hexahedra is 6, etc. Correspondingly, matrix $[A]$ is a sparse matrix with most of the coefficients equal to zero.

In the case of neglecting inertial effects, matrix $[A]$ is diagonally equal (symmetric and diagonally dominant) as it is constructed by the orthogonal component of the discretised surface Laplacian term. The temporal term contributes to the matrix diagonal coefficients beneficially, increasing its diagonal dominance.

The above-listed properties of matrix $[A]$ allow for efficient usage of the iterative linear system solver. The most common choice in FV solid mechanics literature is the incomplete Cholesky pre-conditioned conjugate gradient (ICCG) method [98,99] which is also used in this thesis. Since the solution vector, $\mathbf{b}$, is approximation updated within the outer iterations, there is no need to solve the system to a tight tolerance [18,45,100]. Typically, it is sufficient to reduce the residuals by an order of magnitude. In that way, the number of solver iterations (inner iterations) is held to a relatively small number, thus boosting the overall efficiency.

3.3. Solution Procedure

The overall solution procedure/algorithm can be summarised as follows:

I) Initialise fields using the initial conditions or results from the previous increment.

   i) Update the boundary conditions.

   ii) Discretise the momentum equation and assemble the system of Eqs. (3.30).

   iii) Solve the assembled system of equations in a segregated manner using an iterative solver.

   iv) Update the kinematics and stress.

   v) If the convergence criteria are satisfied proceed to (II), otherwise make a new outer iteration by returning to step (i).

II) Depending on the mathematical formulation used, move the mesh to the deformed configuration.

III) If the final time/load increment is not reached, proceed to the next increment by returning to step I).

Overall, for the case of the segregated solution technique, two key loops can be distinguished. The main loop resolves the given time/load increment using a finite number of outer iterations, i.e corrections. The number of corrections n_{corr} needed to resolve the inter-equation coupling terms and any other lagged terms contained by vector $\mathbf{b}$ is case dependent and can vary between time/load increments. The iterative nature of the solution procedure allows for a simple incorporation and resolution of

any non-linearity; however, in some scenarios, this results in impaired stability. Consequently, to ensure convergence, it is necessary to perform explicit and/or implicit under-relaxation. Values typically used for the explicit under-relaxation factor are in the range 0.95-1, and for the implicit under-relaxation factor the values are 0.99-1. A significant acceleration of the outer iteration loop can be obtained by employing a geometric multigrid technique [21,29,45]; however, this technique comes with a somehow impractical task of generating a hierarchy of coarser grids.

In the case of contact modelling, the overall solution procedure remains the same and contact treatment is incorporated within the first two steps of the outer iteration loop. Within the first step, newly calculated values of the displacement field are used to update the position of the contact boundary. Subsequently, the contact algorithm is invoked to calculate contact traction. In the second step, assembly of the system of equations is performed in accordance with the contact coupling type. In the case of explicit contact coupling only the solution vector is updated; whereas, in the case of the implicit contact coupling, the left-hand side matrix of coefficients is updated as well as the solution vector. Further details on the contact treatment are discussed separately in Chapter 4.

3.3.1. Mesh Movement

In the case of the UL formulation, the computational mesh is moved to a new position at the end of each time/load increment. This task is trivial for formulations in which mesh vertices coincide with computational points; however, this is not the case for the cell-centred formulation. Generally, we can distinguish two approaches in resolving mesh movement. The first approach, proposed in [100–102], updates mesh geometry using the calculated kinematic quantities, while the second approach defines the interpolation procedure to obtain the displacement of mesh vertices [2,23,91,97]. In this thesis, the discrete displacement field defined at cell centres is interpolated to mesh vertices using the interpolation method described below.

The linear distribution of variable $\boldsymbol{\phi}$ around reference position $\mathbf{r}_{ref}$ with reference value $\boldsymbol{\phi}_{ref}$ is:

$$\boldsymbol{\phi}(\mathbf{r}) = \boldsymbol{\phi}_{ref} + \mathbf{c} \cdot \Delta \mathbf{r} = \boldsymbol{\phi}_{ref} + \mathbf{c} \cdot (\mathbf{r} - \mathbf{r}_{ref}), \qquad (3.31)$$

where $\mathbf{c}$ is the unknown coefficient vector. For mesh vertex p surrounded by its neighbouring cell-centres pN, the reference position and the reference value are obtained by the arithmetic average of the neighbouring values. The calculation of

the unknown coefficient vector is done using the ordinary (unweighted) least-squares method [16, 50, 97]:

$$\mathbf{c} = [(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}] \cdot \boldsymbol{\varphi}, \tag{3.32}$$

where each row of matrix $\mathbf{X}$ and vector $\boldsymbol{\varphi}$ are referred to one of the neighbouring cell-centres pN. The rows in vector $\boldsymbol{\varphi}$ are defined as difference $\phi_{pN} - \phi_{ref}$, while the rows in matrix $\mathbf{X}$ are constructed by the components of difference vector $\mathbf{r}_{pN} - \mathbf{r}_{ref}$. The size of matrix $\mathbf{X}$ is $pN \times 3$; therefore the inverse of normal matrix $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$ is cheap since it is done on the symmetric 3×3 matrix. For the boundary vertices, to ensure the invertibility of the normal matrix, the values at the boundary face centres are also included in the interpolation molecule, as shown in Fig. 3.3. If necessary, Eq. (3.32) can be extended to the weighted form by considering different weighting factors for each pN, as shown in [46, 91].
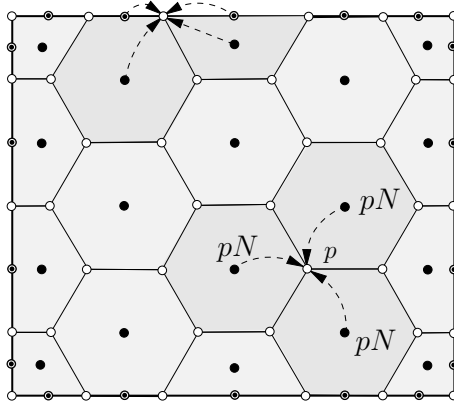
Figure 3.3. The cell to point interpolation: interior and boundary computational molecule for least square interpolation

An alternative approach to the least-square method is interpolation using the inverse distance method [50]:

$$\phi_p = \frac{\displaystyle\sum_{pN} \omega_{pN}\phi_{pN}}{\displaystyle\sum_{pN} \omega_{pN}}, \tag{3.33}$$

where weighting factors ω_{pN} are defined as:

$$\omega_{pN} = \frac{1}{||\mathbf{r}_p - \mathbf{r}_{pN}||}. \tag{3.34}$$

Eq. (3.33) in enhanced form is obtained by including additional terms from the

Taylor series expansion [103]:

$$\phi_p = \frac{\displaystyle\sum_{pN} \omega_{pN} \left[\boldsymbol{\phi}_{pN} + (\mathbf{r}_p - \mathbf{r}_N) \cdot \boldsymbol{\phi}_{pN} + \frac{1}{2}(\mathbf{r}_p - \mathbf{r}_N)^2 \mathbin{\vdots} \boldsymbol{\phi}_{pN} \right]}{\displaystyle\sum_{pN} \omega_{pN}}. \tag{3.35}$$

The inverse distance method allows for efficient interpolation, but it is the less accurate method for the non-orthogonal meshes compared to the least square method, as shown in [50, 97]. The interpolation based on the least-square method maintains the second-order accuracy irrespective of the mesh quality, which is an important feature since in the large deformation cases the mesh quality can degrade during deformation, i.e. an initially perfectly orthogonal mesh can become a highly non-orthogonal mesh.

3.3.2. Alternative Solution and Discretisation Procedures

Block-Coupled Methodology

The main disadvantage of the segregated solution procedure is reduced efficiency while resolving the strong inter-component coupling. To demonstrate the impact of coupling intensity on convergence rate, a simple example of a 2D beam fixed at one end and loaded at the other can be considered, see Fig. 3.4. In the case of axial load, stresses exist only in one direction (except in the vicinity of the fixed end), whereas in the case of tangential load and beam bending, the beam exhibits stresses in both directions. Accordingly, these scenarios are used to represent weak and strong inter-component coupling, respectively. Figure 3.4 shows the convergence history for these scenarios. One can see that in the case of strong inter-component coupling the convergence is drastically affected and significantly more outer iterations are needed to obtain an acceptable level of convergence.

To overcome this drawback, the inter-equation coupling can be implicitly treated using the block-coupled solution methodology [42, 43]. Such solution methodology allows for the implicit treatment of all inter-component terms. In particular, this means that the above example can be solved in one iteration, without performing an outer iteration loop. In contrast to the segregated procedure, assembled matrix $[A]$ is loosely diagonally dominant and calculation is done either by direct solvers or appropriate iterative solvers. Further, the matrix coefficients are represented with second order tensors instead of scalars; therefore the solving of the matrix requires

more storage memory. Nevertheless, less memory is required if compared to the low order FE method [42]. Lastly, the results show exceptional efficiency and further efforts are made to extend the methodology to non-linear constitutive relations [44].
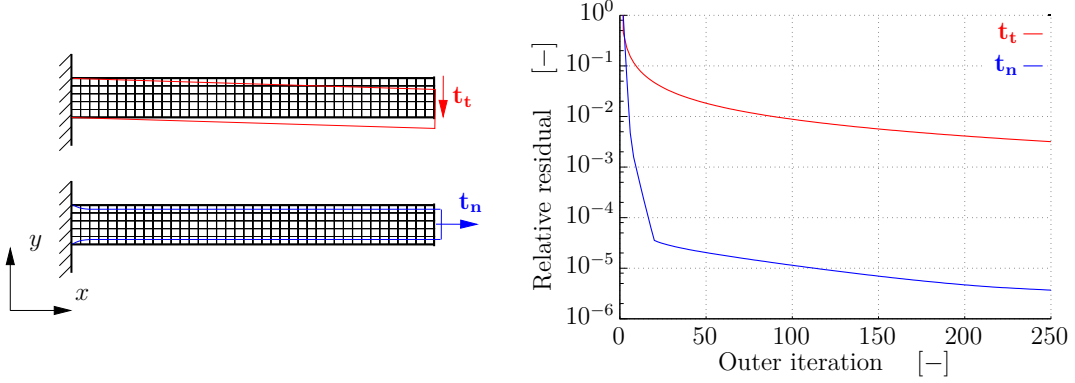
Figure 3.4. Convergence history for beam loaded axially and tangential at the right end

Pressure-Displacement Formulation

Formulating the governing momentum balance with the displacement variable as the only dependent variable has limitations in modelling incompressible materials. Specifically, the formulation exhibits efficiency degradation and volumetric locking while approaching Poisson's ratio of 0.5. To overcome this, a pressure-displacement (mixed) formulation is successfully used [19, 20, 104, 105]. By formulating the constitutive relation via two unknowns (displacement and pressure), a kinematic condition of incompressibility must be introduced to be able to solve the system:

$$\int_\Omega \boldsymbol{\nabla} \cdot \mathbf{u} \, \mathrm{d}\Omega = \int_\Gamma \mathbf{n} \cdot \mathbf{u} \, \mathrm{d}\Gamma = 0. \tag{3.36}$$

Since the kinematic condition and the momentum balance are similar to the pressure velocity coupling in fluid dynamics in form, the SIMPLE-type algorithm is utilised to solve the system. It should be noted that the pressure-displacement formulation can also be applied to modelling linear-elastic compressible materials [20], without impairing the efficiency. Also, it can be extended to consider non-linear constitutive relations [19, 106]. The block-coupled methodology can also be used for the pressure-displacement formulation in order to improve efficiency [107].

Gradient Calculation

In the FV solid mechanics literature the least-square method is the most commonly employed approach in the cell centre gradient calculation [4]. Since the discretisation procedure does not mandate the calculation method, the cell centre gradient can instead be calculated using any other second-order approach [42,91,108,109]. For example, the gradient can be calculated using the Gauss integral theorem:

$$(\boldsymbol{\nabla}\phi)_P = \frac{1}{\Omega_P} \sum_f \boldsymbol{\Gamma}_f \boldsymbol{\phi}_f, \tag{3.37}$$

where $\boldsymbol{\phi}_f$ is the face centre value calculated by linear interpolation between neighbouring cell-centre values:

$$\boldsymbol{\phi}_f \approx \gamma_x \boldsymbol{\phi}_P + (1 - \gamma_x)\boldsymbol{\phi}_N. \tag{3.38}$$

Another option, also based on the Gauss integral theorem, is the point based version [91], in which each face is decomposed into k triangles and the summation is performed using triangles centres values $\boldsymbol{\phi}_k$:

$$(\boldsymbol{\nabla}\phi)_P = \frac{1}{\Omega_P} \sum_f \sum_k \boldsymbol{\Gamma}_k \boldsymbol{\phi}_k, \tag{3.39}$$

where $\boldsymbol{\phi}_k$ is reconstructed from the values at mesh points calculated by the least square method. Further, instead of linear interpolation, the gradient at the face centre can be calculated by the following decomposition:

$$(\boldsymbol{\nabla}\phi)_f = \boldsymbol{\nabla}_t\boldsymbol{\phi}_f + \mathbf{n}_f(\mathbf{n}_f \cdot (\boldsymbol{\nabla}\phi)_f), \tag{3.40}$$

where the normal surface gradient is calculated using the central differencing and non-orthogonal correction. Tangential surface gradient $\boldsymbol{\nabla}_t\boldsymbol{\phi}_f$ is calculated using the Green-Gauss theorem for flat polygonal faces, further details can be found in [91, 110].

Thin Structures

Although the discretisation procedure can handle arbitrarily shaped bodies, bodies with a high aspect ratio between length and thickness (thin structures — plates, beams and shells) are inconvenient to discretise with a reasonable number of CVs,

especially if a high degree of accuracy is requested. Accordingly, to avoid high expenses caused by continuum formulation, many authors have successfully used reduced formulations with the implicit cell-centred FV method [111–115]. A comprehensive list of publications regarding this area is available in [4].

3.4. Summary

This chapter covers the implicit cell-centred unstructured FV framework. The framework description is given in a general manner to cover the class of FV structural solvers capable of solving nonlinear solid mechanics problems. The equation discretisation is given for the general form of the momentum balance, followed by the description of three main types of boundary conditions in the solid mechanics. The characteristics of the system of equations are discussed, and the resulting system coefficients for the solver used in this thesis are given separately. Next, an overall summary of the solution procedure is given with special attention given to mesh movement approaches. In the end, alternative discretisation and solution procedures are discussed. These procedures are developed to overcome specific drawbacks, thus this section simultaneously shows the main weaknesses of the method.

4. Contact treatment

4.1. Introduction

This chapter is devoted to the algorithms for the numerical treatment of contact boundaries within the finite volume framework presented in the previous chapter. The content of the chapter is divided into six distinct sections. In the first section, the main aspects of the contact boundary treatment are outlined by presenting the explicit Neumann-Neumann coupling procedure and penalty regularisation of the contact conditions. In general, contact algorithms can be divided into two distinct phases (contact detection and contact resolution), each of which has its algorithm (contact detection algorithm and contact-force computation algorithm). Accordingly, the second section is devoted to the contact detection phase intended for the considered numerical framework. A contact detection algorithm based on the object-oriented bounding box and the advancing front technique is suggested. The third section presents the pointwise contact algorithm [1, 2], in which the penalty based Neumann-Neumann coupling is initially proposed. Its main features are outlined in order to properly address the differences in relation to the newly proposed segment-to-segment algorithm, introduced in the fourth section. In the fourth section, these algorithms are compared using a few carefully selected simple numerical examples. To improve overall efficiency, the implicit version of the Neumann-Neumann coupling is considered in the penultimate section. Using the newly proposed contact algorithm, the equation for implicit boundary gradient is firstly derived on simple 1D contact problems to be subsequently extended to 3D problems. In the last section, the surface smoothing technique using the Nagata interpolation is incorporated within the pointwise contact algorithm and the impact of surface smoothing is investigated using a few numerical examples.

4.1.1. Discretised Contact Boundary

After discretisation of the solution domain, the continuous deformable body is replaced by a finite number of arbitrarily shaped control volumes that fill the domain space interior, without overlapping. The control volumes are bounded by flat

polygonal faces, which results in a piecewise linear approximation of the continuous boundary. Contact interaction occurs at the boundary, meaning that accurate approximation of the boundary plays an important role in imposing contact boundary conditions.

In the contact treatment, two potential problems caused by the piecewise linear approximation can be distinguished:

- The discretised boundary has a discontinuous surface normal field (it is non-differentiable at mesh points and along edges between adjacent boundary faces) which can lead to various numerical and mathematical difficulties.

- For the highly curved boundary, the piecewise linear discretisation can result in large errors in the geometry description. Such an error consequently affects the achievable level of accuracy.

In this thesis, the standard notation from the computational contact mechanics is adopted and contact boundaries (surfaces) $\Gamma_c^{(1)}$ and $\Gamma_c^{(2)}$ are denoted as slave (concator) and master (target), as shown in Fig. 4.1. The same notation is used for all figures: computational points located at the cell centre are denoted with $\bullet$, the mesh points with $\circ$ and the boundary face centre with $\odot$.
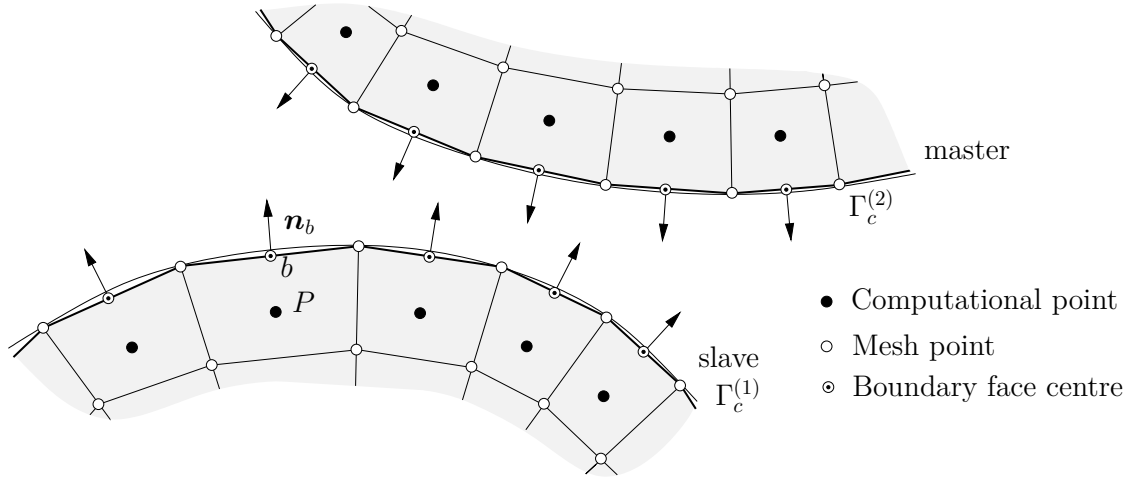
Figure 4.1. Piecewise linear discretisation of contact boundaries and adopted notation

Contact interface

The contact interface can be considered as a particular class of interface because of its underlying physics. The same as for other types of interfaces, the complexity

of the numerical treatment is interrelated to the interface discretisation type. Accordingly, it is possible to distinguish two types of interface discretisation: conformal and non-conformal.

In the conformal interface discretisation (shown in Fig. 4.2) the surface meshes of the contacting bodies coincide. This significantly simplifies the implementation of the contact algorithm; however, it is clear that such a type of discretisation is not suitable for contact problems with finite deformations and large relative sliding. Although it can be used in cases of small deformations and infinitely small relative sliding, ensuring the conformal discretisation during the mesh generation can be a tough and impractical task.
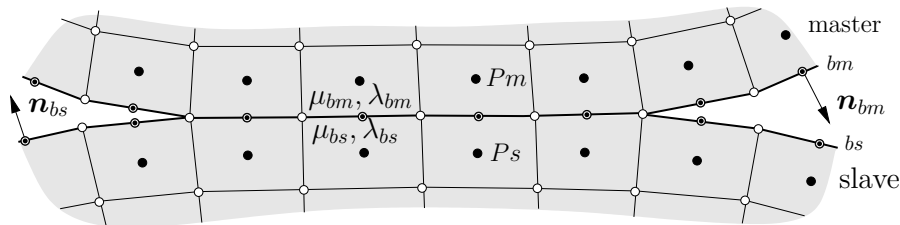
Figure 4.2. Conformal discretisation at contact interface

In the case of the non-conformal interface discretisation, different surface mesh topology is allowed on the interface, as shown in Fig. 4.3. The generality and applicability of the contact algorithm lie in the ability to handle the non-conformal discretisation. Accordingly, the numerical implementation capable of handling the non-conformal interface is the major priority.
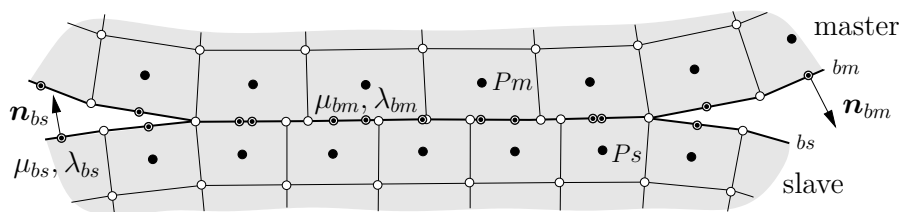
Figure 4.3. Non-conformal discretisation at contact interface

4.1.2. Penalty Method

The unknown components of the contact traction can be treated with a regularisation scheme to obtain their dependence on the corresponding displacement-based contact kinematic measures. An alternative approach would be to treat the contact traction components as independent variables, in addition to the displacement field.

4. Contact treatment

The penalty method is a widely used regularisation method in the computational contact mechanics literature as it is a robust and simple method with vivid physical interpretation [5, 63, 64]. The underlying principle of the penalty method is the penalization of the violation of the contact constraints, with the penalization magnitude proportional to the violation measure. Its mechanical interpretation can be seen as a distributed series of springs on the contact interface, which are activated if the non-penetration condition is violated, schematically presented in Fig. 4.4.

slave

master

a)   b)   c)
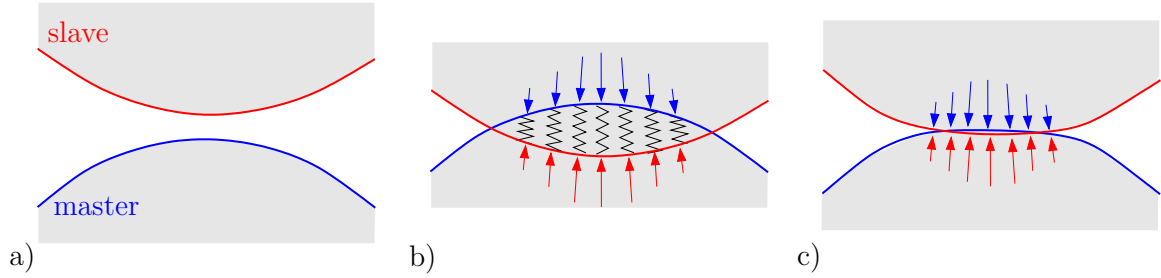
Figure 4.4. Graphical interpretation of penalty method using spring analogy: a) initial configuration, b) violation of non-penetrability and c) equilibrium state [5, 6]

Normal conditions

By applying the penalty method in a linear form, the normal contact pressure is linearly dependent on the normal gap:

$$p_n(g_n) = \varepsilon_n \langle -g_n \rangle = \begin{cases} \varepsilon_n g_n & g_n \leqslant 0, \quad \text{contact} \\ 0 & g_n > 0, \quad \text{no contact} \end{cases}, \tag{4.1}$$

where ε_n is the penalty parameter (also called penalty factor or penalty stiffness). The relationship between normal gap and pressure is described with discontinuous functions, therefore the Macaulay brackets are used for the compact notation:

$$\langle \phi \rangle = \frac{|\phi| + \phi}{2} = \begin{cases} \phi & \text{if } \phi \geqslant 0 \\ 0 & \text{if } \phi < 0 \end{cases}. \tag{4.2}$$

The graphical representation of Eq. (4.1), i.e. penalty regularised normal contact conditions, are shown in Fig. 4.5. As shown, the violation of the non-penetrability condition depends on the penalty parameter and an exact fulfilment is achievable only for $\varepsilon_n \to \infty$. Accordingly, the value of the penalty parameter is proportional to achievable accuracy. However, the numerical implementation does not allow for

large values of penalty parameters, as they lead to stability problems [63,64].
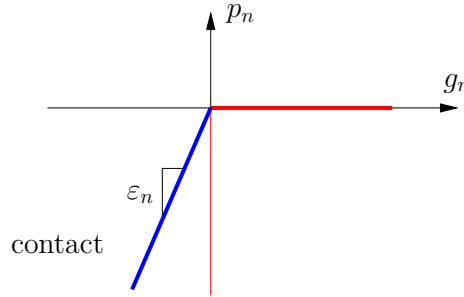
Figure 4.5. Graphical representation of regularised normal contact conditions using penalty method

Tangential conditions

By applying the penalty method to classical Coulomb's law one obtains:

$$
\mathbf{t}_t = \begin{cases} \varepsilon_t \Delta \mathbf{g}_t & \varepsilon_t ||\Delta \mathbf{g}_t|| < \mu |p_n|, \quad \text{stick} \\ \mu |p_n| \dfrac{\Delta \mathbf{g}_t}{||\Delta \mathbf{g}_t||} & \varepsilon_t ||\Delta \mathbf{g}_t|| \geqslant \mu |p_n|, \quad \text{slip} \end{cases}, \tag{4.3}
$$

where ε_t is the tangential penalty parameter which does not need to be equal to the normal penalty parameter [64]. With the regularisation of the Coulomb's law, the stick contact regime is violated and the relative tangential motion, called elastic slip, is allowed for values inside the Coulomb's cone. The violation of the stick regime is necessary to obtain the needed dependency between tangential traction and relative sliding. In the regularised Coulomb's law, the tangential traction for the stick regime is linearly dependent on the relative tangential motion, as shown in Fig. 4.6. To better mimic the real frictional response any other function type can be applied (arctangent functions, hyperbolic tangents) [63]. The same as for the normal direction, the exact representation of the unregularised Coulomb's law can be obtained for $\varepsilon_t \to \infty$. Nonetheless, the regularised Coulomb's law is more physical as elastic slip can be used to model elastic deformation of asperities [63]. On the other hand, for the contact problems with dominant sliding at the contact interface, chasing higher values of the normal penalty parameter is a priority as the sliding traction is a function of the normal pressure.
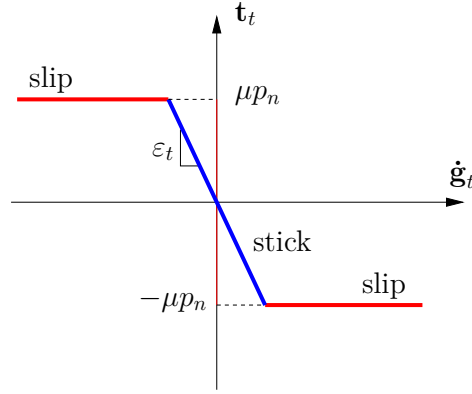
Figure 4.6. Graphical representation of regularised Coulomb's friction law using penalty method

Penalty parameter

In the penalty method, the degree of violation of contact conditions i.e. the resulting level of accuracy is controlled by the penalty parameter value. This represents the main disadvantage of the penalty method which requires special attention since the resulting penetration can be of the same order of magnitude as displacement.

The penalty parameter is the user-defined value, and setting its value to achieve an acceptable degree of accuracy while ensuring stability can be a cumbersome task, especially for an inexperienced user. To alleviate this issue, for the initial guess of the penalty parameters, it is possible to use the following equations [2, 116]:

$$\varepsilon_n = \frac{f_n^{scale} K \, \Gamma_c^2}{\Omega_c}, \tag{4.4a}$$

$$\varepsilon_t = \frac{f_t^{scale} G \, \Gamma_c^2}{\Omega_c}, \tag{4.4b}$$

where f_n^{scale} and f_t^{scale} are the scale factors for the normal and the tangential penalty parameter, respectively. Γ_c is the average area of the contact boundary faces and Ω_c is the average volume of the contacting cells. The scale factors are typically set to 1 and they are subsequently corrected upon resulting penetration or elastic slip at the contact interface. By defining correction criteria, the scale factors can be corrected after each time-step, hence avoiding large penetrations in the cases where contact pressure evolves drastically with time.

4.1.3. Explicit Neumann-Neumann Coupling Procedure

In the explicit Neumann-Neumann coupling procedure the contact interaction is resolved in a deferred correction manner, which allows an efficient and simple treatment of nonlinearities within the iterative solution procedure. Contact traction, prescribed on both contact boundaries, is explicitly calculated using the penalty method. To ensure convergence of the penalty based coupling, the prescribed contact traction is under-relaxed. The overall coupling procedure can be divided into the following steps:

i) Use the newly calculated value of displacement increment to update the contact boundary position.

ii) Perform contact detection to assemble contact pairs (contact detection algorithm).

iii) Upon resulting penetration and relative sliding between the deformable bodies (g_n and $\Delta \mathbf{g}_t$), calculate the corresponding components of the contact traction (contact-force calculation algorithm).

iv) Prescribe contact traction using the accumulated values (from the previous iteration) and the newly calculated values (from step iii)).

For the calculation of the contact traction, it is possible to distinguish two approaches which will here be denoted as asymmetric and symmetric. In the asymmetric approach, contact traction and contact quantities are calculated directly on one of the boundaries; therefore, it is necessary to perform interpolation to transmit contact traction onto the adjacent boundary. In the symmetric approach, contact traction is simultaneously calculated for both sides. The pointwise contact algorithm [1], described in Section 4.3, relies on an asymmetric approach and the area-weighted GGI interpolation is used to transmit contact traction from slave to master boundary. The proposed segment-to-segment contact algorithm, described in Section 4.4, can be characterised as mixed type since normal contact traction is simultaneously calculated while tangential contact traction is calculated separately on both surfaces.

Enforcement of the contact constraints

The penalty method allows robust and efficient contact treatment. However, these characteristics can be attributed to the approximate enforcement of the contact

49

constraints. In other words, the user-controlled violation of the contact constraints introduces error in the solution, hence causing the main drawback of the method.

The final approximate enforcement of the contact conditions is achieved gradually during the iterative procedure by performing augmentation of the contact traction until convergence is met. The augmentation is performed using the under-relaxation method, although it can be performed by any other similar method.

Contact boundary update

The calculation of the kinematic contact quantities requires the update of the contact boundary position within each outer iteration. The update is performed using a recently calculated displacement field, available at the cell centres and the boundary face centres. Accordingly, it can be performed using different interpolation molecules, as shown in Fig. 4.7.

a)   b)
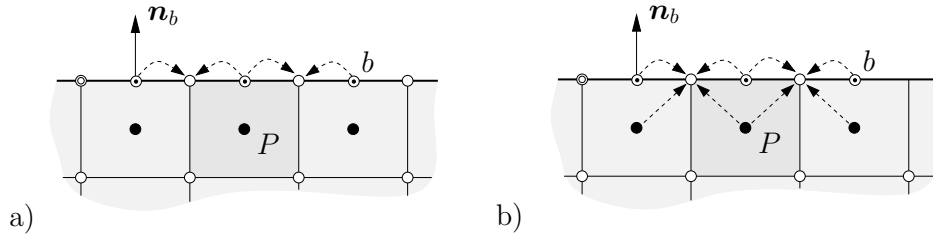
Figure 4.7. Different interpolation molecules: a) surface interpolation molecule, b) mixed molecule

The usage of the surface interpolation molecule in combination with the inverse distance method provides a very efficient interpolation method. However, numerical tests show that the usage of a larger (mixed) interpolation molecule in conjunction with the least-square method is a more suitable approach, as it can improve the convergence rate due to higher accuracy. Also, the normal matrix in Eq. (3.32) is computed only for the first iteration, thus ensuring interpolation efficiency.

Force-loading scenario

During the iterative procedure, the prescribed contact traction does not match the sum of other forces acting on the body resulting in global force imbalance. If the body does not have a prescribed displacement at some location, the global imbalance can lead to spurious displacements, especially if inertia is omitted. Such a problem exists in the force-loading scenario, in which the deformation of the contact body is

governed only by the prescribed boundary tractions. To solve such scenarios using the explicit contact coupling, the system of equations must be heavily under-relaxed, resulting in a very slow convergence rate.

A possible remedy is to prescribe boundary displacement instead of force and to correct it until the targeted force is achieved. The correction can be performed within the time step by using the relative residual as a measure for global force balance. The correction of the displacement is calculated using the secant method (the quasi-Newton method). The described procedure, summarised in Algorithm 1, was used to solve force loading examples presented in Chapter 5 (examples 5.2.2 and 5.2.4). Results show that the targeted force can be achieved in several iterations for a valid guess of initial displacement.

Algorithm 1. Update of the displacement boundary by using secant method.

1: **if** integrated force is not within tolerance of targeted force F_t **then**
2:     **if** relative residual is below prescribed tolerance **then**
3:         Integrate stress at boundary and calculate force $F^{(i)}$
4:         Calculate force difference $\Delta F^{(i)} = F_t - F^{(i)}$
5:         Calculate derivative approximation $S = (F^{(i-1)} - F^{(i)})/(u^{(i)} - u^{(i-1)})$
6:         Calculate and prescribe new displacement $u^{(i+1)} = u^{(i)} + \Delta F^{(i)}/S$
7:     **end if**
8: **end if**

Generalised grid interface interpolation

The generalised grid interface (GGI) was developed to overcome many modelling challenges that primarily arise in turbomachinery simulations [3]. It allows an implicit treatment of coupled internal boundaries, sharing the conformal or non-conformal interface. The implicit treatment, performed on a discretisation level, maintains second-order accuracy and mass conservation across the interface. From the basic requirement of coupling static meshes, the GGI is subsequently extended to handle non-conformal cyclic patches, sliding interface and mixing plane interface [117, 118]. The GGI is also used for other specific types of interfaces (contact or fluid-solid interface [91]) to explicitly transmit traction between surfaces.

In its basis, the GGI is an area weighted interpolation with geometrically defined interpolation molecules. For each surface face j, on the master side, the value at face-centre $\phi_{m,j}$ is calculated using corresponding face-centre values of the slave surface faces:

$$\phi_{m,j} = \sum_k \omega_{j,k} \phi_{s,k}, \tag{4.5}$$

and similarly for slave face k:

$$\phi_{s,k} = \sum_j \omega_{k,j} \phi_{m,j}, \tag{4.6}$$

where $\omega_{j,k}$ and $\omega_{k,j}$ are the weighting factors defined as:

$$\omega_{j,k} = \frac{\Gamma_{m,j} \cap \Gamma_{s,k}}{\Gamma_{m,j}}, \qquad \omega_{k,j} = \frac{\Gamma_{s,k} \cap \Gamma_{m,j}}{\Gamma_{m,j}}. \tag{4.7}$$

Weighting factors in Eq. (4.7) are representing the ratio between the intersection area and the overall face area, see Fig 4.8. It should be noted that taking the inverse of the weights calculated on the master side will not result in weights for the slave side:

$$\omega_{j,k} \neq \frac{1}{\omega_{k,j}} \tag{4.8}$$

The conservativeness of the interpolation is automatically assured since the sum of all weighting factors results in:

$$\sum_j \omega_{k,j} = 1, \qquad \sum_k \omega_{j,k} = 1. \tag{4.9}$$

Eq. (4.9) is violated in the case of non-planar interface surfaces or in the case of hanging faces with partial covering. A possible remedy in such cases is to perform a correction of the interpolation factors by scaling their values proportionally.
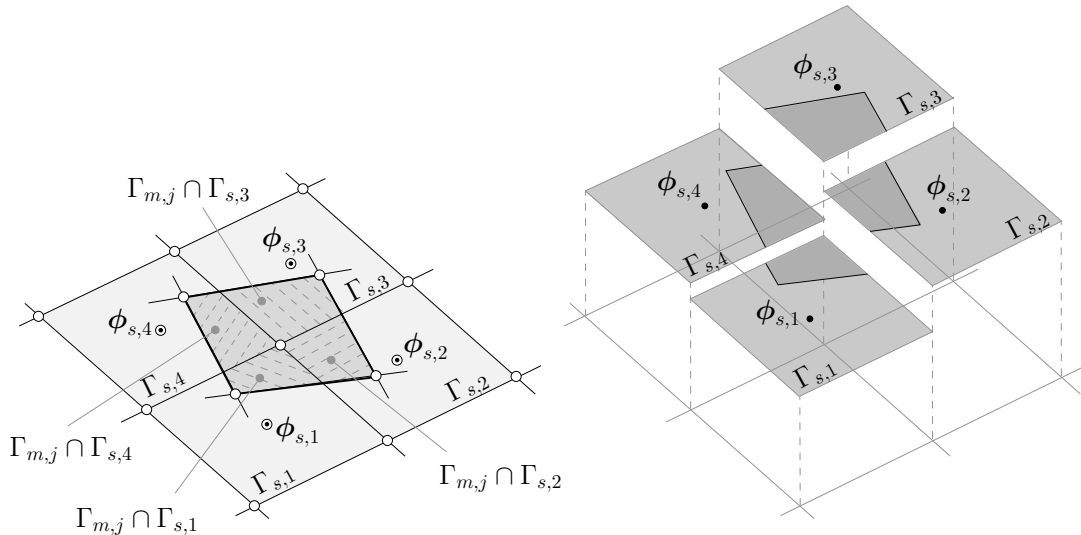
Figure 4.8. Segmentation of master face using corresponding slave faces. The right side of the figure schematically represents Eq. (4.5) — the intersection area (darker shaded) can be interpreted as the integration area

Further details involving the numerical implementation of the GGI interpolation are given within the description of the segment-to-segment contact algorithm in Section 4.4.

4.2.   Contact detection

The first phase of the contact algorithm is contact detection, whose aim is to construct contact pairs between two discretised surfaces. Generally, it is possible to distinguish two different approaches in contact detection [5]:

- Implicit approach: based on the current values it is necessary to estimate potential contact pairs where penetration is expected.

- Explicit approach: contact pairs are assembled upon achieved penetration.

Further, the detection algorithm can be decomposed into two phases [63]:

- Global detection: based on geometrical consideration, detect potential (proximate) contact pairs between slave and master surface meshes.

- Local detection: using potential contact pairs, find pairs with intersection and remove falsely reported pairs.

Although the finite volume framework relies on the implicit equation discretisation, the explicit nature of the contact coupling requires contact detection of the explicit type. In this thesis, the implicit contact coupling is also considered; however, its derivation is also based on explicit contact detection. For large deformation contact problems considered in this thesis the contact interface continuously evolves during the iterative procedure, meaning that contact detection is performed for each iteration. The number of outer iterations can reach up to a few thousand, hence the efficiency of contact detection is of great importance. Therefore, contact detection can present a bottleneck for an efficient implementation of the contact algorithm. The exceptions are contact problems with a negligible relative motion between surfaces, for which efficiency of contact detection is of less priority as the contact detection can be performed only once.

Contact detection, or the so-called collision detection in computer science [119–121], is of paramount importance in many areas, such as computer graphics visualisation, virtual reality, molecular modelling, robotics and many others. Due to

the versatility of its application, it is under intensive research and the corresponding literature is highly abundant. In the field of computational contact mechanics, contact detection algorithms are successfully adopted from computer science. The right choice of the contact detection algorithm depends on the numerical framework and contact treatment methodology [88]. In case the contact pair is a combination of node and surface segment (NTS discretisation), a typical choice among authors is the bucket sorting algorithm [122] or its improved version called position code algorithm [123]. For the STS discretisation, the definition of the contact pair is different, and contact algorithms are based on bounding volumes hierarchies (bounding volume tree) [124, 125]. The most common choices for bounding volumes are discrete orientation polytopes [72] and oriented bounding boxes [126].

In the remaining part of this section, further insight regarding contact detection is given and a contact detection algorithm is presented. The presented contact detection algorithm is intended for the proposed segment-to-segment contact-force calculation method. The contact detection for self-contact problems is not considered.

4.2.1. Global Contact Detection

The task of finding proximate contact pairs between two discretised surfaces is based on the encapsulation of the surface elements (faces) in the bounding volumes with a simple geometric shape, allowing a fast and inexpensive overlap test. The higher geometric complexity of the bounding volume results in a tighter geometry encapsulation and consequently a more accurate overlap detection, but also in a higher computational cost of the overlap test. Depending on the application, the choice of the bounding volume is a trade-off between accuracy and efficiency.

The geometric shape of the bounding volume can be defined in several ways. Figure 4.9 shows some of the most commonly used bounding volumes: bounding spheres (BSs), axis-aligned bounding boxes (AABBs) and oriented bounding boxes (OBBs). The BS is the simplest and most efficient in terms of overlap tests; however, it results in many false contact pairs, especially when bounding the faces with an aspect ratio higher than the unity. The AABB and OBB rely on the same type of bounding volume (cuboid) with differences in the choice of the coordinate system. The surfaces of the AABB are parallel to the global coordinate system, while the OBB uses the local coordinate system for its orientation. The usage of the local coordinate system results in a higher computational cost of construction and overlap

test. On the other hand, it results in better detection accuracy due to higher tightness, as shown in Fig. 4.9.
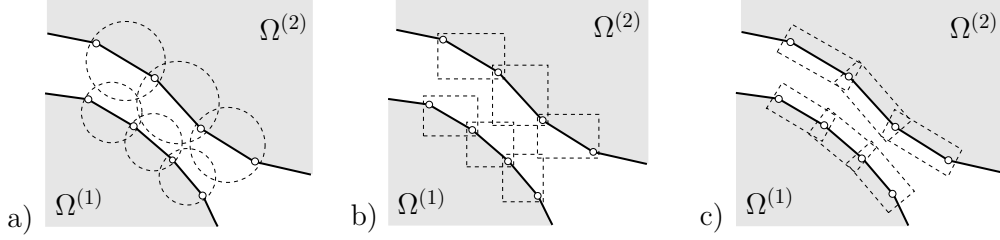
Figure 4.9. Bounding volume types in 2D [7]: a) BSs, b) AABBs, c) OBBs

This thesis employs OBBs for the definition of bounding volumes. The usage of the OBBs results in a more accurate global detection with a lower number of contact pairs than the detection with BS or AABB [72, 119]. Although it results in a more computationally expensive approach, it also minimizes the number of false contact pairs, therefore the efficiency of the local detection can be significantly improved.

Construction of oriented bounding boxes

The definition of an OBB is achieved via a set of three parameters: centre point vector $\mathbf{r}_c$, orientation matrix $\mathbf{R}_a$ and extend vector $\mathbf{a}$. By default, these parameters are calculated by assembling the covariance matrix, whose eigenvectors define the orientation matrix [127]. When applied to point clouds, such an approach results in a near-optimal solution, however for symmetrically distributed points or coplanar points the resulting covariance matrix cannot give a unique orientation of OBB [119]. Since such a problem exists in the current application, a different construction technique is proposed.

With known boundary face normal $\mathbf{n}_b$ the orientation of the OBB is reduced to a 2D problem of finding the minimum-area encasing the rectangle, see Fig. 4.10 a). Such a problem is solvable in linear time using the rotating calipers method [128]. For the height of the OBB, the boundary normal is used for extrusion in both directions:

$$e^+ = 0.25\bar{h}, \qquad e^- = \bar{h}, \tag{4.10}$$

where e^+ and e^- are the extrusion distances in positive and negative normal directions, respectively. $\bar{h}$ denotes the average height of cells at the contact boundary, and here it is used as maximum detectable penetration. To avoid any potential robustness issues, the resulting rectangle is inflated by 5% and also extruded in the

positive direction, as shown in Fig. 4.10.
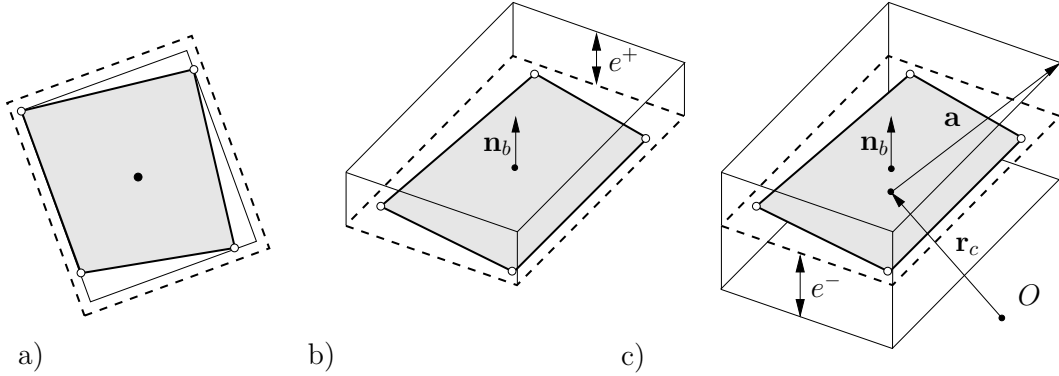
Figure 4.10. OBB construction steps [7]: a) inflated face minimum-area encasing rectangle, b) extrusion in positive normal direction, c) extrusion in negative normal direction

4.2.2. Contact Detection Algorithm

The straightforward implementation of contact detection is all-to-all detection, referred to as naive brute force detection. For n_s faces on the slave surface and n_m on the master, such an approach results in $O(n_m n_s)$ overlap tests, which is an unacceptable computational cost.

Irrespective of the algorithm implementation, the cost of contact detection can be minimized by reducing the portion of the contact boundary subjected to detection. This can be easily achieved during the run-time by prescribing a contact zone inside which the detection is performed. The size and location of the contact zone are determined as the intersection of master and slave surface bounding volumes, as shown in Fig. 4.11.
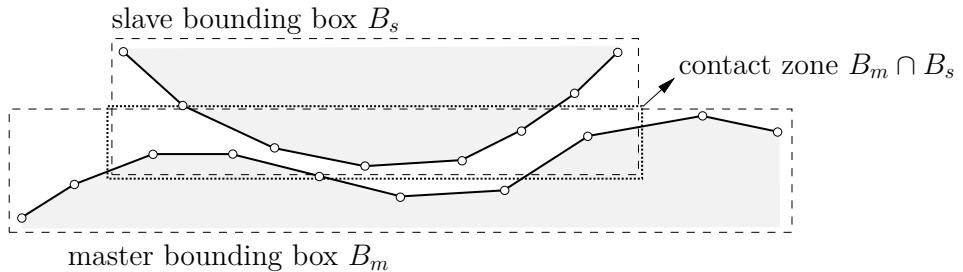
Figure 4.11. Determination of contact zone using AABBs [5]

To perform contact detection efficiently, the detection algorithm proposed in this thesis exploits the available mesh topology information. It relies on the available

face-to-face addressing, which allows the usage of the advancing front technique. Figure 4.12 illustrates the working principle of such an approach — the current master bounding volume (for which pairs are searched) is coloured in red, while the neighbouring slave bounding volumes are coloured in black. For better clarity, the bounding volumes of the slave surfaces do not overlap, which is not true for a real implementation.
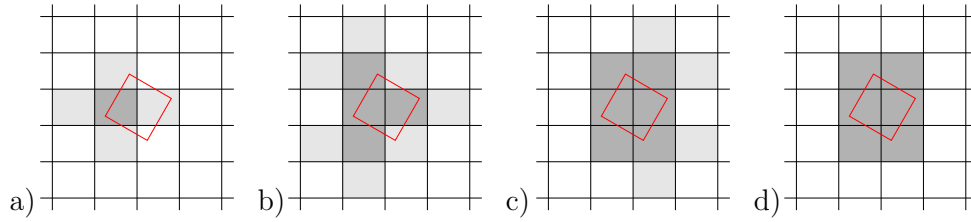
Figure 4.12. Steps of contact detection using advancing front technique, illustrated in 2D. Overlapping bounding volumes are coloured in grey, whereas their neighbouring volumes are light grey

To start the detection using the advancing front technique, one overlapping bounding volume (seed) must be supplied, Fig. 4.12 a). Using face-to-face addressing, the neighbourhood of the supplied seed is reported and checked for overlap, Fig. 4.12 b). Only the neighbours of the overlapping bounding volumes are further reported and checked — care must be taken to avoid duplicate tests of the processed volumes. The algorithm proceeds until there is no available neighbour to check.

If supplied seed is available, the computational cost of the described algorithm is $O(k)$, where k is the overall number of the reported neighbours [59]. Since detection is invoked for each outer iteration, there is no abrupt change in contact pairs, thus previous contact pairs are supplied to the algorithm as a seed, as illustrated in Fig. 4.13. By doing this, an admissible computational cost is ensured.
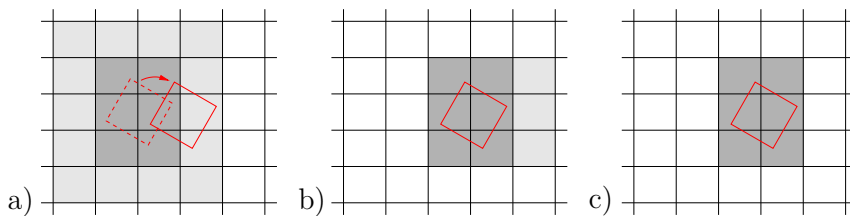
Figure 4.13. Usage of previous contact pairs as supplied seed for contact detection algorithm. See Fig. 4.12 for colour explanation

In some scenarios, the supplied seed does not contain any overlapping contact pair and the initial seed needs to be found in order to conduct the detection. The

simplest solution, adopted here, is to perform all-to-all detection until the initial seed is found, meaning that the computational cost for such a scenario is increased to $O(n_s + k)$. Alternatively, the supplied seed can be extended with its neighbourhood. Although such an approach would be more efficient, it is less robust since it cannot ensure non-existence of the overlapping candidate.

For the iterations near the converged solution, there is no change in contact pairs, thus recalculation of the contact pairs is not necessary. Accordingly, further efficiency improvements are possible if their update stops after some predefined solution relative tolerance, say $5 \cdot 10^{-6}$.

Algorithm 2 gives the pseudo-code for the described contact detection procedure.

Algorithm 2. Contact detection algorithm (global phase)

1: **if** the new time step is reached **then**
2:     Initialise contact zone   ▷ Fig. 4.11
3: **end if**
4: **for all** master and slave faces within contact zone **do**
5:     Construct bounding box   ▷ Section 4.2.1.
6: **end for**
7: **for all** master bounding boxes **do**
8:     **if** the list of the previous candidates exists for current master face **then**
9:         Perform contact detection using advancing front technique   ▷ Fig. 4.13 a) – c).
10:        **if** conducted search find any candidate **then**
11:            **continue** to the next master bounding box
12:        **end if**
13:    **end if**
14:    Perform brute force search to find the first candidate (i.e. the seed)   ▷ Fig. 4.12 a).
15:    **if** seed is found **then**
16:        Perform contact detection using advancing front technique   ▷ Fig. 4.12 b) – d).
17:    **end if**
18: **end for**

The proposed Algorithm 2 is general in terms of the choice of the bounding volume — the different choice of the bounding volume affects the memory requirements and the type of overlap test. For the OBBs used here the implementation of the overlap test is based on the 3D interpretation of the separating axis theorem (SAT). Details regarding the implementation of the overlap test are available in [119, 121].

The SAT is also used for the local phase of contact detection in which false contact pairs are filtered out by performing 2D overlap tests. For each master face, available slave pairs are projected onto its plane and 2D implementation of the SAT algorithm is used to check their overlap.

4.3. Pointwise Contact Algorithm

This section gives an outline of the main aspects of the contact algorithm proposed by [1,2,50]. Here, it is named the pointwise contact algorithm (PCA) because the violation of the impenetrability conditions are checked on the points (vertices) of the choosen surface. The algorithm was first introduced in [1,50], where it was used to solve frictionless contact problems with linear-elastic solids. Subsequently, in [2], it was extended to consider frictional large deformation contact problems. The introduction of this algorithm made a breakthrough regarding contact modelling using the finite volume method since it was the first one reported to be able to provide a satisfactory level of robustness and accuracy in solving challenging contact problems.

The pointwise contact algorithm can be classified as a single-pass algorithm — contact quantities and the resulting components of the contact traction are calculated on the assigned slave surface. The normal component of the contact traction is calculated at the slave surface points and it is subsequently interpolated at the slave face centre, denoted with bs:

$$(\mathbf{t}_n)_{bs} = -\frac{1}{||\boldsymbol{\Gamma}_{bs}||} \frac{\sum\limits_p \omega_p \varepsilon_n |g_n|}{\sum\limits_p \omega_p} \mathbf{n}_{bs}, \tag{4.11}$$

where $\boldsymbol{\Gamma}_{bs}$ is the slave face area vector, $\mathbf{n}_{bs}$ is the slave face normal vector and ω_p is the weighting factor of each face point p. The weighting factor is set to unity, meaning that each point value has the same contribution in the overall face-centre value. Normal gap value g_n is obtained using the orthogonal projection of the slave point onto corresponding master face, see Eq. (2.34). The tangential (frictional) component of the contact traction is calculated directly at the slave face-centres using Eq. (4.3) with the face-centre incremental slip defined as:

$$(\Delta \mathbf{g}_t)_{bs} = (\mathbf{I} - \mathbf{n}_{bs}\mathbf{n}_{bs}) \boldsymbol{\cdot} (\Delta \mathbf{u}_{bs} - \Delta \hat{\mathbf{u}}_{bm}), \tag{4.12}$$

where $(\mathbf{I} - \mathbf{n}_{bs}\mathbf{n}_{bs})$ is the projection tensor and $\Delta \hat{\mathbf{u}}_{bm}$ is the master displacement increment interpolated to the slave surface face centres. To ensure convergence, the final value of the slave contact traction is obtained after applying under-relaxation

4. Contact treatment

of both traction components (subscript bs is omitted for clarity):

$$\mathbf{t} = \alpha_n \mathbf{t}_n + (1 - \alpha_n)\boldsymbol{\lambda}_n + \alpha_t \mathbf{t}_t + (1 - \alpha_t)\boldsymbol{\lambda}_t, \tag{4.13}$$

where $\boldsymbol{\lambda}_n$ and $\boldsymbol{\lambda}_t$ are the accumulated normal and tangential contact tractions. α_n and α_t are the under-relaxation factors for normal and tangential direction.

The interpolation of the slave surface traction onto the master surface is performed using the GGI surface-to-surface interpolation, presented in Section 4.1.3. The same interpolation is also used to interpolate the master displacement increment in Eq. (4.12). To conclude the description, the main numerical issues of the pointwise contact algorithm are outlined as follows:

i) The calculation of the normal gap at slave points, using the closest point projection (orthogonal projection), introduces the asymmetry in the problem. In other words, the results are dependent on the choice of the master and slave surface definitions.

ii) For the faces in partial contact, Eq. (4.11) is not able to correctly predict the normal contact traction.

iii) Master surface points are allowed to freely penetrate the slave surface, thus their penetration can go undetected. This problem is highlighted in the case of large differences in mesh resolution, as shown in Fig. 4.14 b).

iv) The non-smooth boundaries are causing oscillations of the computed contact force as the slave points are sliding over segments. Also, special attention must be given when calculating the closest point projection as it is not a trivial task due to the discontinuities of the normal field, see Fig. 4.14 a).

v) For the curved contact boundaries, the conservativeness of the GGI interpolation is not enforced (correction of the weighting factors is not admissible).

a)   b)
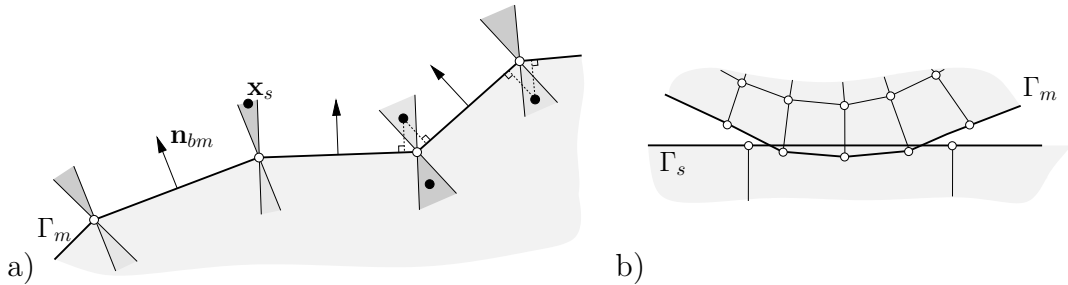
Figure 4.14. Typical problems when penetration is calculated at slave points: a) internal and external blind spots with undefined or multiple solutions of orthogonal projection, b) undetected penetration of master points

60

4.4.   Segment-to-Segment Contact Algorithm

For the non-smooth representation of contact boundaries, the resulting distribution of the gap function is piecewise continuous with derivate discontinuities occurring at the surface points and alongside the face edges. This is illustrated in Fig. 4.15, which shows the distribution of the gap function for the case of contact between non-conformal contact boundaries in 2D settings. As shown, the resulting distribution of the normal gap is composed of C^0-continuous segments with a derivative discontinuity at locations of mesh points of either surface. Accordingly, if the calculation of the normal gap is performed only at the points of one surface, an exact evaluation of the normal gap is not possible.
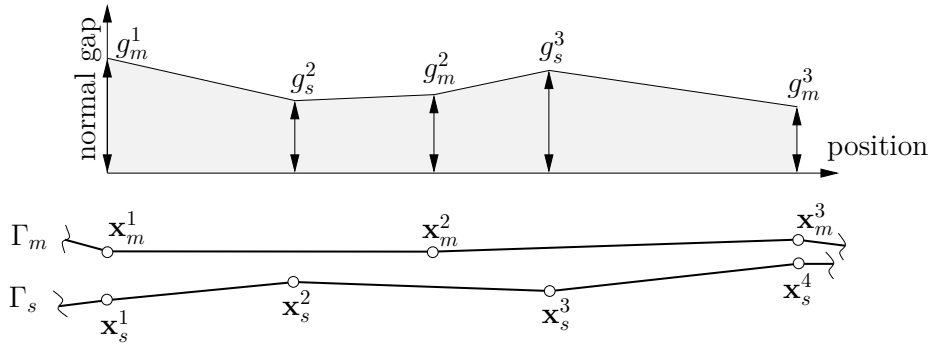
Figure 4.15. Variation of normal gap between two surfaces in 2D (normal gap is measured using vertical direction)

By performing an integration that will take geometrical information of both surfaces into account, an exact evaluation of the normal gap is possible. Therefore, it is necessary to divide the integration area into segments with a continuos variation of the normal gap. The segments are represented as a line or surface elements, depending on problem dimensions, see Fig. 4.16.
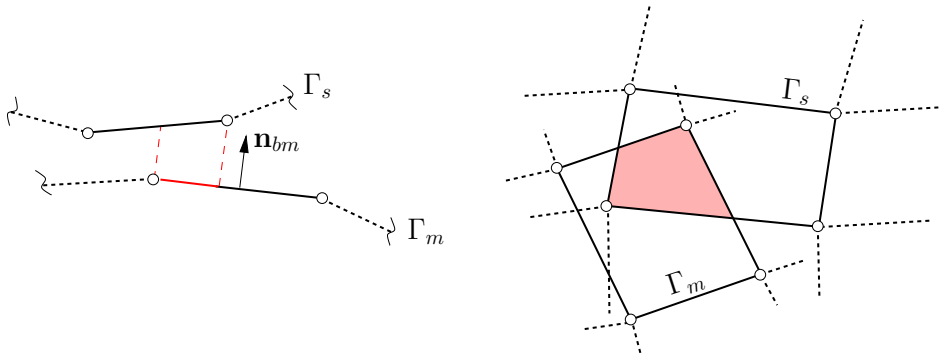
Figure 4.16. 2D and 3D representation of segment domain (coloured in red)

61

4. Contact treatment

To define the segments it is necessary to establish projection rule. Here, the normal vector field of the master surface is adopted as projection direction. Accordingly, for an arbitrary point on master boundary $\mathbf{x}_m$ its corresponding projection point $\hat{\mathbf{x}}_s$ on the slave boundary is defined as:

$$\hat{\mathbf{x}}_s = \mathbf{x}_m + g_n \mathbf{n}_{bm} \tag{4.14}$$

By using Eq. (4.14) the normal gap on both sides is measured in the master normal direction, see Fig. 4.17. Such an approach is commonly termed the ray-tracing projection, and it can be characterized as an inverse projection relative to the closest point projection. Defining the gap function using the ray-tracing projection offers several appealing benefits [87]. Most importantly, it offers a unique projection, meaning that classical problems related to the closest point projection are avoided, see Fig. 4.14 a). Moreover, the known projection direction significantly reduces the complexity of the contact detection algorithm.
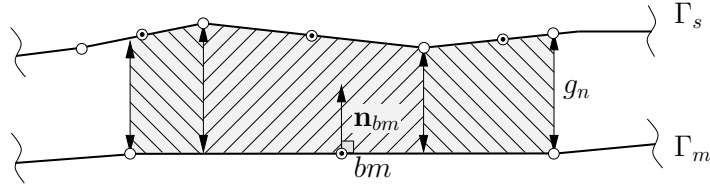
Figure 4.17. Master face and corresponding segments in 2D. Projection using master normal $\mathbf{n}_{bm}$

The discontinuity of the master surface normal field leads to an ambiguous projection at the points and alongside the mesh edges. To overcome this, one possibility is to smooth the surface normal field by using averaged normal vectors at discontinuity locations. This approach may be suitable for 2D; however, it is a much more complex task for 3D implementations. Another possibility is to permit ambiguous projections in such a way that each master segment has its individual projection direction. As shown in Fig. 4.18, this will lead to overlapping or undefined integration areas in the case of curved boundaries. Nonetheless, the introduced error is negligible since the relationship between penetration and segment length is very small [69]. Moreover, error diminishes with penetration reduction and mesh refinement.
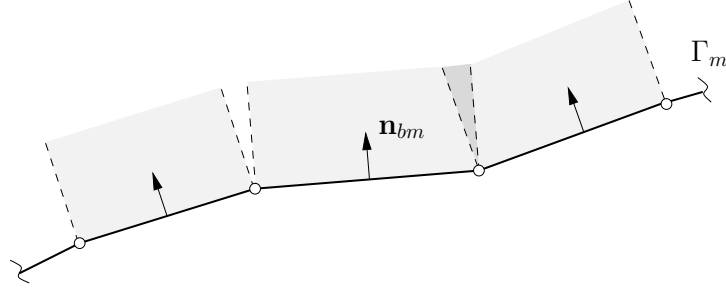
Figure 4.18. Undefined and overlapping projection areas

In the proposed segment-to-segment algorithm, the normal contact pressure acting on each master segment is integrated to obtain the integral value of face normal contact traction:

$$(\mathbf{t}_n)_{bm} = \frac{1}{||\mathbf{\Gamma}_m||} \int_{\Gamma_m} p_n \mathrm{d}\mathbf{\Gamma}_m. \qquad (4.15)$$

Using the linear relationship introduced by the penalty method in Eq. (4.1), Eq. (4.15) can be rewritten as:

$$(\mathbf{t}_n)_{bm} = \frac{1}{||\mathbf{\Gamma}_m||^2} \int_{\Gamma_m} \varepsilon_n g_n \mathrm{d}\mathbf{\Gamma}_m, \qquad (4.16)$$

thus the unknown distribution of the normal contact pressure is replaced with the distribution of normal gap g_n. In order to integrate the distribution of the normal gap, the slave faces are projected onto the master faces, decomposing each master face into a finite number of segments k with the C^1-continuous variation of gap function:

$$\int_{\Gamma_m} \varepsilon_n g_n \mathrm{d}\mathbf{\Gamma}_m = \varepsilon_n \sum_k \int_{\Gamma_k} g_n \mathrm{d}\mathbf{\Gamma}_k, \qquad (4.17)$$

where $\mathbf{\Gamma}_k$ is the area vector of segment k. The sum of the segments areas corresponds to the overall area of the master face:

$$\mathbf{\Gamma}_m = \sum_k \mathbf{\Gamma}_k. \qquad (4.18)$$

The master face and its corresponding segments are coplanar; therefore the segment area vector can be written as $\mathbf{\Gamma}_k = \Gamma_k \mathbf{n}_{bm}$. Since each segment k has a linear variation of the normal gap, the integral value of the normal gap in Eq. (4.17) is replaced with the sum of the segments averaged gap values $\bar{g}_{n,k}$:

$$\varepsilon_n \sum_k \int_{\Gamma_k} g_n \mathrm{d}\mathbf{\Gamma}_k = \varepsilon_n \mathbf{n}_{bm} \sum_k \bar{g}_{n,k} \Gamma_k. \qquad (4.19)$$

63

4. Contact treatment

The term $\bar{g}_{n,k}\Gamma_k$ represents the occluded volume of the corresponding master face (hatched areas in Fig. 4.16). The occluded volume is uniquely defined, resulting in a symmetric approach in the calculation of the normal contact traction. Accordingly, the normal contact traction on the slave side is simply obtained by summing the corresponding segments contributions which are calculated on the master side.

The tangential (frictional) component of the contact traction is calculated using the linear penalty method:

$$(\mathbf{t}_t^{tr})_{bm} = \varepsilon_t (\Delta \mathbf{g}_t)_{bm}, \tag{4.20}$$

where $\mathbf{t}_t^{tr}$ is the trial frictional traction used to determine the contact status:

$$\text{stick} \quad \mu |p_n| - ||\mathbf{t}_t^{tr}|| > 0 \quad \text{then} \quad \mathbf{t}_t = \mathbf{t}_t^{tr}, \tag{4.21}$$

$$\text{slip} \quad \mu |p_n| - ||\mathbf{t}_t^{tr}|| \leqslant 0 \quad \text{then} \quad \mathbf{t}_t = \mu |p_n| \frac{\mathbf{t}_t^{tr}}{||\mathbf{t}_t^{tr}||}. \tag{4.22}$$

Note that final frictional traction $\mathbf{t}_t$ which obeys Coulomb's friction law is calculated with the current value of the normal contact pressure. For the master side, the incremental tangential slip in Eq. (4.20) is calculated in accordance with Eqs. (2.36) and (2.37) as follows:

$$(\Delta \mathbf{g}_t)_{bm} = (\mathbf{I} - \mathbf{n}_{bm}\mathbf{n}_{bm}) \cdot (\Delta \mathbf{u}_{bm} - \Delta \hat{\mathbf{u}}_{bs}), \tag{4.23}$$

where $(\mathbf{I} - \mathbf{n}_{bm}\mathbf{n}_{bm})$ is the projection tensor which ensures that the frictional traction is in the tangent plane associated with the master face normal. $\Delta \hat{\mathbf{u}}_{bs}$ is the displacement increment of the slave surface at a location determined by the projection of the master face-centre. The projection of the master face-centre can be performed either by orthogonal projection or ray-tracing projection. Some preliminary tests showed that differences in results are negligible, thus ray-tracing may be preferable to avoid common problems of the orthogonal projection. Besides, it should be noted that both projections tend to the same solution with penetration reduction.

Following the same procedure, i.e. Eqs. (4.20)–(4.23), the frictional contact traction is calculated on both contact surfaces independently. This potentially results in a violation of the action-reaction principle; however, the results showed that the introduced error is insignificant.

Numerical implementation

This subsection outlines details involving the numerical implementation of the proposed contact-force calculation algorithm. The implementation is done as an extension of the GGI interpolation procedure, available within the `foam-extend` package. Primary components of the GGI code structure intended for projections, overlap tests and intersection construction have served as a starting point for the contact algorithm implementation.

The proposed calculation method of the normal component of the contact traction requires integration of the normal gap distance across each master face. By projecting the corresponding slave face pairs onto the master face, positions of discontinuities are located and the integration is performed over segments with C^1-continuous variation of the normal gap distance. This is illustrated in Fig. 4.19 and summarized for one contact pair as follows:

a) Construct an auxiliary plane $\mathbf{p}$ using the master face centre point $\mathbf{r}_{bm}$ and the corresponding unit normal vector $\mathbf{n}_{bm}$.

b) Project points of the slave and the master face onto auxiliary plane $\mathbf{p}$ using the plane normal. The master face points are projected to remove any possible warping. The projected faces (polygons) are denoted as $\overline{m}$ and $\overline{s}$. At this stage, the overlap test between the projected polygons is performed to check for the existence of intersection. If polygon $\overline{s}$ does not overlap with master polygon $\overline{m}$, the rest of the procedure is skipped.

c) Clip the overlapping faces using a clipping algorithm and form the intersection polygon, i.e. segment $k = \overline{s} \cap \overline{m}$.

d) Calculate geometric centre point $\mathbf{x}_c$ and decompose the intersection polygon into i triangles.

e) Using the auxiliary plane normal project points of the intersection polygon $\mathbf{x}_{k,i}$ onto slave face s and calculate the normal gap distance $g_{n,i}$ of each point, including centre point $\mathbf{x}_c$. Finally, the segment contribution in Eq. (4.19) is calculated by summing up the contribution of each triangle:

$$\bar{g}_{n,k}\Gamma_k = \sum_i \frac{1}{3}\left(g_{n,i} + g_{n,i+1} + g_{n,c}\right)\Gamma_{k,i}, \tag{4.24}$$

where $\Gamma_{k,i}$ is the surface area of corresponding triangle i.

4. Contact treatment
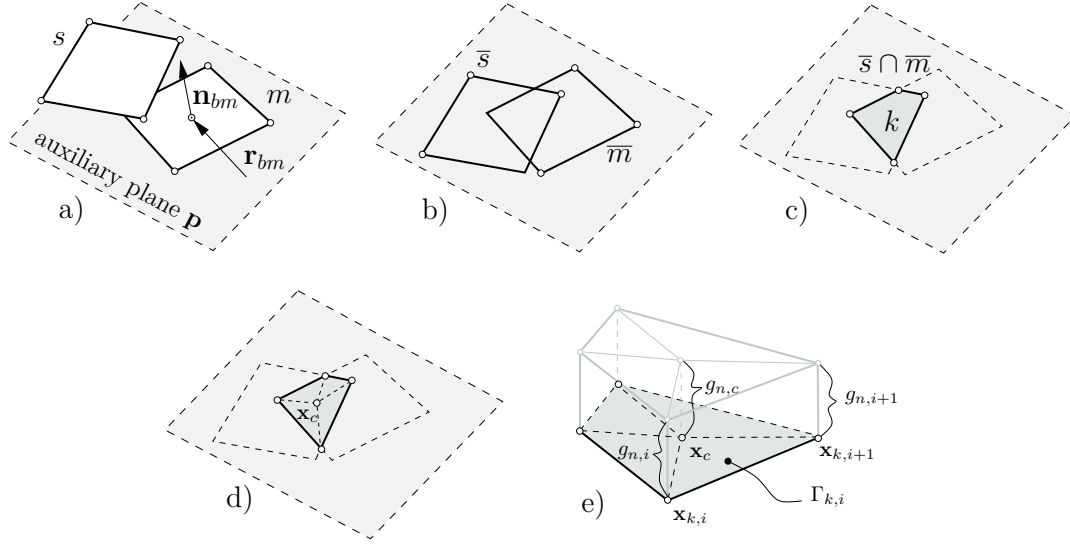
Figure 4.19. Main steps of integration algorithm: a) construction of auxiliary plane, b) projection onto auxiliary plane, c) construction of intersection polygon, d) central decomposition and e) computing segment contribution

In the case of partial contact, the distribution of the normal gap over the segment is discontinuous. To take this into account, steps d) – e) are modified to consider the region with a non-positive normal gap (integrand) distribution (the steps refer to Fig. 4.20):

c*) Use the normal gap values at points of polygon k to loop over its edges and form polygon $k^{'}$ which has non-positive normal gap distance at all points and edges.

d) Calculate geometric centre point $\mathbf{x}_{c^{'}}$ of polygon $k^{'}$, and decompose it into i triangles.

e) The segment contribution in Eq. (4.19) is calculated summing up the contribution of each triangle i of which polygon $k^{'}$ is composed:

$$\bar{g}_{n,k}\Gamma_k = \sum_i \frac{1}{3}\left(g_{n,i} + g_{n,i+1} + g_{n,c^{'}}\right)\Gamma_{k^{'},i}. \tag{4.25}$$
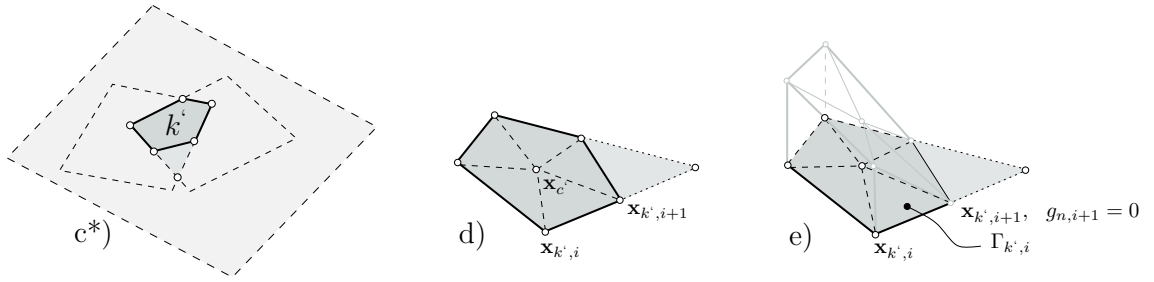
Figure 4.20. Calculation of segment contribution in the case of partial contact. Darker area represents polygonal region in contact

The in-plane overlap tests between polygons m and s were performed using the SAT algorithm. The Boolean Gilbert-Johnson-Keerthi (GJK) algorithm [129] was also implemented and tested, but as expected, the results showed that the SAT algorithm provided higher efficiency for triangular and rectangular faces. Since the efficiency of the SAT algorithm depends on the number of axes tested, the GJK algorithm can ensure higher efficiency for some specific cases (a large number of falsely reported contact pairs / polygonal meshes). Nonetheless, the efficiency of both algorithms is similar and any potential efficiency gain on this step is minor.

After the projection stage, the intersection construction or the so-called polygon clipping is performed to obtain the corresponding segment geometry. Because of the finite volume discretisation, the overlapping polygons are convex, which simplifies the required complexity of the polygon clipping algorithm. Accordingly, polygon clipping is performed using the Sutherland-Hodgman algorithm [8], which is fast, robust and can operate on any convex n-sided polygon. As a result, an ordered list of intersection polygon points is generated. Figure 4.21 illustrate the underlying strategy of the Sutherland-Hodgman algorithm — for further details see, for example, [119, 121]. Although the Sutherland-Hodgman algorithm is the optimal choice for the application presented here, it is possible to use other algorithms, for example the Weiler-Atherton algorithm [130], the Vatti algorithm [131] or the Greiner-Hormann [132] polygon clipping algorithm.

The property of the convex polygons is that their intersection is also convex, which is important for ensuring the existence of a valid (interior) centre point for the centre-based triangulation performed in step d). Alternatively, decomposition can be performed using the Delaunay or fan triangulation; however, it is expected that the centre-based triangulation will provide the best robustness. Here, it is important to notice that the overall procedure relies on the fact that faces s and m will remain convex after projection, which is true for the small deviation between

their normals.

It should be also noticed that the first three steps a)–c) are in fact the main steps of the GGI interpolation procedure. Accordingly, the calculation of the intersection polygon area allows the construction of the GGI area-weighted interpolation. This is an important feature since it allows the interpolation of other physical quantities across the contact interface (for example heat and electric flux in thermo-electro-mechanical contact).
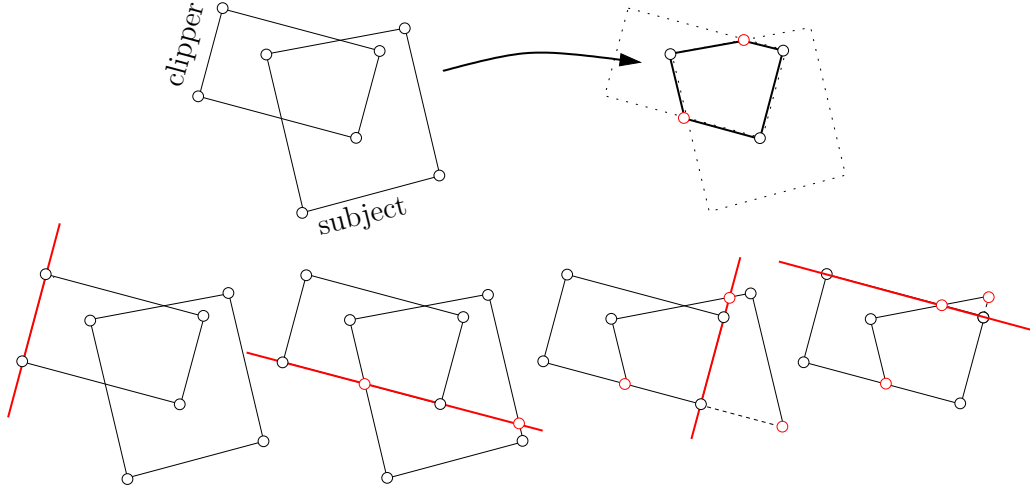
Figure 4.21. Illustration of Sutherland-Hodgman clipping algorithm [8]. Edges of clipping polygon (clipper) are used for calculation of possible intersection points and for rejection of points lying outside clipper

The calculation of the tangential slip increment using Eq. (4.23) requires the displacement increment at the location where the projection of the corresponding face centre is established. After finding a face with a valid (interior) point projection, the face is decomposed into triangular facets using the geometric centre. Next, for a triangular facet containing projection point $\hat{\mathbf{x}}$, the normalised barycentric (areal) coordinates are calculated and the value at the projection point is obtained using the barycentric interpolation:

$$(\Delta \mathbf{u})_{\hat{\mathbf{x}}} = \sum_{i}^{3} \Delta \mathbf{u}_i \zeta_i \tag{4.26}$$

where ζ_i are the normalised barycentric coordinates:

$$\zeta_1 + \zeta_2 + \zeta_3 = 1 \tag{4.27}$$

The described procedure is illustrated in Fig. 4.22. Note that the value of the

displacement increment at surface points $\Delta\mathbf{u}_i$ is already obtained with the least-squares method during the boundary update (value at the central point is calculated by averaging the face points values).
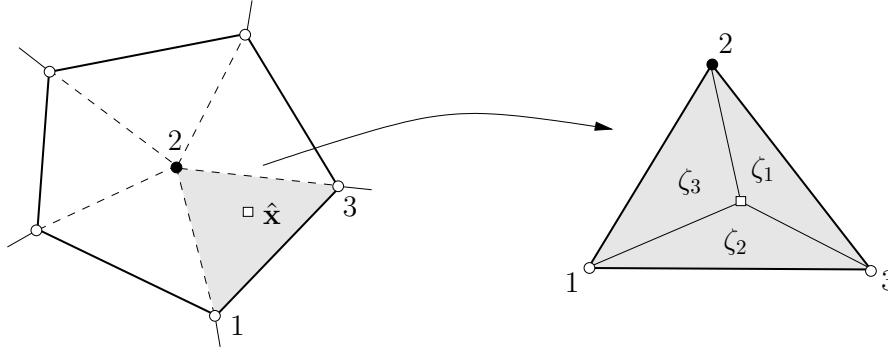
Figure 4.22. Central decomposition and interpolation using normalised barycentric coordinates

4.4.1. Comparison With the Pointwise Contact Algorithm

In this section, the main differences between the pointwise and the segment-to-segment contact algorithm are outlined by solving simple contact examples. In diagrams, the segment-to-segment contact algorithm is denoted with SCA and the pointwise algorithm with PCA.

In the first example, the contact between two concentric hollow cylinders is considered. A quarter of the geometry is modelled, and both cylinders have the same material properties. The inner cylinder is pressurised with pressure p, and the example is solved in one load increment. At the contact surface, a constant pressure distribution is expected. The inner cylinder is discretised in the circumferential direction with 40 CVs, while the outer cylinder is discredited with 30 CVs, see Fig. 4.23 a). Due to dissimilar discretisation in the circumferential direction, the resulting pressure field is not constant, and it exhibits oscillations in the circumferential direction. Accordingly, the goal of this example is to show which type of contact algorithm show less oscillatory behaviour.

Figure 4.24 shows the distribution of the resulting radial stresses σ_{rr} at the contact interface for different choices of the master surface. The average value of the radial stresses is for all cases in good agreement with the analytical solution [133]; however, resulting stress distribution exhibits visible oscillations. Although these oscillations can be removed by conformal contact discretisation or suppressed by finer discretisation, their existence is inevitable in cases with the non-conformal

piecewise linear discretisation of the curved contact boundaries. Accordingly, for practical applications, ensuring less oscillatory behaviour of the contact stresses is of great importance. One can see that the pointwise contact algorithm exhibits considerably larger oscillations of the contact pressure, especially when a coarser surface is chosen as a slave surface. In contrast to that the segment-to-segment contact algorithm exhibits the same magnitude of oscillations irrespective of the choice of the master surface. Moreover, the magnitude of oscillations is smaller and the same in all cases.
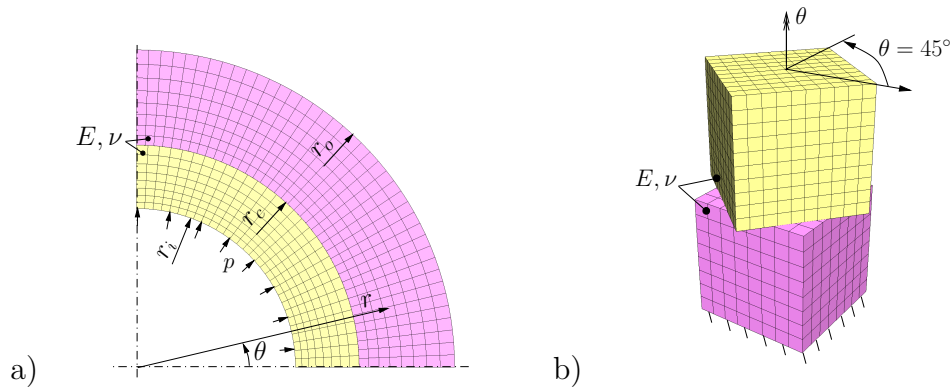
Figure 4.23. Contact between two concentric hollow cylinders ($r_o/r_i = 2, r_c = 1.4r_i$) a) and twisting contact between cubic blocks at $\theta = 45°$ b). Computational meshes
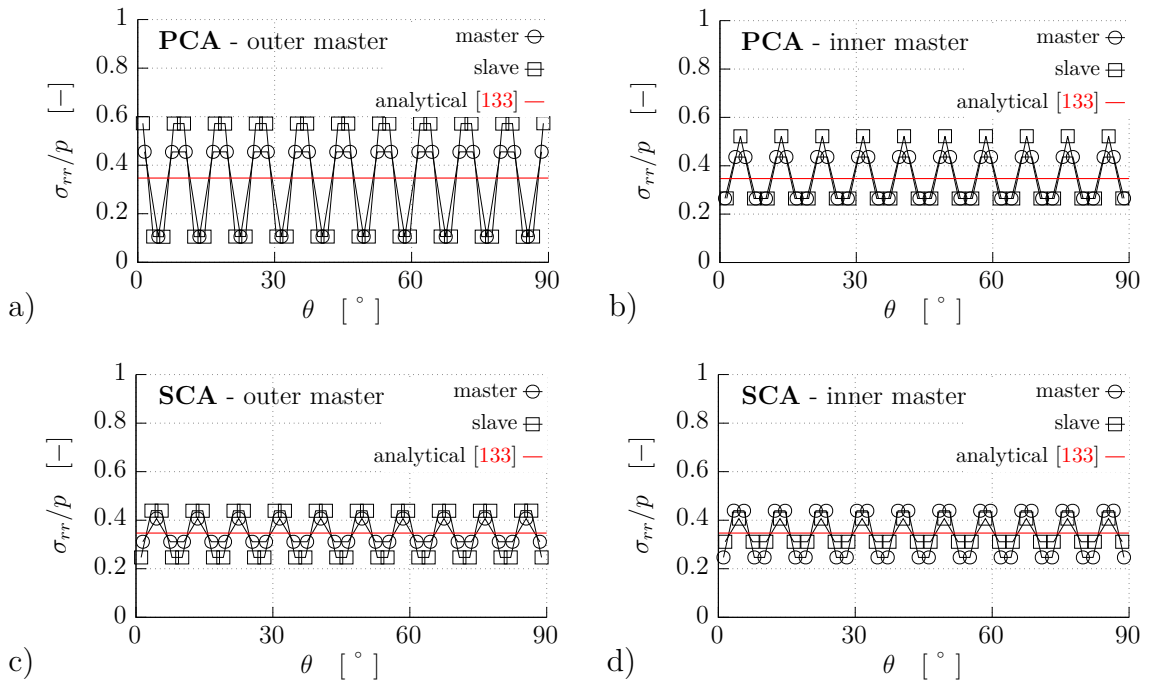
Figure 4.24. Contact pressure oscillations at contact interface

In the second example, gradual loss of contact is analysed by considering the contact between two unit blocks, see Fig 4.23 b). Both blocks have the same geometrical and material properties. The upper block is discretised with 1,000 CVs and the lower with 512 CVs. The bottom surface of the lower block is fixed, while the top surface of the upper block has prescribed vertical displacement and rotation. Prescribed displacement is held constant during the subsequently applied rotation of 90°.

Figure 4.25 shows the evolution of the vertical reaction force presented as normalised reaction pressure at the lower block bottom surface. While the segment-to-segment contact algorithm produces the same results irrespective of the choice of the master surface, the pointwise contact algorithm shows a significant difference between the results. The cause for this lies in impossibility to describe the gradual change in the contacted area. More precisely, the pointwise contact algorithm cannot correctly evaluate contact stress for faces in partial contact, leading to over-estimation or under-estimation of the contact stresses.
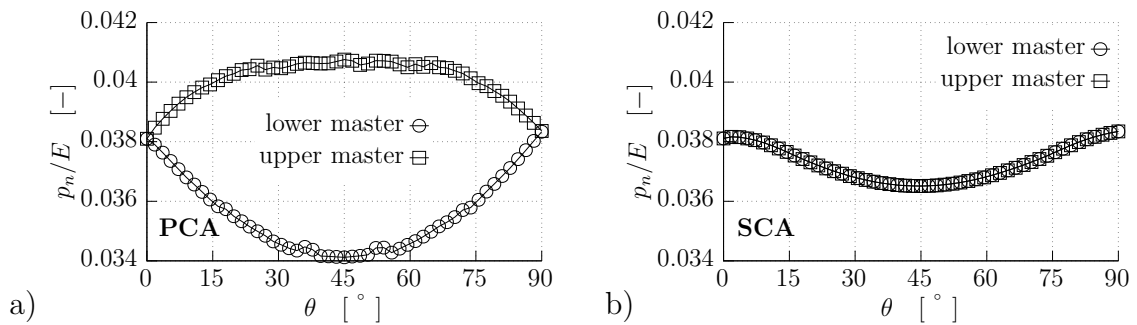
Figure 4.25. Twisting contact between cubic blocks: evolution of vertical reaction force

4.5. Implicit Neumann-Neumann Coupling Procedure

For many problems in which contact is the dominant phenomenon that governs deformation, the efficiency of the segregated solution procedure can be significantly impacted due to the slow convergence rate of the explicit contact coupling. In addition, the calculation expenses of the contact algorithm are relatively high, leading to a significant calculation overhead of each outer iteration.

The efficiency of the explicit contact coupling depends on two user-defined parameters: the penalty parameter and the under-relaxation factor. Hence, their value must be carefully set to obtain an acceptable level of accuracy and to ensure an acceptable convergence rate. The acceptable range value of the under-relaxation factor is inversely proportional to the specified penalty parameter, and tuning these factors can be a troublesome task, especially for an inexperienced user.

With the implicit implementation of the contact coupling, it is expected that the calculated contact force can be prescribed at the contact boundary without under-relaxation. Consequently, an optimal convergence rate can be achieved for any user-defined penalty factor.

While deriving the implicit coupling procedure, the following must be kept in mind:

- Contact traction is a function of the boundary displacement of both contacted surfaces.

- Computational points are located at the cell-centres; boundary displacement is not directly available.

- Only the converged solutions satisfy the momentum balance; any intermediate solution has unresolved inter-equation coupling.

- Contact conditions are described with non-differentiable functions and the resulting contact area is a priori unknown.

To successfully derive the implicit coupling procedure all of these considerations must be taken into account. In addition to the listed considerations, the assumption that implicit coupling requires an implicit contact detection algorithm is valid. Nevertheless, derivation of the implicit coupling which relies on a simpler explicit contact detection has been shown to be possible.

The proposed procedure, presented in the following sections, relies on the linearisation of the contact traction and derivation of the implicit equation for the boundary displacement, i.e. boundary gradient. Firstly, the simplest contact problems are considered and the complexity gradually increases.

4.5.1. 1D Contact Problems

In order to understand the principle of the herein presented implicit coupling procedure, the best way is to start with as simple example as possible. Let us consider an axial bar with constant axial stiffness AE and without axial load $q(x) = 0$, as shown in Fig. 4.26. For such a case, the differential equation in terms of displacement $\mathbf{u}$ has the following form:

$$\frac{d^2\mathbf{u}}{dx^2} = 0. \tag{4.28}$$

Following the basic finite volume discretisation procedures, Eq. 4.28 is discretised as follows:

$$\sum_f \mathbf{n}_f \cdot \left(\frac{d\mathbf{u}}{dx}\right)_f = 0. \tag{4.29}$$

Using Eq. 4.29, the system of equations for the discretised bar shown in Fig. 4.26 is:

$$\begin{bmatrix} -9 & 3 & 0 \\ 3 & -6 & 3 \\ 0 & 3 & -3 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -u_{0,b}/\delta \\ 0 \\ -F_c \end{bmatrix}, \tag{4.30}$$

where $u_{0,b}$ is the prescribed boundary displacement of the CV-0, F_c is the contact force and $\delta = \Delta x/2$ is distance between boundary face centre and cell centre. Since this is 1D example, in the case of contact, the contact area is known. However, even for this simple example the procedure which will allow contact detection and contact treatment must be defined. The straightforward way to go is to check and update the contact boundary after each iteration of the solution procedure. Accordingly, at the first iteration, the bar is not in contact and contact force F_c is set to zero, therefore the obtained solution is equal in every cell and it is equal to prescribed boundary displacement $u_{0,b}$. However, at the second iteration, in the case when prescribed displacement $u_{0,b}$ on the left side of the bar is larger than the initial gap g, contact will occur and the bar will undergo axial compression.

4. Contact treatment
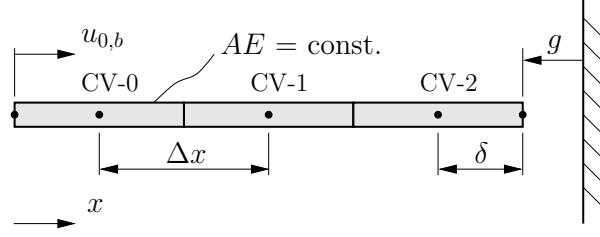
Figure 4.26. 1D bar in contact with rigid plane. Overall bar length and axial stiffness AE have unit value

Although the contact force directly depends on the gap value, the gap value is mathematically described with a non-linear function, thus deriving a linear equation for the contact force is not possible. Introducing the assumption that the boundary will remain in contact for the current iteration, the explicit gap value can be used to derive a linear equation for the contact force. That means that the contact force is a direct function of the boundary displacement. With that in mind, for this simple example the following equation can be derived:

$$F_c = \underbrace{\varepsilon g^*}_{\text{explicit part}} - \underbrace{\varepsilon \left(u_{2,b} - u_{2,b}^* \right)}_{\text{implicit correction}}, \qquad g^* = \begin{cases} g^* \text{ if } g^* \leqslant 0 \\ 0 \text{ if } g > 0 \end{cases}. \qquad (4.31)$$

where ε is the penalty parameter, g^* is the gap value from the previous iteration and $u_{2,b}$ is the CV-2 boundary displacement from the previous iteration. The first term in Eq. (4.31) represents the contact force calculated using the achieved penetration, whereas the second term represents the implicit correction, which tends to zero during the iterative procedure. Actually, Eq. (4.31) matches the equation from the Picard method, commonly used in FVM [134]:

$$S \approx S^* + \left(\frac{\partial S}{\partial \phi} \right)^* (\phi - \phi^*), \qquad (4.32)$$

where $S = S(\phi)$. To account for the iterations where there is no contact (such as the first iteration for the considered example), Eq. (4.31) is modified as follows:

$$F_c = \varepsilon g^* - \beta \varepsilon \left(u_{2,b} - u_{2,b}^* \right), \qquad \beta = \begin{cases} 1 \text{ if } g < 0 \\ 0 \text{ if } g = 0 \end{cases}. \qquad (4.33)$$

Since the gap value can take only negative values, in the case that there is no

74

contact, Eq. (4.33) is equal to zero which means that it can describe the stress-free boundary.

At this point, it is clear that Eq. (4.33) cannot be directly used without an equation for boundary displacement $u_{2,b}$. To resolve this problem the first logical attempt would be to assume that the boundary displacement is by value close to the displacement at the cell centre. By introducing such an assumption, Eq. (4.33) can be rewritten as:

$$F_c = \varepsilon g^* - \beta \varepsilon \left(u_{2,P} - u_{2,P}^* \right).$$ (4.34)

Introducing Eq. (4.34) into Eq. (4.30) we obtain the following system of equation:

$$
\begin{bmatrix}
-9 & 3 & 0 \\
3 & -6 & 3 \\
0 & 3 & -3 - \beta\varepsilon
\end{bmatrix}
\begin{bmatrix}
u_0 \\
u_1 \\
u_2
\end{bmatrix}
=
\begin{bmatrix}
-u_{0,b}/\delta \\
0 \\
-\varepsilon g^* - \beta\varepsilon u_{2,P}^*
\end{bmatrix}.
$$ (4.35)

Table 4.1 depicts the convergence history for two different scenarios. The first scenario is the case in which variable β is updated in each iteration. In that case, the system of equations cannot converge. More precisely, the bar exhibits oscillatory behaviour between active and inactive contact status. Although convergence is achieved in the second scenario, it is assumed that the bar is always in contact ($\beta = 1$) and that the gap value can take a positive value.

Table 4.1. 1D bar example: convergence history in the case that contact force in Eq. (4.33) is calculated with ($u_{0,b} = 0.1$, $\varepsilon = 1e5$): a) $\beta \neq$ const. , b) $\beta = 1$. Initial distance between bar and rigid surface is set to zero

a)

iteration	u_0	u_1	u_2	g^*
1	0.100	0.100	0.100	0
2	0.0800	0.0400	1.2e-6	-0.100
3	0.100	0.100	0.100	0
4	0.0800	0.0400	1.2e-6	-0.100
5	0.100	0.100	0.100	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

b)

iteration	u_0	u_1	u_2	g^*
1	0.0800	0.0400	1.2e-6	0
2	0.0840	0.0520	0.0200	0.01999
3	0.0832	0.0496	0.0160	-0.00400
4	0.0833	0.0500	0.0168	0.00072
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
15	0.0833	0.0500	0.0166	-0.0010
16	0.0833	0.0500	0.0166	-0.0009̇
17	0.0833	0.0500	0.0166	-0.0009̇

As shown, even for this simple example, substituting the boundary displacement with the cell-centre displacement does not provide convergence. Since the bar undergoes a linear distribution of displacement, the displacement gradient is constant along the bar, therefore boundary displacement in Eq. (4.33) can be extrapolated

from the interior using the implicit gradient:

$$u_{2,b} = u_2 + \frac{u_2 - u_1}{\Delta x}\,\delta. \tag{4.36}$$

Substituting Eq. (4.36) into Eq. (4.33), the following system of equations is obtained:

$$\begin{bmatrix} -9 & 3 & 0 \\ 3 & -6 & 3 \\ 0 & 3 + 0.5\beta\varepsilon & -3 - 1.5\beta\varepsilon \end{bmatrix}\begin{bmatrix} u_0 \\ u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -u_{0,b}/\delta \\ 0 \\ -\varepsilon g^* - \beta\varepsilon u_{2,b}^* \end{bmatrix}. \tag{4.37}$$

Finally, the system of equation converges within two iterations as depicted in Table 4.2. As expected, the implicit correction is zero at the third iteration.

Table 4.2. 1D bar example: convergence history ($u_{0,b} = 0.1$, $\varepsilon = 1e5$). Initial distance between bar and rigid surface is set to zero

iteration	u_0	u_1	u_2	g^*	$F_c = \varepsilon g^* - \beta\varepsilon\left(u_{2,b} - u_{2,b}^*\right),$
1	0.100	0.100	0.100	0	$= 0$
2	0.0833	0.0500	0.0166	-0.100	$= -\varepsilon \cdot 0.1 + 1 \cdot \varepsilon \cdot (9.999e\text{-}7 - 0.1) = -0.099$
3	0.0833	0.0500	0.0166	9.9e-7	$= -\varepsilon \cdot 9.9e\text{-}7 + \underbrace{1 \cdot \varepsilon \cdot (9.999e\text{-}7 - 9.999e\text{-}7)}_{=0} = -0.099$

Without introducing an additional set of equations for the boundary displacement, the boundary displacement can be derived in an implicit manner if the implicit equation for the displacement gradient at the cell centre is available. Generally, that equation is not available; Eq. (4.36) is valid only for this specific case.

To resolve this, a method that eliminates the need for an equation for the gradient is proposed:

$$F_c = AE\left(\frac{du}{dx}\right)_{2,b} = \varepsilon g^* - \beta\varepsilon\left(u_{2,b} - u_{2,b}^*\right), \tag{4.38}$$

$$\left(\frac{du}{dx}\right)_{2,b} = \frac{u_{2,b} - u_{2,P}}{\delta}. \tag{4.39}$$

Combining Eq. (4.38) and Eq. (4.39), the equation for boundary displacement is obtained:

$$u_{2,b} = \frac{\delta\varepsilon}{1 + \delta\beta\varepsilon}(g^* + \beta u_{2,b}^*) + \frac{1}{1 + \delta\beta\varepsilon}u_{2,P}. \tag{4.40}$$

Finally, the boundary displacement is a function of the solution variable, i.e. displacement at the cell-centre. Please note that the term AE is omitted for clarity since the unit value is used for the examples given herein.

Since the contact boundary is treated as the Neumann boundary, Eq. (4.39) is used once again in conjunction with Eq. (4.40) to derive the implicit equation for the boundary gradient:

$$g_{2,b} = \frac{\varepsilon}{1 + \delta\beta\varepsilon}(g^* + \beta u_{2,b}^*) - \frac{\varepsilon\beta}{1 + \delta\beta\varepsilon}u_{2,P}.$$

(4.41)

Combining Eq. (4.41) with Eq. (4.30), the following system of equations is obtained:

$$\begin{bmatrix} -9 & 3 & 0 \\ 3 & -6 & 3 \\ 0 & 3 & -3 - \dfrac{\varepsilon\beta}{1 + \delta\beta\varepsilon} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -u_{0,b}/\delta \\ 0 \\ -\dfrac{\varepsilon}{1 + \delta\varepsilon}(g^* + \beta u_{2,b}^*) \end{bmatrix}.$$

(4.42)

By solving the presented system of equations the same results as the results depicted in Table 4.2 are obtained. To conclude, the proposed procedure can handle the evolution of contact status during the iterative procedure. Also, it does not include an additional set of equations to describe the boundary displacement. To extend the proposed method for the contact between two deformable bodies, an example in which two bars are coming into contact is considered, as shown in Fig. 4.27.
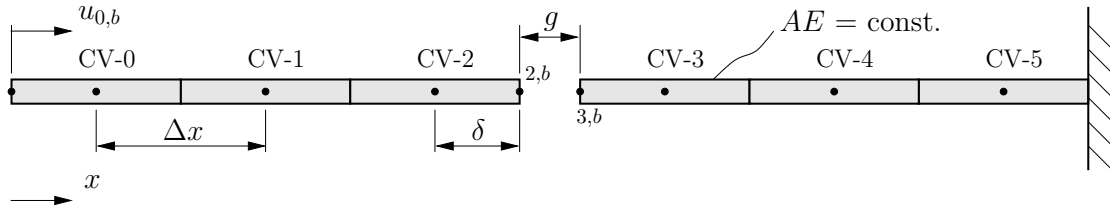
Figure 4.27. Two 1D bars coming into contact, both bars have unit axial stiffness AE and unit length

In the previous example, the contact force is a function solely of the one variable, however, in the case of the contact between two bodies it is a function of the boundary displacement of both bodies:

$$F_{2,c} = \varepsilon g^* - \beta\varepsilon(u_{2,b} - u_{2,b}^*) + \beta\varepsilon\left(u_{3,b} - u_{3,b}^*\right).$$

(4.43)

Again, Eq. (4.43) matches the equation from the Picard method:

$$S \approx S^* + \left(\frac{\partial S}{\partial \phi_a}\right)^* (\phi_a - \phi_a^*) + \left(\frac{\partial S}{\partial \phi_b}\right)^* (\phi_b - \phi_b^*),$$

(4.44)

4. Contact treatment

where S is a quantity that depends on two variables $S = S(\phi_a, \phi_b)$.

Because of the action-reaction principle, we can introduce equality $F_{2,c} = -F_{3,c}$ and combine it with Eq. (4.39) to obtain the boundary displacement in the following form:

$$u_{2,b} = u_{3,P} + u_{2,P} - u_{3,b}. \tag{4.45}$$

Substituting Eq. (4.45) to Eq. (4.43), the boundary displacement is obtained:

$$u_{2,b} = \frac{\delta\varepsilon}{1 + 2\delta\varepsilon\beta}(g^* + \beta u_{2,b}^* - \beta u_{3,b}^*) + \frac{\delta\varepsilon\beta}{1 + 2\delta\varepsilon\beta}u_{3,P} + \frac{1 + \delta\varepsilon\beta}{1 + 2\delta\varepsilon\beta}u_{2,P}. \tag{4.46}$$

Finally, the boundary gradient is obtained substituting Eq. (4.46) to Eq. (4.39):

$$g_{2,b} = \frac{\varepsilon}{1 + 2\delta\varepsilon\beta}(g^* + \beta u_{2,b}^* - \beta u_{3,b}^*) + \frac{\varepsilon\beta}{1 + 2\delta\varepsilon\beta}u_{3,P} - \frac{\varepsilon\beta}{1 + 2\delta\varepsilon\beta}u_{2,P}. \tag{4.47}$$

The boundary gradient for the right bar is obtained by following the same procedure. After discretisation, the resulting system of equations is:

$$
\begin{bmatrix}
-9 & 3 & 0 & 0 & 0 & 0 \\
3 & -6 & 3 & 0 & 0 & 0 \\
0 & 3 & -3 - c\beta & c\beta & 0 & 0 \\
0 & 0 & c\beta & -3 - c\beta & 3 & 0 \\
0 & 0 & 0 & 3 & -6 & 3 \\
0 & 0 & 0 & 0 & 3 & -9
\end{bmatrix}
\begin{bmatrix}
u_0 \\
u_1 \\
u_2 \\
u_3 \\
u_4 \\
u_5
\end{bmatrix}
=
\begin{bmatrix}
-u_{0,b}/\delta \\
0 \\
c(g^* + \beta u_{2,b}^* - \beta u_{3,b}^*) \\
c(g^* + \beta u_{2,b}^* - \beta u_{3,b}^*) \\
0 \\
0
\end{bmatrix}, \tag{4.48}
$$

where $c = \varepsilon(1 + 2\delta\beta\varepsilon)^{-1}$. The term c contributes to diagonal coefficients only for the cells adjacent to the contact boundary and it is a coupling term between two cells in contact. The resulting system of equations maintains a symmetric and diagonal dominant form produced by discretising the Laplace operator. The solution is obtained iteratively by treating the solution vector and term c in a deferred correction manner. The presented example is solved in two iterations, as depicted in Table 4.3. In the first iteration, the contact is detected, whereas in the second iteration a final solution is obtained and no further iterations are needed.

Table 4.3. Two 1D bars coming into contact: convergence history ($u_{0,b} = 0.1$, $\varepsilon = 1e5$). Initial distance between bars is set to zero

iteration	u_0	u_1	u_2	u_3	u_4	u_5	g^*
1	0.100	0.100	0.100	0	0	0	0
2	0.0916667	0.0750001	0.0583335	0.0416665	0.0249999	0.0083333	0.1
3	0.0916667	0.0750001	0.0583335	0.0416665	0.0249999	0.0083333	5e-7

In contrast, if the same example is solved using the explicit contact coupling, the required number of iterations can easily reach up to a few hundred. For demonstration, Table 4.4 depicts the convergence history in the case of under-relaxation $\alpha = 2.5 \cdot 10^{-6}$ for which 40 iterations are required to reach convergence.

Table 4.4. Two 1D bars coming into contact: convergence history in the case of explicit contact coupling

iteration	u_0	u_1	u_2	u_3	u_4	u_5	g^*
1	0.100	0.100	0.100	0	0	0	0
2	0.0958333	0.8750000	0.0791666	0.0208333	0.0125000	0.0004166	0.1
3	0.0937500	0.0812500	0.0687500	0.0312499	0.0187499	0.0006249	5e-2
4	0.0927083	0.0781250	0.0635417	0.0364582	0.0218749	0.0072916	2.5e-2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
20	0.0916667	0.0750002	0.0583336	0.0416664	0.0249998	0.0083332	8.8e-7
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
40	0.0916667	0.0750001	0.0583335	0.0416665	0.0249999	0.0083333	5e-7

4.5.2. 2D and 3D Contact Problems

The next step is to extend the proposed procedure to 2D and 3D frictionless contact problems. Firstly, the procedure for linear-elastic solids is derived and the extension to the nonlinear material model is considered later on. Regardless of the contact type, to obtain the gradient at the contact boundary in an implicit manner, the equation for boundary traction and boundary gradient are the starting equations, therefore they are listed below:

• Boundary traction in over-relaxed form:

$$\mathbf{t}_b = (2\mu_b + \lambda_b)\mathbf{n}_b \boldsymbol{\cdot} (\boldsymbol{\nabla}\mathbf{u})_b - (\mu_b + \lambda_b)\mathbf{n}_b \boldsymbol{\cdot} (\boldsymbol{\nabla}\mathbf{u})_b + \mu_b \mathbf{n}_b \boldsymbol{\cdot} (\boldsymbol{\nabla}\mathbf{u})_b^{\mathrm{T}} + \lambda_b \mathbf{n}_b \mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_b). \quad (4.49)$$

• Boundary gradient is the dot product of outward-pointing normal and gradient:

$$\mathbf{g}_b = \mathbf{n}_b \boldsymbol{\cdot} (\boldsymbol{\nabla}\mathbf{u})_b = \frac{\mathbf{u}_b - \mathbf{u}_P}{\delta_{nb}}, \quad (4.50)$$

where $\delta_{nb} = \mathbf{n}_b \cdot \mathbf{d}_b$. Please note that Eq. (4.50) assumes an orthogonal mesh at the contact boundary, i.e.vectors $\mathbf{d}_b$ and $\mathbf{d}_n$ are perfectly parallel, see Fig. 3.2. The extension that will consider boundary non-orthogonality will be introduced later. All equations are derived with total displacement $\mathbf{u}$ as a dependent variable.

Contact between deformable body and rigid surface

The contact between a deformable body and a rigid surface can be considered as a special case of contact between two bodies, however, for the sake of simplicity, we will first consider such a case.

Since we have only one deformable body, the contact boundary is denoted with subscript b, as shown in Fig. 4.28. It is a common practice in the FV literature to use the first and second Lamé parameters in the case of linear-elasticity; this notation is also used herein although it is somehow impractical since the coefficient of friction shares the same symbol as the second Lamé parameter. However, frictionless contact is assumed, therefore μ refers solely to the second Lamé parameter.
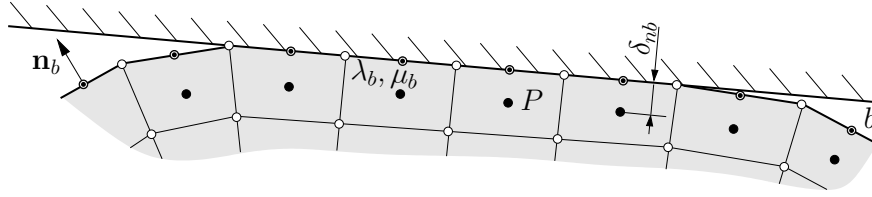
Figure 4.28. Contact between deformable body and rigid surface

Combining Eqs. (4.49) and (4.50) results in:

$$\frac{\mathbf{u}_b - \mathbf{u}_P}{\delta_{nb}} = \frac{\mathbf{t}_b - \mathbf{n}_b \cdot \left[\mu_b (\boldsymbol{\nabla}\mathbf{u})_b^{\mathrm{T}} - (\mu_b + \lambda_b)(\boldsymbol{\nabla}\mathbf{u})_b \right] - \mathbf{n}_b \cdot \lambda_b \mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_b)}{(2\mu_b + \lambda_b)}. \quad (4.51)$$

The linearised equation for boundary traction t_b is:

$$\mathbf{t}_b = \varepsilon_n \mathbf{n}_b^* g_n^* - \beta \varepsilon_n (\mathbf{n}_b^* \mathbf{n}_b^*) \cdot (\mathbf{u}_b - \mathbf{u}_b^*), \quad (4.52)$$

where $\mathbf{n}_b^*$ is the boundary normal vector calculated using the displacement field from the previous iteration. The term $(\mathbf{n}_b^* \mathbf{n}_b^*)$ is added to account for only the normal component of the boundary displacement, i.e. the tangential component does not contribute to the normal contact pressure. The term β is calculated as the ratio

between face area in contact and face total area. More precisely, it represents the relative area in contact which is bounded between zero and unit value.

The equation for the boundary displacement is obtained by substituting Eq. (4.52) into Eq. (4.51):

$$
\begin{aligned}
\mathbf{u}_b = {} & \frac{\varepsilon_n \delta_{nb}}{(2\mu_b + \lambda_b)} \left[\mathbf{I} + \frac{\varepsilon_n \delta_{nb}\beta}{(2\mu_b + \lambda_b)} \mathbf{n}_b^* \mathbf{n}_b^* \right]^{-1} \boldsymbol{\cdot} (\mathbf{n}_b^* \mathbf{n}_b^*) \boldsymbol{\cdot} (\beta \mathbf{u}_b^* + \mathbf{n}_b^* g_n^*) \\
& + \frac{\delta_{nb}}{(2\mu_b + \lambda_b)} \left[\mathbf{I} + \frac{\varepsilon_n \delta_{nb}\beta}{(2\mu_b + \lambda_b)} \mathbf{n}_b^* \mathbf{n}_b^* \right]^{-1} \boldsymbol{\cdot} \left[\mathbf{n}_b \boldsymbol{\cdot} \left[(\mu_b + \lambda_b)(\boldsymbol{\nabla}\mathbf{u})_b - \mu_b (\boldsymbol{\nabla}\mathbf{u})_b^{\mathrm{T}} \right] \right] \\
& - \frac{\delta_{nb}}{(2\mu_b + \lambda_b)} \left[\mathbf{I} + \frac{\varepsilon_n \delta_{nb}\beta}{(2\mu_b + \lambda_b)} \mathbf{n}_b^* \mathbf{n}_b^* \right]^{-1} \boldsymbol{\cdot} \left[\mathbf{n}_b \boldsymbol{\cdot} \lambda_b \mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_b) \right] \\
& + \left[\mathbf{I} + \frac{\varepsilon_n \delta_{nb}\beta}{(2\mu_b + \lambda_b)} \mathbf{n}_b^* \mathbf{n}_b^* \right]^{-1} \boldsymbol{\cdot} \mathbf{u}_P.
\end{aligned}
\tag{4.53}
$$

Finally, the boundary gradient is obtained combining Eq. (4.50) and Eq. (4.53):

$$
\mathbf{g}_b = \tag{4.54}
$$

$$
\frac{\varepsilon_n}{(2\mu_b + \lambda_b)} \left[\mathbf{I} + \frac{\varepsilon_n \delta_{nb}\beta}{(2\mu_b + \lambda_b)} \mathbf{n}_b^* \mathbf{n}_b^* \right]^{-1} \boldsymbol{\cdot} (\mathbf{n}_b^* \mathbf{n}_b^*) \boldsymbol{\cdot} (\beta \mathbf{u}_b^* + \mathbf{n}_b^* g_n^*) \tag{i}
$$

$$
+ \frac{1}{(2\mu_b + \lambda_b)} \left[\mathbf{I} + \frac{\varepsilon_n \delta_{nb}\beta}{(2\mu_b + \lambda_b)} \mathbf{n}_b^* \mathbf{n}_b^* \right]^{-1} \boldsymbol{\cdot} \left[\mathbf{n}_b \boldsymbol{\cdot} \left[(\mu_b + \lambda_b)(\boldsymbol{\nabla}\mathbf{u})_b - \mu_b (\boldsymbol{\nabla}\mathbf{u})_b^{\mathrm{T}} \right] \right] \tag{ii}
$$

$$
- \frac{1}{(2\mu_b + \lambda_b)} \left[\mathbf{I} + \frac{\varepsilon_n \delta_{nb}\beta}{(2\mu_b + \lambda_b)} \mathbf{n}_b^* \mathbf{n}_b^* \right]^{-1} \boldsymbol{\cdot} \left[\mathbf{n}_b \boldsymbol{\cdot} \lambda_b \mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_b) \right] \tag{iii}
$$

$$
+ \frac{1}{\delta_{nb}} \left\{ \left[\mathbf{I} + \frac{\varepsilon_n \delta_{nb}\beta}{(2\mu_b + \lambda_b)} \mathbf{n}_b^* \mathbf{n}_b^* \right]^{-1} \right\}_D \boldsymbol{\cdot} \mathbf{u}_P \tag{iv}
$$

$$
+ \frac{1}{\delta_{nb}} \left\{ \left[\mathbf{I} + \frac{\varepsilon_n \delta_{nb}\beta}{(2\mu_b + \lambda_b)} \mathbf{n}_b^* \mathbf{n}_b^* \right]^{-1} \right\}_{LU} \boldsymbol{\cdot} \mathbf{u}_P^* \tag{v}
$$

$$
- \frac{1}{\delta_{nb}} \mathbf{u}_P. \tag{vi}
$$

Terms (i), (ii) and (iii) contribute to the right-hand side vector of the resulting system of the algebraic equation. Note that the boundary gradient, its trace and transpose are calculated explicitly, superscript * is omitted for clarity. Terms (iv) and (v) represent the same term, however (iv) consists only of the tensor diagonal components, whereas (v) includes only the off-diagonal tensor components. The diagonal tensor components together with term (vi) are added to the diagonal coefficients of the left-hand side matrix. The off-diagonal components are multiplied with

4. Contact treatment

the solution from the previous iteration and subsequently added to the right-hand side vector. The explained decoupling into off-diagonal and diagonal components is a common strategy in preserving efficiency when the segregated solution procedure is used. One can see that Eq. (4.54) is similar to Eq. (4.41), derived for the simple bar contact. The difference is in lagged traction components (ii and iii) and term (v).

To demonstrate the performance of the above-proposed procedure, Hertzian contact problems described in Subsections 5.2.1 and 5.2.5 are considered. Figure 4.29 a) shows the normal pressure distribution at the cylinder contact surface, whereas Fig. 4.29 b) shows the half-sphere subsurface stresses along the z-axis. As shown, the same solution is obtained, i.e. accuracy is not affected by the implicit implementation.
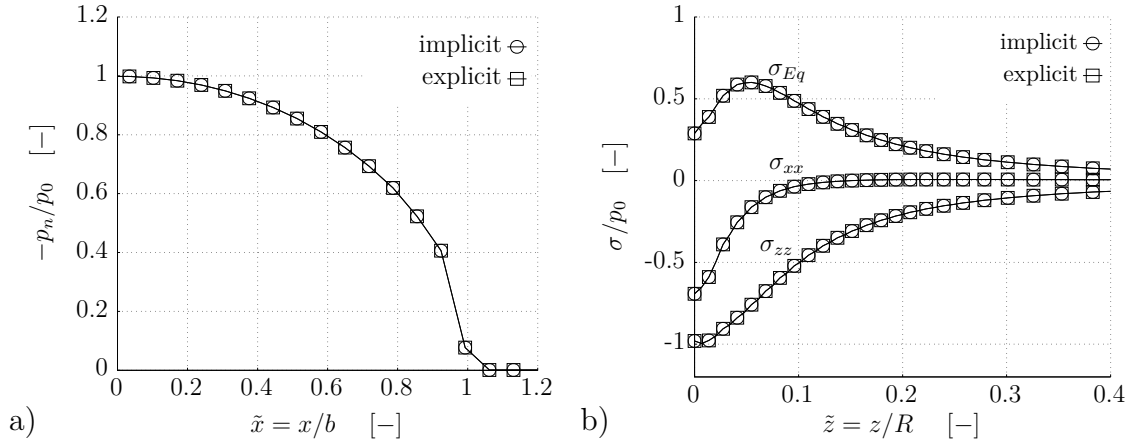
Figure 4.29. Comparison of stress distribution between explicit and implicit contact coupling: a) cylinder and b) sphere contact with rigid plane. Both examples are solved using the fine mesh

Table 4.5. Efficiency comparison between implicit and explicit coupling for different values of penalty parameter — contact between cylinder/sphere and rigid plane

Sphere — rigid plane

ε_n/E_s	Implicit n_{corr}	Explicit (optimal) n_{corr}	Avg. relative gap [%]
10	58	64	13
100	60	245	2
1,000	61	1,107	0.2

Cylinder — rigid plane

ε_n/E_s	Implicit n_{corr}	Explicit (optimal) n_{corr}	Avg. relative gap [%]
10	52	114	4
100	57	672	0.5
1,000	60	2,332	0.05

The results depicted in Table 4.5 in which a comparison between the number of outer iterations is made for different penalty factors are more interesting. Since the efficiency of the explicit coupling procedure depends on the prescribed under-relaxation factor, the depicted number of outer iterations is obtained using its optimal value, which is manually obtained in an iterative manner. One can see that the implicit coupling maintains the same number of iterations with a penetration reduction. On the other hand, in the case of explicit coupling, the number of iterations increases proportionally with a reduction in penetration. Although in the case of a lower value of the penalty parameter the explicit coupling can have similar efficiency as the implicit coupling, the tests show that this typically leads to an unsatisfactory level of accuracy. When a high level of accuracy is required, the implicit coupling shows much better convergence behaviour — the expected reduction in the number of outer iterations is at least in the range of $\sim$2-3.

Conformal discretisation at the contact interface

In this section, the contact between two deformable bodies with the assumption of conformal discretisation at the contact surface is considered. The slave and master notation is introduced for the bodies in contact to be distinguished — subscript b is replaced with the corresponding subscript for master or slave boundary. Combining Eqs. (4.49) and (4.50) for the for master side we get:

$$\frac{\mathbf{u}_{bm} - \mathbf{u}_{Pm}}{\delta_{nm}} = \frac{\mathbf{t}_{bm} - \mathbf{n}_{bm} \boldsymbol{\cdot} \left[\mu_{bm}(\boldsymbol{\nabla}\mathbf{u})_{bm}^{\mathrm{T}} - (\mu_{bm} + \lambda_{bm})(\boldsymbol{\nabla}\mathbf{u})_{bm}\right] - \mathbf{n}_{bm} \boldsymbol{\cdot} \lambda_{bm}\mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_{bm})}{(2\mu_{bm} + \lambda_{bm})}$$
(4.55)

The same is obtained for the slave side:

$$\frac{\mathbf{u}_{bs} - \mathbf{u}_{Ps}}{\delta_{ns}} = \frac{\mathbf{t}_{bs} - \mathbf{n}_{bs} \boldsymbol{\cdot} \left[\mu_{bs}(\boldsymbol{\nabla}\mathbf{u})_{bs}^{\mathrm{T}} - (\mu_{bs} + \lambda_{bs})(\boldsymbol{\nabla}\mathbf{u})_{bs}\right] - \mathbf{n}_{bs} \boldsymbol{\cdot} \lambda_{bs}\mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_{bs})}{(2\mu_{bs} + \lambda_{bs})}$$
(4.56)

The next step is to express the boundary displacement as a function of cell centre displacements and the boundary displacement of the adjacent cell. To accomplish this we will take the equality of contact traction into account:

$$\mathbf{t}_{bm} = -\mathbf{t}_{bs}.$$
(4.57)

4. Contact treatment

Combining Eqs. (4.55), (4.56) and (4.57) results in the following expressions for the master and slave boundary displacement:

$$
\begin{aligned}
\mathbf{u}_{bm} =\ & \frac{\delta_{nm}(2\mu_{bs}+\lambda_{bs})}{\delta_{ns}(2\mu_{bm}+\lambda_{bm})}\left(\mathbf{u}_{Ps}-\mathbf{u}_{bs}\right)+\mathbf{u}_{Pm}\\
& +\frac{\delta_{nm}}{(2\mu_{bm}+\lambda_{bm})}\left\{\mathbf{n}_{bm}\boldsymbol{\cdot}\left[(\mu_{bm}+\lambda_{bm})(\boldsymbol{\nabla}\mathbf{u})_{bm}-\mu_{bm}(\boldsymbol{\nabla}\mathbf{u})_{bm}^{\mathrm{T}}\right]-\mathbf{n}_{bm}\boldsymbol{\cdot}\lambda_{bm}\mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_{bm})\right\}\\
& +\frac{\delta_{nm}}{(2\mu_{bm}+\lambda_{bm})}\left\{\mathbf{n}_{bs}\boldsymbol{\cdot}\left[(\mu_{bs}+\lambda_{bs})(\boldsymbol{\nabla}\mathbf{u})_{bs}-\mu_{bs}(\boldsymbol{\nabla}\mathbf{u})_{bs}^{\mathrm{T}}\right]-\mathbf{n}_{bs}\boldsymbol{\cdot}\lambda_{bs}\mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_{bs})\right\},
\end{aligned}\tag{4.58}
$$

$$
\begin{aligned}
\mathbf{u}_{bs} =\ & \frac{\delta_{ns}(2\mu_{bm}+\lambda_{bm})}{\delta_{nm}(2\mu_{bs}+\lambda_{bs})}\left(\mathbf{u}_{Pm}-\mathbf{u}_{bm}\right)+\mathbf{u}_{Ps}\\
& +\frac{\delta_{ns}}{(2\mu_{bs}+\lambda_{bs})}\left\{\mathbf{n}_{bm}\boldsymbol{\cdot}\left[(\mu_{bm}+\lambda_{bm})(\boldsymbol{\nabla}\mathbf{u})_{bm}-\mu_{bm}(\boldsymbol{\nabla}\mathbf{u})_{bm}^{\mathrm{T}}\right]-\mathbf{n}_{bm}\boldsymbol{\cdot}\lambda_{bm}\mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_{bm})\right\}\\
& +\frac{\delta_{ns}}{(2\mu_{bs}+\lambda_{bs})}\left\{\mathbf{n}_{bs}\boldsymbol{\cdot}\left[(\mu_{bs}+\lambda_{bs})(\boldsymbol{\nabla}\mathbf{u})_{bs}-\mu_{bs}(\boldsymbol{\nabla}\mathbf{u})_{bs}^{\mathrm{T}}\right]-\mathbf{n}_{bs}\boldsymbol{\cdot}\lambda_{bs}\mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_{bs})\right\}.
\end{aligned}\tag{4.59}
$$

Per Picard linearization, the linearised normal contact traction for both sides is:

$$
\mathbf{t}_{bm} = (\mathbf{n}_{bm}^{*}\mathbf{n}_{bm}^{*})\boldsymbol{\cdot}\left[\varepsilon_n\mathbf{n}_{bm}^{*}(g_n)_m^{*}-\varepsilon_n\beta_m(\mathbf{u}_{bm}-\mathbf{u}_{bm}^{*})+\varepsilon_n\beta_m(\mathbf{u}_{bs}-\mathbf{u}_{bs}^{*})\right],\tag{4.60}
$$

$$
\mathbf{t}_{bs} = (\mathbf{n}_{bs}^{*}\mathbf{n}_{bs}^{*})\boldsymbol{\cdot}\left[\varepsilon_n\mathbf{n}_{bs}^{*}(g_n)_s^{*}-\varepsilon_n\beta_s(\mathbf{u}_{bs}-\mathbf{u}_{bs}^{*})+\varepsilon_n\beta_s(\mathbf{u}_{bm}-\mathbf{u}_{bm}^{*})\right],\tag{4.61}
$$

where $\mathbf{n}_{bm}^{*}$ and $\mathbf{n}_{bs}^{*}$ are the boundary normal vectors, calculated on both sides independently, using the displacement solution from the previous iteration. Due to the conformal discretisation, the relative area in contact is the same for both sides $\beta_m = \beta_s$. Combining Eqs. (4.55), (4.59) and (4.60) the master boundary displacement as a function of the cell-centre displacement is obtained:

$$
\begin{aligned}
\mathbf{u}_{bm} =\ & \left[\mathbf{I}+\varepsilon_n\beta_m\left(\frac{\delta_{nm}}{(2\mu_{bm}+\lambda_{bm})}+\frac{\delta_{ns}}{(2\mu_{bs}+\lambda_{bs})}\right)(\mathbf{n}_{bm}^{*}\mathbf{n}_{bm}^{*})\right]^{-1}\\
& \boldsymbol{\cdot}\left\{\ \varepsilon_n\frac{\delta_{nm}}{(2\mu_{bm}+\lambda_{bm})}(\mathbf{n}_{bm}^{*}\mathbf{n}_{bm}^{*})\boldsymbol{\cdot}[\mathbf{n}_{bm}^{*}(g_n)_m^{*}+\beta_m\mathbf{u}_{bm}^{*}-\beta_m\mathbf{u}_{bs}^{*}]\right.\\
& +\varepsilon_n\beta_m\frac{\delta_{nm}\delta_{ns}}{(2\mu_{bm}+\lambda_{bm})(2\mu_{bs}+\lambda_{bs})}(\mathbf{n}_{bm}^{*}\mathbf{n}_{bm}^{*})\\
& \qquad\boldsymbol{\cdot}\left[\mathbf{n}_{bm}\boldsymbol{\cdot}\left[(\mu_{bm}+\lambda_{bm})(\boldsymbol{\nabla}\mathbf{u})_{bm}-\mu_{bm}(\boldsymbol{\nabla}\mathbf{u})_{bm}^{\mathrm{T}}\right]-\mathbf{n}_{bm}\boldsymbol{\cdot}\lambda_{bm}\mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_{bm})\right.\\
& \qquad\left.+\mathbf{n}_{bs}\boldsymbol{\cdot}\left[(\mu_{bs}+\lambda_{bs})(\boldsymbol{\nabla}\mathbf{u})_{bs}-\mu_{bs}(\boldsymbol{\nabla}\mathbf{u})_{bs}^{\mathrm{T}}\right]-\mathbf{n}_{bs}\boldsymbol{\cdot}\lambda_{bs}\mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_{bs})\right]\\
& +\frac{\delta_{nm}}{(2\mu_{bm}+\lambda_{bm})}\\
& \qquad\boldsymbol{\cdot}\left[\mathbf{n}_{bm}\boldsymbol{\cdot}\left[(\mu_{bm}+\lambda_{bm})(\boldsymbol{\nabla}\mathbf{u})_{bm}-\mu_{bm}(\boldsymbol{\nabla}\mathbf{u})_{bm}^{\mathrm{T}}\right]-\mathbf{n}_{bm}\boldsymbol{\cdot}\lambda_{bm}\mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_{bm})\right]\\
& \left.+\left[\mathbf{I}+\varepsilon_n\beta_m\frac{\delta_{ns}}{(2\mu_{bs}+\lambda_{bs})}(\mathbf{n}_{bm}^{*}\mathbf{n}_{bm}^{*})\right]\boldsymbol{\cdot}\mathbf{u}_{Pm}+\varepsilon_n\beta_m\frac{\delta_{nm}}{(2\mu_{bm}+\lambda_{bm})}(\mathbf{n}_{bm}^{*}\mathbf{n}_{bm}^{*})\boldsymbol{\cdot}\mathbf{u}_{Ps}\right\}.
\end{aligned}\tag{4.62}
$$

Finally, the expression for the boundary gradient on the master side is obtained combining Eq. (4.62) and (4.50):

$$
\mathbf{g}_{bm} = \left[\mathbf{I} + \varepsilon_n \beta_m \left(\frac{\delta_{nm}}{(2\mu_{bm} + \lambda_{bm})} + \frac{\delta_{ns}}{(2\mu_{bs} + \lambda_{bs})} \right) (\mathbf{n}_{bm}^* \mathbf{n}_{bm}^*) \right]^{-1}
$$

$$
\cdot \left\{ \varepsilon_n \frac{1}{(2\mu_{bm} + \lambda_{bm})} (\mathbf{n}_{bm}^* \mathbf{n}_{bm}^*) \cdot [\mathbf{n}_{bm}^* (g_n)_m^* + \beta_m \mathbf{u}_{bm}^* - \beta_m \mathbf{u}_{bs}^*] \right.
$$

$$
+ \varepsilon_n \beta_m \frac{\delta_{ns}}{(2\mu_{bm} + \lambda_{bm})(2\mu_{bs} + \lambda_{bs})} (\mathbf{n}_{bm}^* \mathbf{n}_{bm}^*)
$$

$$
\cdot \left[\mathbf{n}_{bm} \cdot \left[(\mu_{bm} + \lambda_{bm})(\boldsymbol{\nabla}\mathbf{u})_{bm} - \mu_{bm}(\boldsymbol{\nabla}\mathbf{u})_{bm}^{\mathrm{T}} \right] - \mathbf{n}_{bm} \cdot \lambda_{bm} \mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_{bm}) \right.
$$

$$
\left. + \mathbf{n}_{bs} \cdot \left[(\mu_{bs} + \lambda_{bs})(\boldsymbol{\nabla}\mathbf{u})_{bs} - \mu_{bs}(\boldsymbol{\nabla}\mathbf{u})_{bs}^{\mathrm{T}} \right] - \mathbf{n}_{bs} \cdot \lambda_{bs} \mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_{bs}) \right]
$$

$$
\left. + \frac{1}{(2\mu_{bm} + \lambda_{bm})} \cdot \left[\mathbf{n}_{bm} \cdot \left[(\mu_{bm} + \lambda_{bm})(\boldsymbol{\nabla}\mathbf{u})_{bm} - \mu_{bm}(\boldsymbol{\nabla}\mathbf{u})_{bm}^{\mathrm{T}} \right] - \mathbf{n}_{bm} \cdot \lambda_{bm} \mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_{bm}) \right] \right\}
$$

$$
+ \frac{1}{\delta_{nm}} \left\{ \frac{\varepsilon_n \beta_m \delta_{ns}}{(2\mu_{bs} + \lambda_{bs})} \left[\mathbf{I} + \varepsilon_n \beta_m \left(\frac{\delta_{nm}}{(2\mu_{bm} + \lambda_{bm})} + \frac{\delta_{ns}}{(2\mu_{bs} + \lambda_{bs})} \right) (\mathbf{n}_{bm}^* \mathbf{n}_{bm}^*) \right]^{-1} \cdot (\mathbf{n}_{bm}^* \mathbf{n}_{bm}^*) \right.
$$

$$
\left. + \left[\mathbf{I} + \varepsilon_n \beta_m \left(\frac{\delta_{nm}}{(2\mu_{bm} + \lambda_{bm})} + \frac{\delta_{ns}}{(2\mu_{bs} + \lambda_{bs})} \right) (\mathbf{n}_{bm}^* \mathbf{n}_{bm}^*) \right]^{-1} - \mathbf{I} \right\}_{LU} \cdot \mathbf{u}_{Pm}^*
$$

$$
+ \frac{1}{\delta_{nm}} \left\{ \color{red}{\frac{\varepsilon_n \delta_{ns} \beta_m}{(2\mu_{bs} + \lambda_{bs})} \left[\mathbf{I} + \varepsilon_n \beta_m \left(\frac{\delta_{nm}}{(2\mu_{bm} + \lambda_{bm})} + \frac{\delta_{ns}}{(2\mu_{bs} + \lambda_{bs})} \right) (\mathbf{n}_{bm}^* \mathbf{n}_{bm}^*) \right]^{-1} \cdot (\mathbf{n}_{bm}^* \mathbf{n}_{bm}^*)} \right.
$$

$$
\left. \color{red}{+ \left[\mathbf{I} + \varepsilon_n \beta_m \left(\frac{\delta_{nm}}{(2\mu_{bm} + \lambda_{bm})} + \frac{\delta_{ns}}{(2\mu_{bs} + \lambda_{bs})} \right) (\mathbf{n}_{nm}^* \mathbf{n}_{bm}^*) \right]^{-1} - \mathbf{I}} \right\}_D \color{red}{\cdot \mathbf{u}_{Pm}}
$$

$$
+ \left\{ \frac{\varepsilon_n \beta_m}{(2\mu_{bm} + \lambda_{bm})} \left[\mathbf{I} + \varepsilon_n \beta_m \left(\frac{\delta_{nm}}{(2\mu_{bm} + \lambda_{bm})} + \frac{\delta_{ns}}{(2\mu_{bs} + \lambda_{bs})} \right) (\mathbf{n}_{bm}^* \mathbf{n}_{bm}^*) \right]^{-1} \cdot (\mathbf{n}_{bm}^* \mathbf{n}_{bm}^*) \right\}_{LU} \cdot \mathbf{u}_{Ps}^*
$$

$$
+ \color{blue}{\left\{ \frac{\varepsilon_n \beta_m}{(2\mu_{bm} + \lambda_{bm})} \left[\mathbf{I} + \varepsilon_n \beta_m \left(\frac{\delta_{nm}}{(2\mu_{bm} + \lambda_{bm})} + \frac{\delta_{ns}}{(2\mu_{bs} + \lambda_{bs})} \right) (\mathbf{n}_{bm}^* \mathbf{n}_{bm}^*) \right]^{-1} \cdot (\mathbf{n}_{bm}^* \mathbf{n}_{bm}^*) \right\}_D \cdot \mathbf{u}_{Ps}},
$$

(4.63)

where the red-coloured term contributes to the diagonal components of the left-hand side matrix, while the blue-coloured term represents the coupling term which is added to the corresponding off-diagonal coefficient within the left-hand side matrix. Other terms, coloured in black, are added to the right-hand side vector of the resulting system of the equations. Note that, the same as in the previous section,

some terms are decoupled into diagonal and off-diagonal components to ensure the best possible efficiency while using the segregated solution procedure. During the iterative procedure, all these terms are updated, meaning that the left-hand side matrix as well as the right-hand side vector evolve to the final solution. For the boundary faces that are not in contact, the relative contact area is $\beta = 0$, consequently Eqs. (4.62) and (4.63) are reduced to the form which presents a stress-free boundary.

For the slave side, the expression for boundary displacement and boundary gradient is obtained following the same procedure. The slave boundary displacement is obtained by combining Eqs. (4.56), (4.59) and (4.61), whereas the boundary gradient is obtained by combining the resulting equation with (4.50).

Introducing equality $\mathbf{n}_{bm} = -\mathbf{n}_{bs}$ it is possible to further simplify the presented equations, however, it is not done here to maintain the possibility of extending them to non-conformal contact scenarios.

Before proceeding to the non-conformal discretisation at the contact interface, the presented equations are rewritten in a compact form by introducing the following substitutions:

$$c_i = \frac{\delta_{ni}}{(2\mu_{bi} + \lambda_{bi})}, \tag{4.64}$$

$$\mathbf{a}_i = \mathbf{n}_{bi} \cdot \left[(\mu_{bi} + \lambda_{bi})(\boldsymbol{\nabla}\mathbf{u})_{bi}^* - \mu_{bi}(\boldsymbol{\nabla}\mathbf{u})_{bi}^{*,\mathrm{T}} \right] - \mathbf{n}_{bi} \cdot \lambda_{bi}\mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_{bi}^*), \tag{4.65}$$

$$\mathbf{N}_i = \mathbf{n}_{bi}^* \mathbf{n}_{bi}^*, \tag{4.66}$$

$$\mathbf{G}_i = \mathbf{I} + \varepsilon_n \beta_i (c_i + c_h) \mathbf{N}_i, \tag{4.67}$$

$$\mathbf{e}_i = \varepsilon_n c_i \mathbf{N}_i \cdot (\mathbf{n}_{bi}^*(g_n)_{bi}^* + \beta_i \mathbf{u}_{bi}^* - \beta_i \mathbf{u}_{bh}^*) + \varepsilon_n \beta_i c_i c_h \mathbf{N}_i \cdot (\mathbf{a}_i + \mathbf{a}_h) + c_i \mathbf{a}_i, \tag{4.68}$$

where subscripts i and h represent the current and the adjacent boundary, respectively. By substituting m for i and s for h the equation set for the master side is obtained. The same is valid for the slave side — s is substituted for i and m for h. Using the above substitutions, the compact expression for the boundary displacement is:

$$\mathbf{u}_{bi} = \mathbf{G}_i^{-1} \cdot \mathbf{e}_i + \mathbf{G}_i^{-1} \cdot (\mathbf{I} + \varepsilon_n \beta_i c_h \mathbf{N}_i) \cdot \mathbf{u}_{Pi} + \varepsilon_n \beta_i c_i \mathbf{G}_i^{-1} \cdot \mathbf{N}_i \cdot \mathbf{u}_{Ph}, \tag{4.69}$$

and for the boundary gradient the expression is:

$$
\begin{aligned}
\mathbf{g}_{bi} = {}& \delta_{ni}^{-1} \mathbf{G}_i^{-1} \boldsymbol{\cdot} \mathbf{e}_i \\
& + \delta_{ni}^{-1} \left(\mathbf{G}_i^{-1} + \varepsilon_n \beta_i c_h \mathbf{G}_i^{-1} \boldsymbol{\cdot} \mathbf{N}_i - \mathbf{I} \right)_{LU} \boldsymbol{\cdot} \mathbf{u}_{Pi}^* \\
& + \textcolor{red}{\delta_{ni}^{-1} \left(\mathbf{G}_i^{-1} + \varepsilon_n \beta_i c_h \mathbf{G}_i^{-1} \boldsymbol{\cdot} \mathbf{N}_i - \mathbf{I} \right)_{D} \boldsymbol{\cdot} \mathbf{u}_{Pi}} \\
& + \delta_{ni}^{-1} (\varepsilon_n \beta_i c_i \mathbf{G}_i^{-1} \boldsymbol{\cdot} \mathbf{N}_i)_{LU} \boldsymbol{\cdot} \mathbf{u}_{Ph}^* \\
& + \textcolor{blue}{\delta_{ni}^{-1} (\varepsilon_n \beta_i c_i \mathbf{G}_i^{-1} \boldsymbol{\cdot} \mathbf{N}_i)_{D} \boldsymbol{\cdot} \mathbf{u}_{Ph}}.
\end{aligned}
\tag{4.70}
$$

Again, the red-coloured term contributes to the diagonal coefficient, whereas the blue-coloured term contributes to the corresponding off-diagonal coefficient in the left-hand side matrix.

Boundary non-orthogonality correction

The above-presented equations are derived with the assumption of a perfectly orthogonal mesh at contact boundaries, as shown in Fig. 4.30 a). Preserving the boundary orthogonality during the mesh generation is an impractical task, especially for the finite volume method which facilitates the application of automatic mesh generation. In addition, for many large deformation contact problems, the initial orthogonality cannot be preserved during the subsequent body deformation.
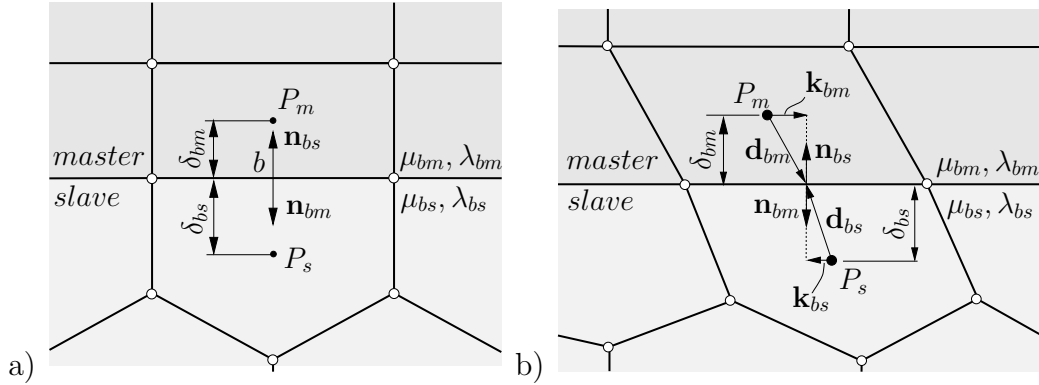

Figure 4.30. Cells sharing conformal contact interface: a) orthogonal cells and b) non-orthogonal cells

As explained in Section 3.2.3, the second-order accuracy at the boundary is preserved by adding non-orthogonal correction term $\mathbf{u}_{bi}^{corr}$ to the boundary displacement calculated using Eq. (4.69). The correction term is calculated using the available

4. Contact treatment

gradient at the cell centre as follows:

$$\mathbf{u}_{bi}^{corr} = \mathbf{k}_{bi} \cdot \boldsymbol{\nabla} \mathbf{u}_{Pi}^*, \tag{4.71}$$

where $\mathbf{k}_{bi} = \mathbf{d}_{bi} - \mathbf{n}_{bi}(\mathbf{d}_{bi} \cdot \mathbf{n}_{bi})$, see Fig. 4.30 b). The same procedure is followed during the explicit calculation of the boundary normal gradient, which is used to calculate stress at the boundary.

Non-conformal discretisation at the contact interface

Using the derived equations for the conformal contact interface, the extension to the non-conformal discretisation is straightforward if for the adjacent contacted cell a virtual cell is constructed. For each cell at boundary i, the adjacent virtual cell at boundary h is composed of cells sharing its boundary face. The construction of the virtual cell is performed geometrically by using area-weights, which is a concept used by the GGI interpolation for the implicit coupling of the non-conformal internal boundaries.

For each cell j at boundary i, the adjacent virtual cell is composed of k cells which are in contact with cell j at the contact interface. Cell centre displacement of the virtual cell $\mathbf{u}_{Ph,j}^v$ is determined using the area-weight of each cell k, see Eqs. (4.5) and (4.7):

$$\mathbf{u}_{Ph,j}^v = \sum_k \omega_{j,k}\mathbf{u}_{Ph,k}. \tag{4.72}$$

Weighting factors $\omega_{j,k}$ are calculated within the segment-to-segment algorithm (step c) in Fig. 4.19), thus no additional computational overhead is added. Substituting Eq. (4.72) into Eqs. (4.69) and (4.70), implicit equations for boundary displacement and boundary gradient are obtained; the equations maintain the same form and the only difference is the substitution of $\mathbf{u}_{Ph,j}$ with $\mathbf{u}_{Ph,j}^v$.

To test the implementation, two examples with non-conformal discretisation are solved: a Hertzian contact between sphere and block (case description is given in subsection 5.2.5) and a contact between cylindrical punch and foundation (case description is given in subsection 5.2.4). As in the previous section, both examples are solved with various values of the penalty parameter and with the optimal value of the under-relaxation factor. Figure 4.31 a) shows the normalised pressure distribution on the cylindrical foundation for the case of the contact between punch and foundation, while Fig. 4.31 b) shows normalised subsurface stresses along the z-axis for the case of the contact between sphere and block. As shown, the implementa-

tion handles the non-conformal discretisation correctly and a consistent behaviour is obtained; the resulting stress distribution is not affected by the coupling type. Table 4.6 summarises the results of the convergence behaviour for the case of the contact between cylindrical punch and cylindrical foundation, and Table 4.7 for the case of the contact between sphere and block. By comparing the results, it can be noticed that the optimally tuned explicit coupling can converge the same as the implicit coupling only for lower values of the penalty parameter. For the case of the contact between cylindrical punch and cylindrical foundation, the explicit coupling shows the same level of efficiency for the case with the resulting average gap value (averaged gap divided by average contact cell height) of 1%. This can be attributed to the minimal number of faces in partial contact (axisymmetric computational model). The same is observed for the 2D case of the cylinder and rigid block (see Table 4.5). Nevertheless, it should be noted that the efficiency of the explicit coupling can degrade rapidly for a slight change in the under-relaxation. Furthermore, in practice, the explicit coupling is never optimally tuned, which makes the implicit coupling even more superior in terms of efficiency.
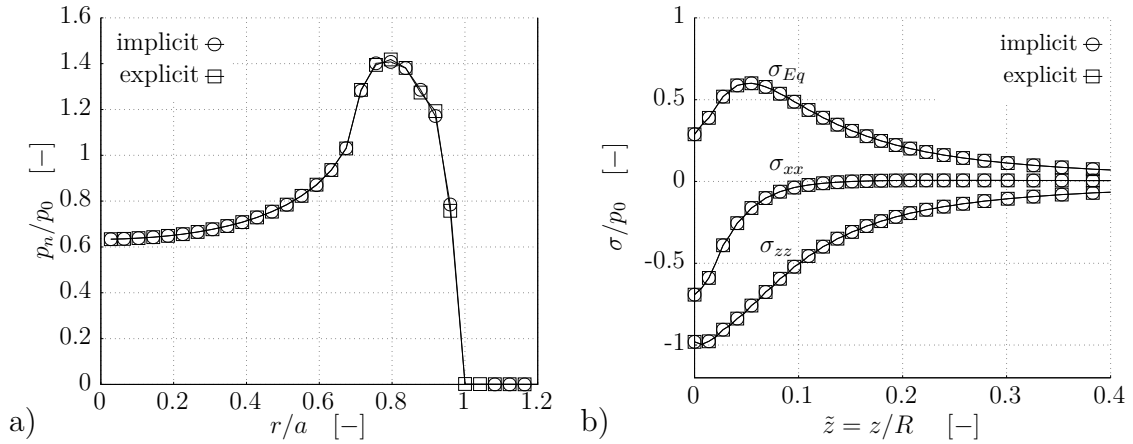
Figure 4.31. Comparison of stress distribution between explicit and implicit contact coupling: a) contact between cylindrical punch and cylindrical foundation and b) contact between sphere and block (fine mesh)

Table 4.6. Comparison between efficiency of implicit and explicit coupling for different values of penalty parameter — contact between cylindrical punch and cylindrical foundation

ε_n/E_s	Implicit n_{corr}	Explicit (optimal) n_{corr}	Avg. relative gap [%]
100	105	98	3
1,000	106	110	0.3
10,000	105	755	0.04

Table 4.7. Comparison between efficiency of implicit and explicit coupling for different values of penalty parameter — contact between sphere and block

ε_n/E_s	Implicit n_{corr}	Explicit (optimal) n_{corr}	Avg. relative gap [%]
10	73	68	11
100	79	230	1.5
1,000	81	903	0.2

Large deformation contact — hyperplastic constitutive relation

The last step is to extend the implicit contact coupling to account for nonlinear material behaviour. The starting equation for the boundary traction (4.49) is restricted to an isothermal linear-elastic material and to extend it to a general form, term $(2\mu_b + \lambda_b)\mathbf{n}_b \cdot (\boldsymbol{\nabla}\mathbf{u})_b - (2\mu_b + \lambda_b)\mathbf{n}_b \cdot (\boldsymbol{\nabla}\mathbf{u})_b$ is added:

$$
\begin{aligned}
\mathbf{t}_b = (2\mu_b + \lambda_b)\mathbf{n}_b \cdot (\boldsymbol{\nabla}\mathbf{u})_b &- (2\mu_b + \lambda_b)\mathbf{n}_b \cdot (\boldsymbol{\nabla}\mathbf{u})_b \\
&+ \underbrace{\mu_b\mathbf{n}_b \cdot (\boldsymbol{\nabla}\mathbf{u})_b + \mu_b\mathbf{n}_b \cdot (\boldsymbol{\nabla}\mathbf{u})_b^{\mathrm{T}} + \lambda_b\mathbf{n}_b\mathrm{tr}((\boldsymbol{\nabla}\mathbf{u})_b)}_{\mathbf{n}_b \cdot \sigma_b}
\end{aligned}
\tag{4.73}
$$

Rearranging Eq. (4.73) the equation for the boundary gradient is obtained:

$$
\mathbf{g}_b = \mathbf{n}_b \cdot (\boldsymbol{\nabla}\mathbf{u})_b = \frac{1}{(2\mu_b + \lambda_b)}(\mathbf{t}_b - \mathbf{n}_b \cdot \sigma_b) + \mathbf{n}_b \cdot (\boldsymbol{\nabla}\mathbf{u})_b,
\tag{4.74}
$$

Using Eq. (4.74) as a starting equation instead of Eq. (4.49) the same set of equations is derived (4.64)–(4.70), with the only difference in term $\mathbf{a}_i$, which is transformed into the following form:

$$
\mathbf{a}_i = (2\mu_{bi} + \lambda_{bi})\mathbf{n}_{bi} \cdot (\boldsymbol{\nabla}\mathbf{u})_{bi}^* - \mathbf{n}_{bi}^* \cdot \boldsymbol{\sigma}_{bi}^*.
\tag{4.75}
$$

With this last modification, the procedure is ready to be tested on the contact problems with large deformations and large sliding of the bodies in contact.

Two examples are examined: shallow ironing (case description is given in subsection 5.3.2) and a contact between two curved beams (case description is given in subsection 5.3.3, beams are modelled as hyperelastic with the same Young's modulus). Since these examples are solved incrementally, the typically used value of the under-relaxation factor is prescribed. Both examples are solved using explicit and implicit contact coupling, and the difference between the resulting forces and the convergence speed is analysed. Figure 4.32 shows the evolution of the horizontal and the vertical reaction force in the shallow ironing example. One can see that the

curves match, which means that the implicit coupling can produce results that are consistent with the results obtained using explicit coupling. This is also confirmed in the second example, see Fig 4.34.

Figure 4.33 shows the achieved speed-up for the shallow ironing example. Speed-up is measured as a ratio between the number of outer iterations required for the implicit and the explicit coupling to converge per time step. Since the cost of each outer iteration is approximately the same, the used ratio can faithfully depict the speed-up. In the shallow ironing example, the implicit coupling shows a more than ten times faster convergence. In the second example, the achieved speed-up varies between the factors of two and four, as shown in Fig. 4.35. Although the second example shows less speed-up, reducing the computational time by the factor of two can be still described as a noteworthy improvement.

The application of the implicit coupling to the shallow ironing example is shown to be stable with high values of the penalty parameter. However, the contact between two curved beams is incapable of converging with an acceptable value of the penalty parameter, i.e., the resulting averaged relative penetration is 12% which can be considered high. While the sources of this problem can be numerous, a valid guess is that the calculation of the contact force using solely information from the previous iteration is not an enough stable approach when the segregated solution procedure is used. Further, failure occurs when the position of the contact interface is close to 45 degrees. In such configurations, the contribution of the coupling term is minimal, resulting in a reduced implicitness of the coupling procedure. It is believed that by performing some type of augmentation of the contact force stability can be improved, thus further development is necessary to establish an implicit coupling procedure with a stability equal to the explicit coupling procedure.

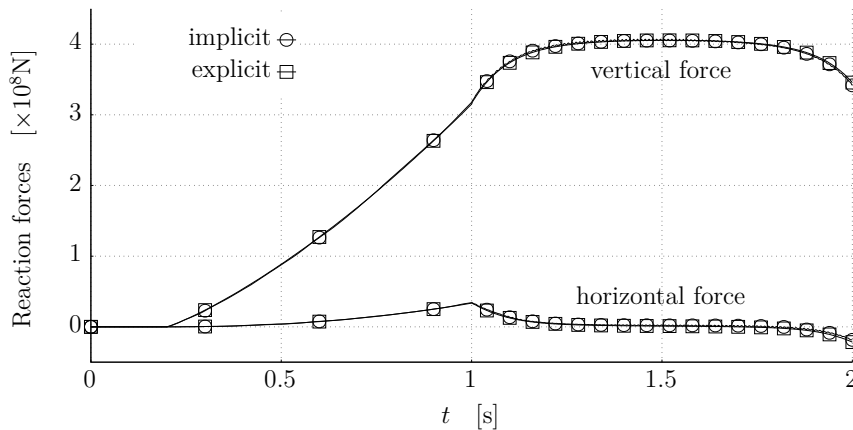Figure 4.32. Shallow ironing: comparison of reaction forces
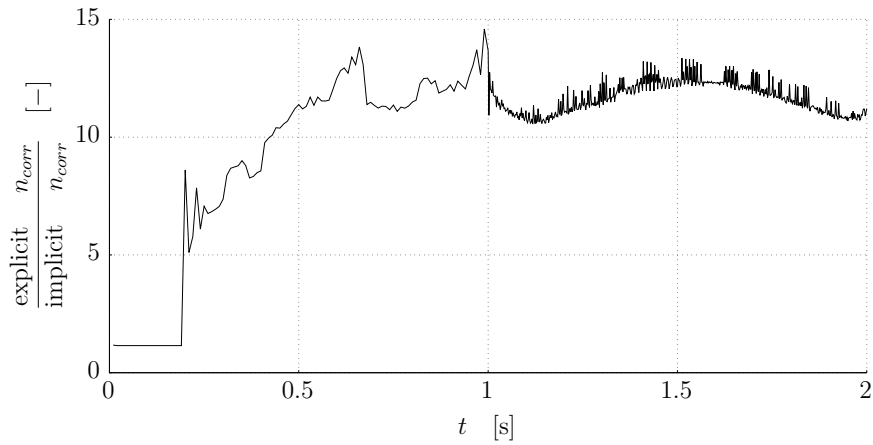
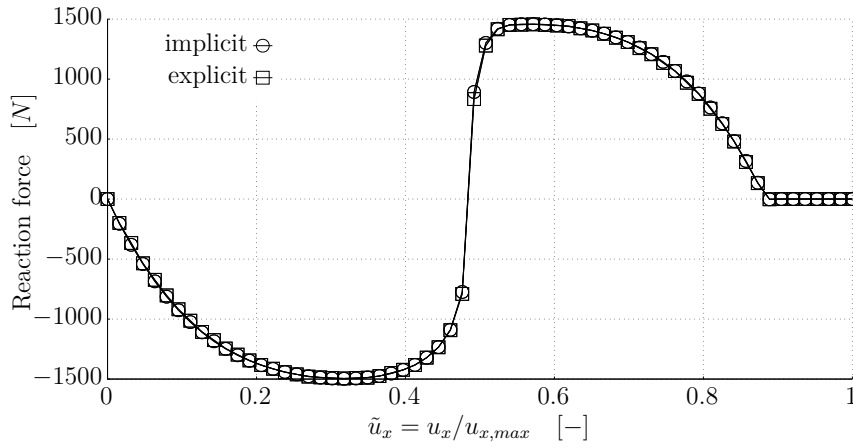Figure 4.33. Shallow ironing: speed-up measured in terms of number of outer iterations per time-step

Figure 4.34. Contact between curved beams: comparison of total reaction force in y-direction
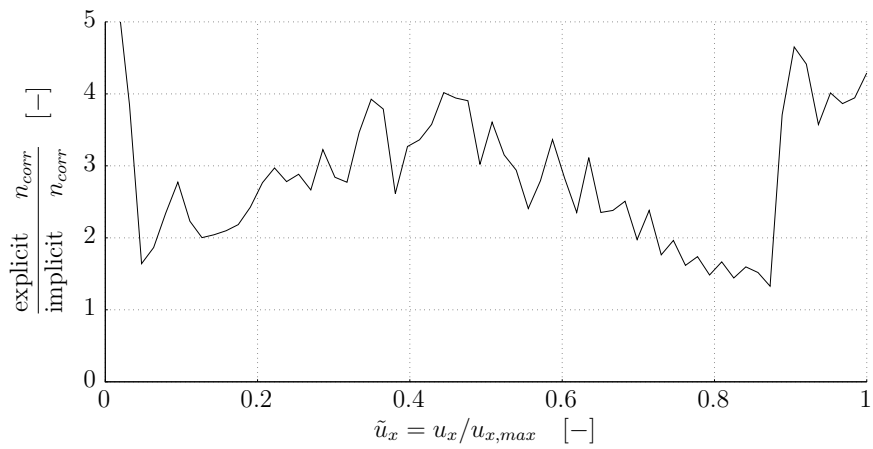

Figure 4.35. Contact between curved beams: speed-up measured in terms of number of outer iterations per prescribed displacement increment

4.6. Surface Smoothing

Surface smoothing, or the so-called contact smoothing, is a procedure in which higher-order interpolations are employed to produce a smooth description of the contact surface (without altering the underlying bulk discretisation). Generally, surface smoothing is conducted mostly for two reasons:

- Lower error in geometry description results in a more accurate prediction of contact stresses.

- Smoothing the surface normal vector field has a positive effect on the convergence and robustness of the contact treatment.

Surface smoothing is extensively investigated in the FEM literature, and many smoothing techniques were proposed for low order finite elements [135]. Typically, surface smoothing is employed with the goal to alleviate common issues of the NTS contact discretisation, i.e. to reduce the non-physical oscillations of the contact force during the large sliding, and to improve the robustness and convergence rate of the solution procedure [6, 135]. For the NTS contact discretisation, only the master surface is smoothed, meaning that contact interaction occurs between slave nodes and the smoothed master surface. For 2D problems, the following interpolation methods were proposed: cubic spline interpolation [136], non-uniform rational basis splines [137], cubic Bézier [89] and Hermite interpolation [138]. Extension of these interpolations to 3D is extremely difficult, and it is restricted only to the structured quadrilateral surface mesh topology [138]. Accordingly, for unstructured surface meshes composed of the quadrilateral faces interpolation using Gregory patches is proposed in [139]. For unstructured triangular meshes, [140] proposed the interpolation using quartic Bézier patches. The only interpolation capable of handling the arbitrary surface mesh topology is the interpolation using Nagata patches [74], thus making it the most suitable candidate for the unstructured finite volume method.

The incorporation of the surface smoothing in the contact algorithm adds a considerable computational overhead, which must be kept in mind since the contact algorithm is invoked within each outer iteration of the segregated solution procedure. Nevertheless, there are many cases in which the usage of the smoothing procedure can be quite desirable or justified. The best example is the contact between a rigid and a deformable body in which the smoothing of the rigid body is performed only once. Also, the rigid body can have coarse underlying discretisation, thus minimizing the cost of contact detection.

In the remaining part of this section, the surface smoothing using the Nagata interpolation is outlined for the interpolation of triangular and quadrilateral faces. The smoothing procedure is incorporated with the pointwise contact algorithm to smooth the master surface. The main features of such a procedure are addressed using a few simple numerical examples. To the best of the author's knowledge, this is the first attempt to use the surface smoothing for solving contact problems using the finite volume method.

4.6.1. Surface Smoothing Using Nagata Interpolation

The Nagata patch interpolation was originally introduced in [73] as a efficient interpolation which can overcome drawbacks of the existing surface interpolation methods at the time. It should be noted that here patch denotes a single boundary face, while in the finite volume terminology patch refers to a group of boundary faces. The interpolation completely local and it is based on the quadratic polynomial, requiring only position and normal vectors at the vertices of the surface mesh. Also, it is applicable to the arbitrary geometry topology and its formulation can easily handle the discontinuity of normals, sharp edges and singular points.

The first step in the interpolation is the construction of edge curves which are afterwards used to complete the definition of patch interior. For the edge with position vectors of the corresponding end points and unit normals at the end points the Nagata curve is defined using a quadratic polynomial as (see Fig. 4.36):

$$\mathbf{x}(\zeta) = \mathbf{x}_0 + (\mathbf{d} - \mathbf{c})\zeta + \mathbf{c}\zeta^2, \tag{4.76}$$

where $\mathbf{d} = \mathbf{x}_1 - \mathbf{x}_0$ is the vector joining the end points and ζ is the local coordinate which satisfied the condition $0 \leqslant \zeta \leqslant 1$.
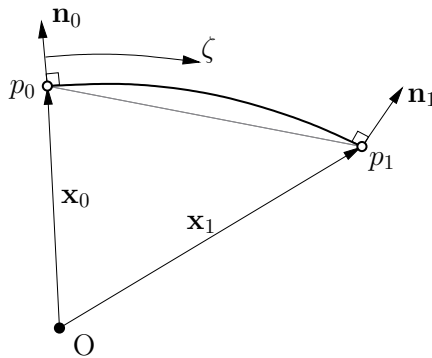

Figure 4.36. Nagata interpolation of surface edge

The unknown coefficient vector $\mathbf{c}$, called the curvature parameter, adds curvature to the edge. Assuming that the interpolated curve is orthogonal to endpoints unit normals $\mathbf{n}_0$ and $\mathbf{n}_1$, coefficient vector $\mathbf{c}$ is obtained by minimising its norm [73]:

$$\mathbf{c}(\mathbf{d}, \mathbf{n}_0, \mathbf{n}_1) = \begin{cases} \dfrac{[\mathbf{n}_0, \mathbf{n}_1]}{1 - a^2} \begin{bmatrix} 1 & -a \\ -a & 1 \end{bmatrix} \begin{Bmatrix} \mathbf{n}_0 \cdot \mathbf{d} \\ -\mathbf{n}_1 \cdot \mathbf{d} \end{Bmatrix} & (a \neq \pm 1) \\[3mm] \dfrac{[\mathbf{n}_0, \pm\mathbf{n}_0]}{2} \begin{Bmatrix} \mathbf{n}_0 \cdot \mathbf{d} \\ \mp\mathbf{n}_0 \cdot \mathbf{d} \end{Bmatrix} = 0 & (a = \pm 1) \end{cases}, \qquad (4.77)$$

where a is the scalar product between end unit normals $a = \mathbf{n}_0 \cdot \mathbf{n}_1$. In the case that the normals are parallel $|a| = 1$, the curvature adding vector is zero and the interpolated edge matches to the linear edge (exact interpolation).

Figure 4.37 shows the interpolated surface of the triangular and quadrilateral boundary faces with corresponding input parameters: end points position vectors and end points unit normals. The triangular and quadrilateral boundary faces, considered in this thesis, are the most common types of boundary faces; however, the Nagata interpolation can be extended to n-sided polygonal faces, thus making it a convenient candidate for the unstructured finite volume method.

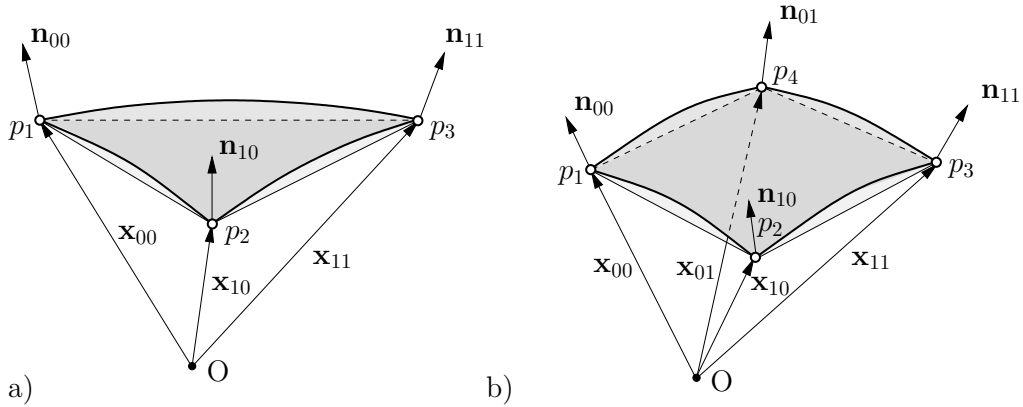Figure 4.37. Nagata patch interpolation for: a) triangular faces, b) quadrilateral faces

For the triangular Nagata patch, the quadratic interpolation polynomial has the following form:

$$\mathbf{x}_\triangle(\eta, \zeta) = \mathbf{c}_{00} + \mathbf{c}_{10}\eta + \mathbf{c}_{01}\zeta + \mathbf{c}_{11}\eta^2 + \mathbf{c}_{02}\zeta^2, \qquad (4.78)$$

while in the case of the quadrilateral Nagata patch the quadratic interpolation

4. Contact treatment

polynomial has additional two terms ($\eta^2\zeta$ and $\eta\zeta^2$):

$$\mathbf{x}_\square(\eta, \zeta) = \mathbf{c}_{00} + \mathbf{c}_{10}\eta + \mathbf{c}_{01}\zeta + \mathbf{c}_{11}\eta\zeta + \mathbf{c}_{20}\eta^2 + \mathbf{c}_{21}\eta^2\zeta + \mathbf{c}_{12}\eta\zeta^2, \qquad (4.79)$$

where η and ζ are the local coordinates. The domain of the local coordinates for both cases is shown in Fig. 4.38. The domain of the local coordinates for the triangular patch is defined as $0 \leqslant \zeta \leqslant \eta \leqslant 1$, while for the quadrilateral patch it is defined as $0 \leqslant \zeta, \eta \leqslant 1$.
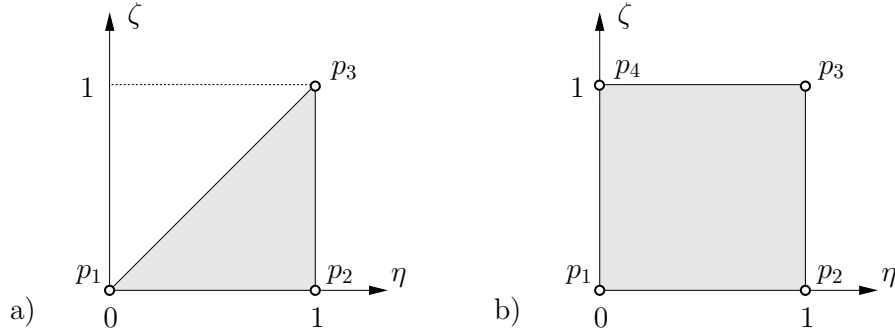

Figure 4.38. Patch domain in local coordinates for: a) triangular faces, b) quadrilateral faces

The coefficient vectors for the triangular and quadrilateral Nagata patch are defined via the curvature parameters and the known point positions, summarised in Table 4.8.

Table 4.8. Coefficient vectors for triangular and quadrilateral Nagata patch

$\mathbf{x}_\triangle(\eta, \zeta)$	$\mathbf{x}_\square(\eta, \zeta)$
$\mathbf{c}_{00} = \mathbf{x}_{00}$	$\mathbf{c}_{00} = \mathbf{x}_{00}$
$\mathbf{c}_{10} = \mathbf{x}_{10} - \mathbf{x}_{00} - \mathbf{c}_1$	$\mathbf{c}_{10} = \mathbf{x}_{10} - \mathbf{x}_{00} - \mathbf{c}_1$
$\mathbf{c}_{01} = \mathbf{x}_{11} - \mathbf{x}_{10} + \mathbf{c}_1 - \mathbf{c}_3$	$\mathbf{c}_{01} = \mathbf{x}_{01} - \mathbf{x}_{00} - \mathbf{c}_4$
$\mathbf{c}_{11} = \mathbf{c}_3 - \mathbf{c}_1 - \mathbf{c}_2$	$\mathbf{c}_{11} = \mathbf{x}_{11} - \mathbf{x}_{10} - \mathbf{x}_{01} + \mathbf{x}_{00} + \mathbf{c}_1 - \mathbf{c}_2 - \mathbf{c}_3 + \mathbf{c}_4$
$\mathbf{c}_{20} = \mathbf{c}_1$	$\mathbf{c}_{20} = \mathbf{c}_1$
$\mathbf{c}_{02} = \mathbf{c}_2$	$\mathbf{c}_{02} = \mathbf{c}_4$
	$\mathbf{c}_{21} = \mathbf{c}_3 - \mathbf{c}_1$
	$\mathbf{c}_{12} = \mathbf{c}_2 - \mathbf{c}_4$

By applying Eq. (4.77) to each edge of the boundary face, the curvature parameters are determined. Table 4.9 depicts input arguments for the curvature parameter function.

Table 4.9. Curvature parameters for edges of triangular and quadrilateral Nagata patch

$\mathbf{x}_{\triangle}(\eta, \zeta)$		$\mathbf{x}_{\square}(\eta, \zeta)$	
$\mathbf{d}_1 = \mathbf{x}_{10} - \mathbf{x}_{00},$	$\mathbf{c}_1 = \mathbf{c}(\mathbf{d}_1, \mathbf{n}_{00}, \mathbf{n}_{10})$	$\mathbf{d}_1 = \mathbf{x}_{10} - \mathbf{x}_{00},$	$\mathbf{c}_1 = \mathbf{c}(\mathbf{d}_1, \mathbf{n}_{00}, \mathbf{n}_{10})$
$\mathbf{d}_2 = \mathbf{x}_{11} - \mathbf{x}_{10},$	$\mathbf{c}_2 = \mathbf{c}(\mathbf{d}_2, \mathbf{n}_{10}, \mathbf{n}_{11})$	$\mathbf{d}_2 = \mathbf{x}_{11} - \mathbf{x}_{10},$	$\mathbf{c}_2 = \mathbf{c}(\mathbf{d}_2, \mathbf{n}_{10}, \mathbf{n}_{11})$
$\mathbf{d}_3 = \mathbf{x}_{11} - \mathbf{x}_{00},$	$\mathbf{c}_3 = \mathbf{c}(\mathbf{d}_3, \mathbf{n}_{00}, \mathbf{n}_{11})$	$\mathbf{d}_3 = \mathbf{x}_{11} - \mathbf{x}_{01},$	$\mathbf{c}_3 = \mathbf{c}(\mathbf{d}_3, \mathbf{n}_{01}, \mathbf{n}_{11})$
		$\mathbf{d}_4 = \mathbf{x}_{01} - \mathbf{x}_{00},$	$\mathbf{c}_4 = \mathbf{c}(\mathbf{d}_4, \mathbf{n}_{00}, \mathbf{n}_{01})$

The definition of the curvature parameter in Eq. (4.77) can lead to stability problems manifested in inflated and wrongly oriented edge curves. These problems arise for singular cases in which the unit normals are near parallel (zero denominators in Eq. (4.77)). Accordingly, special care must be taken for the numerical implementation to avoid this drawback; [141] proposed the usage of a threshold value to increase the domain of the linear interpolation near the singularity. Another possible solution is to modify the equation for the curvature parameter to be less strict in the enforcement of the surface orthogonality to imposed normal vectors, see [6, 142] for further details.

By defining the initially smooth surface using analytical functions, the level of accuracy achieved by employing the Nagata interpolation can easily be determined. Such an analysis is not conducted here as it is already available in the literature [6, 73]. When compared to the piecewise linear description of the boundary, the results show that the Nagata interpolation provides a two orders of magnitude lower error with a higher error reduction rate.

Point normal calculation

The construction of the Nagata patch requires two inputs, the points position vectors and the points unit normal vectors. Usually, the unit normals at the boundary points are not determined, as the face normal is used for all calculations. For the calculation of the discrete surface curvature, i.e. points normals, two different approaches exist in the literature [6]. In the first approach, mesh points are fitted with a smooth parametric surface and point normals are subsequently calculated using the newly generated surface [143]. In the second approach, the point normal is determined using the weighted average of the surrounding face normals. A special case is a rigid-deformable contact in which for the smoothed rigid body the evaluation can be conducted either analytically for simpler geometries or from a CAD model used for mesh generation [144].

97

4. Contact treatment

In this thesis, the calculation of the points normal vector is conducted using the weighted sum of the surrounding faces normals:

$$\mathbf{n}_p = \frac{\sum_i \omega_i \mathbf{n}_{b,i}}{||\sum_i \omega_i \mathbf{n}_{b,i}||} \tag{4.80}$$

where ω_i denotes the face weighting factor and boundary face normal $\mathbf{n}_b$, is calculated using central decomposition. As shown in Fig. 4.39, it is possible to use different topological quantities for the calculation of the face weighting factor. Here, the simplest method in which each surrounding facet has equal weight is adopted. To improve accuracy, other methods listed in Table 4.10 can be used. A detailed discussion and comparison of these methods is available in [6].
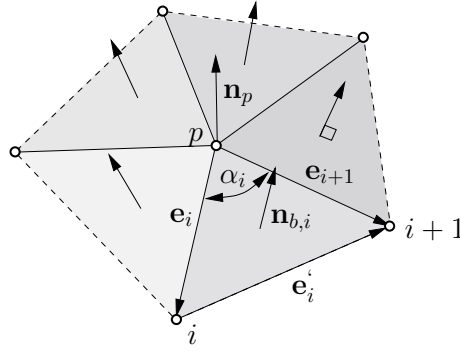

Figure 4.39. Surrounding faces of point p with calculated unit normal vector $\mathbf{n}_p$. Notation for weighted average calculation is included

Table 4.10. Different methods for calculation of weighting factor ω_i

Mean weighted by:	Weighting factor
Equally [145]	$\omega_i = 1$
Angle [146]	$\omega_i = \alpha_i$
Area [147]	$\omega_i = ||\mathbf{e}_i \times \mathbf{e}_{i+1}||$
Edge length reciprocal [147]	$\omega_i = (\,||\mathbf{e}_i||\,||\mathbf{e}_{i+1}||\,)^{-1}$
Angle and edge length reciprocal [147]	$\omega_i = \sin(\alpha_i)(\,||\mathbf{e}_i||\,||\mathbf{e}_{i+1}||\,)^{-1}$
Angle and area of circumscribed circle [148]	$\omega_i = \sin(\,\alpha_i/||\mathbf{e}_i^{\backprime}||\,)^2\,\alpha_i$

The calculation using the weighted sum of the normal vectors is a robust, simple and computationally cheap approach with a good level of accuracy. In some specific cases, to preserve accuracy, the method required a simple correction of the resulting normal [6]. The examples are symmetry boundaries and transitional zones between flat and curved parts of the surface, as shown in Fig 4.40.
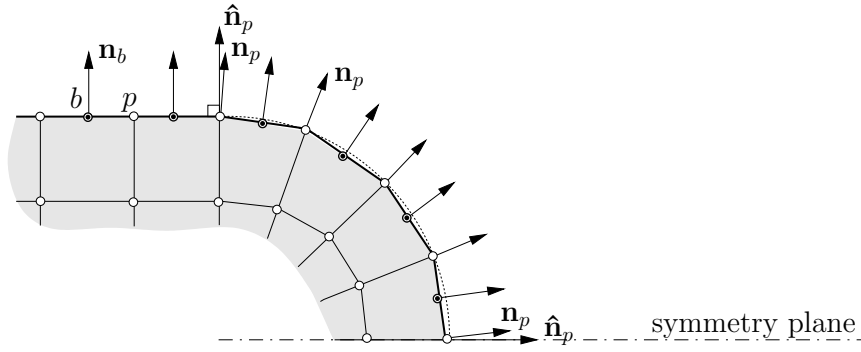
Figure 4.40. Correction of calculated point normals at symmetry plane and at transition point. Corrected unit normal is denoted as $\hat{\mathbf{n}}_p$

Normal distance function

For the Nagata patch, the calculation of normal distance function g_n requires more effort since the interpolated boundary face is not planar.
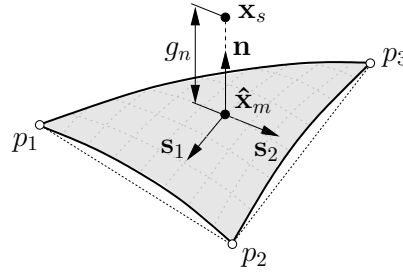

Figure 4.41. Orthogonal projection of generic point $\mathbf{x}_s$ onto triangular Nagata patch

The closest point between a generic point on slave surface $\mathbf{x}_s$ and the Nagata patch on the master surface (see Fig. 4.41) results in an orthogonal projection problem formulated as:

$$\mathbf{f}(\eta, \zeta, g_n) = \hat{\mathbf{x}}_m(\eta, \zeta) + g_n \mathbf{n}(\eta, \zeta) - \mathbf{x}_s = 0, \tag{4.81}$$

where $\hat{\mathbf{x}}_m$ is the corresponding projected surface point, $\mathbf{n}$ is the surface outward unit normal vector and $\mathbf{f}$ is the solution vector which satisfies $\mathbf{f} = 0$. The outward unit normal vector is defined by a cross product of the patch partial derivatives:

$$\mathbf{n}(\eta, \zeta) = \frac{\dfrac{\partial \mathbf{x}(\eta, \zeta)}{\partial \eta} \times \dfrac{\partial \mathbf{x}(\eta, \zeta)}{\partial \zeta}}{\left\| \dfrac{\partial \mathbf{x}(\eta, \zeta)}{\partial \eta} \times \dfrac{\partial \mathbf{x}(\eta, \zeta)}{\partial \zeta} \right\|} = \frac{\mathbf{s}_1 \times \mathbf{s}_2}{||\mathbf{s}_1 \times \mathbf{s}_2||}, \tag{4.82}$$

where $\mathbf{s}_1$ and $\mathbf{s}_1$ are the surface tangent vectors corresponding to partial derivatives. The equations for the calculation of the partial derivatives in the case of the triangular and quadrilateral Nagata patches are given in Table 4.11. In order to solve the nonlinear system of equations $\mathbf{f}(\eta, \zeta, g_n)=0$, the Newton–Raphson iterative algorithm is used, formulating Eq. (4.81) into:

$$\mathbf{r}^{(i+i)} = \mathbf{r}^{(i)} - [\boldsymbol{\nabla}\mathbf{f}(\mathbf{r}^{(i)})]^{-1}\mathbf{f}(\mathbf{r}^{(i)}), \tag{4.83}$$

where $\mathbf{r} = [\eta, \zeta, g_n]^{\mathrm{T}}$ is the solution vector containing the local coordinates of the closest point and its normal distance. Jacobian matrix $\boldsymbol{\nabla}\mathbf{f}$ of the system of equations is defined as:

$$
\begin{aligned}
\boldsymbol{\nabla}\mathbf{f}(\mathbf{r}) &= \left[\frac{\partial}{\partial\eta}, \frac{\partial}{\partial\zeta}, \frac{\partial}{\partial g_n}\right]\mathbf{f}(\mathbf{r}) \\
&= \left[\frac{\partial\hat{\mathbf{x}}_m}{\partial\eta} + g_n\frac{\partial\mathbf{n}}{\partial\eta}, \ \frac{\partial\hat{\mathbf{x}}_m}{\partial\zeta} + g_n\frac{\partial\mathbf{n}}{\partial\zeta}, \ \mathbf{n}\right]
\end{aligned}
\tag{4.84}
$$

The partial derivatives in Eq. (4.84) are given in Table 4.11, while the gradient of the normal vector with respect to the local coordinates is calculated using the Weingarten formula [10, 89].

Table 4.11. Patch surface tangent vectors definition

	$\triangle$	$\square$
$\mathbf{s}_1 = \dfrac{\partial\mathbf{x}(\eta, \zeta)}{\partial\eta} =$	$\mathbf{c}_{10} + \mathbf{c}_{11}\zeta + 2\mathbf{c}_{20}\eta$	$\mathbf{c}_{10} + \mathbf{c}_{11}\zeta + 2\mathbf{c}_{20}\eta + 2\mathbf{c}_{21}\eta\zeta + \mathbf{c}_{12}\zeta^2$
$\mathbf{s}_2 = \dfrac{\partial\mathbf{x}(\eta, \zeta)}{\partial\zeta} =$	$\mathbf{c}_{01} + \mathbf{c}_{11}\eta + 2\mathbf{c}_{02}\zeta$	$\mathbf{c}_{01} + \mathbf{c}_{11}\eta + 2\mathbf{c}_{02}\zeta + \mathbf{c}_{21}\eta^2 + 2\mathbf{c}_{12}\eta\zeta$

Incorporation with the pointwise contact algorithm

To demonstrate potential benefits in using the surface smoothing, the Nagata surface interpolation method is incorporated within the pointwise contact algorithm, described in Section 4.3. Smoothing is performed for the chosen master boundary, meaning that points (vertices) of the slave surface are in interaction with the Nagata patches constructed on the master side. Accordingly, violation of the impenetrability condition is checked between the slave points and the smoothed description of the master surface. For each slave point, its corresponding neighbouring Nagata patches are determined and the projection point with the minimum normal gap value is selected as the solution. The normal gap is calculated iteratively by solving Eq.

(4.81) using the Newton-Raphson algorithm, described in the previous subsection. For the initial guess of the solution, the midpoint of the Nagata patch is prescribed. Due to the quadratic rate of convergence of the Newton-Raphson algorithm, the solution is typically found in three to four iterations. Nonetheless, to avoid any problems which can potentially arise, the maximum number of iterations is limited to ten. Finally, the normal component of the contact traction is calculated at the slave face-centres using Eq. (4.11).

Numerical example: block sliding in a half-tube

In this example, initially proposed in [140], the large frictional sliding of an elastic block in a rigid tube is considered. The radius and length of the tube are 3 and 15 m, respectively. The block dimensions are $2 \times 2 \times 2$ m with a contact surface curved the same way as the tube. Block Young's modulus is 100 Pa and Poisson's ratio is 0.3. The coefficient of friction is set to 0.1. At initial configuration, the block overlaps the tube by 0.01 units. The prescribed axial displacement of the block top surface of 10 units is resolved with 100 equally spaced displacement increments. To analyse the effects of the mesh density, the tube is discretised with two meshes, the coarse mesh has 780 triangles, while the fine mesh has 2,441 triangles, see Fig. 4.42. The block is discretised with 1,000 CVs and the mesh is kept the same for both cases.

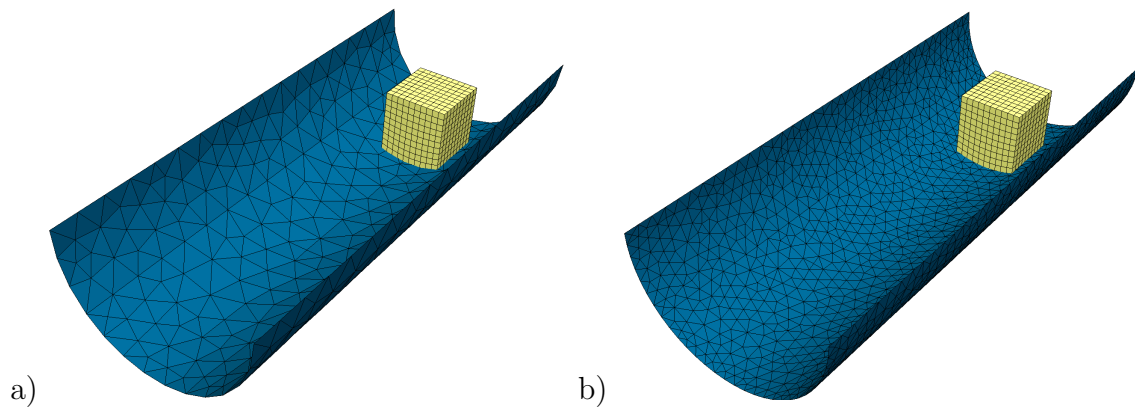a)   b)

Figure 4.42. Block sliding in a half-tube: a) coarse mesh, b) fine mesh

Figure 4.43 shows the evolution of the horizontal reaction force for three different types of contact algorithms on both meshes. The contact algorithm with the Nagata smoothing has a negligible difference between the results for the coarse and the fine mesh in contrast to the pointwise and segment contact algorithms in which

poor approximation of the cylinder geometry results in a difference between the computed force. Also, the poor approximation of the cylinder surface leads to oscillations of the reaction force which are reduced with mesh refinement. With the finer mesh, cylinder geometry is better described, thus these algorithms converge to the results obtained with the surface smoothing. On both meshes, the application of the segment contact algorithm results in higher accuracy as the force tends to be closer to the force obtained using the smoothing.

In [74] the same example is analysed using the linear finite elements with and without surface smoothing using Nagata interpolation. The authors use two mesh densities for the tube: 486 and 3880 triangles. The block is discretised identically as here. Reported results obtained using finer tube discretisation (with and without smoothing) are added in Fig. 4.43. One can see that an almost identical deviation is captured between results with and without tube surface smoothing. It should be noted that their deviation is slightly lower due to finer tube discretisation. Most important, results obtained using the smoothing procedure are in great agreement with the results obtained herein.
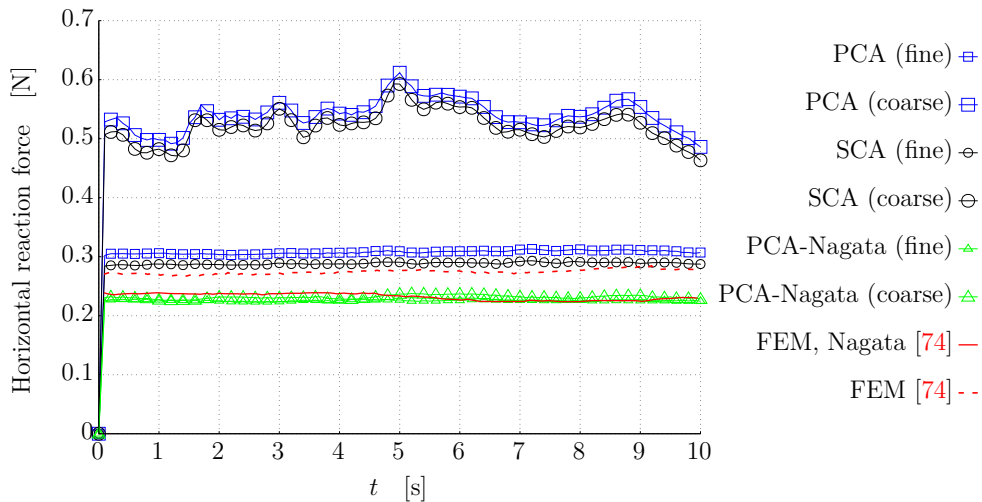

Figure 4.43. Block sliding in a half-tube: evolution of horizontal reaction force

Table 4.12 compares the computational efficiency between the contact algorithms in terms of the average number of outer iterations and computational time ratio. As depicted, the usage of surface smoothing has a positive side-effect on efficiency, especially if the coarse mesh is used. These differences tend to reduce with mesh refinement; however, the application of surface smoothing can provide very accurate results with coarse discretisation which can be extremely beneficial if the accuracy of pressure distribution on the smoothed surface is of no big interest. Moreover, for

large scale problems, the usage of a coarse mesh can significantly reduce the cost of contact detection.

Table 4.12. Block sliding in a half-tube: average number of outer correctors $\bar{n}_{corr}$ and CPU time ratio

	Coarse mesh		Fine mesh	
	$\bar{n}_{corr}$	$\dfrac{\text{CPU time}}{\text{CPU time (Nagata)}}$	$\bar{n}_{corr}$	$\dfrac{\text{CPU time}}{\text{CPU time (Nagata)}}$
SCA	430	1.5	390	1.2
PCA	420	1.4	340	1.1
PCA-Nagata	320	1	340	1

4.7. Summary

In this chapter, the numerical treatment of large sliding frictional contact problems within the finite volume framework was discussed. The penalty based Neumann-Neumann coupling procedure is considered and various aspects of this procedure are given in detail. A novel segment-to-segment contact-force calculation algorithm is introduced along with a suitable contact detection algorithm. Including a few examples, the main features of the proposed algorithm are highlighted and compared with the available pointwise contact algorithm, which is currently the only algorithm capable of solving the considered class of contact problems. In order to improve efficiency, the implicit version of the Neumann-Neumann coupling procedure is proposed by deriving an implicit equation for the boundary gradient. It is shown that the implicit contact coupling is a promising approach to achieve better convergence of the coupling procedure. In addition, surface smoothing using the Nagata surface interpolation is incorporated within the pointwise contact algorithm to show potential benefits of such a procedure.

5. Numerical examples

5.1. Introduction

This chapter presents various numerical examples to demonstrate the accuracy and robustness of the newly developed contact algorithm. The examples are presented in three sections. In the first section, contact problems with available analytical solutions are considered and the analytical solutions given in Appendix A are used to examine the accuracy. The analytical solutions are derived under the assumption of infinitesimal strains; however, the compressible neo-Hookean constitute relation is used as it is consistent with linear elasticity. The second section is devoted to contact problems with finite deformations and large relative sliding of bodies in contact. The chosen benchmark examples, extracted from the literature on the computational contact mechanics, are specially designed to challenge the robustness of the contact treatment. For example, it is reported that the FEM contact algorithm with the standard implementation of the NTS discretisation exhibits premature failure in some of the considered examples. The obtained solutions in this section are compared with the reported solutions calculated by using the FEM. Finally, in the last section, the application on industrial grade contact problems is shown by solving different metal forming problems: wire drawing, wire rolling and wire compacting. The application of the numerical method in the solving of such problems is of crucial importance for engineering practice since it allows for fast optimisation of tool and workpiece geometry as well as other parameters such as homogenisation of the residual stress and damage reduction.

To allow for reproducibility, for each benchmark problem, a suitable description is given regarding boundary conditions and the computational model. Additionally, the considered examples are summarised in Table 5.1, and the literature in which they were used is listed.

In all examples gravity is neglected and the chosen penalty factor is held constant for all time steps. If density is not mentioned within the problem description, the problem is solved as quasi-static, thus time step refers to loading increment. The relative residual is considered as the convergence criteria and its value is set to $1 \cdot 10^{-6}$. For the purpose of reducing the computational time, in some examples

a geometrically simplified computational model is used if the example permits the usage of the symmetry boundary condition. The simulations were executed on a workstation machine with 32 CPU cores (Intel Xeon Gold 6226R 2.90GHz) with the available random-access memory of 384 GB.

From the aspects of the contact algorithm, the problem is undoubtedly more challenging in the case of coarse discretisation at the contact interface. In addition, for solving large scale problems, it is commonly inconvenient to ensure fine mesh resolution at contact interfaces. Accordingly, all examples are tested on relatively coarse grids to check if the contact algorithm is robust enough and capable of providing quality results. Further, to check the impact of the grid resolution on the results of interest, some examples are solved on a set of uniformly refined grids. It should be noted that the coarse grids are chosen to be similar or the same by density as the grids used in the literature for FEM calculations with linear finite elements.

Table 5.1. Benchmarks problems dealt with in this thesis. The first source, separated from the rest by a comma, denotes the paper in which the example was proposed initially

5.2.1. Cylinder and Flat Rigid Surface	[84], [149, 150]
5.2.2. Cylinders With Parallel Axis	[84], [64, 151]
5.2.4. Cylindrical Punch With Rounded Edge	[9], [57, 58, 152]
5.3.1. Cylinder Drawn Through a Die	[153]
5.3.2. Shallow Ironing	[154], [87, 88, 155, 156]
5.3.3. Contact Between Curved Beams	[84], [10, 157, 158]
5.3.4. Compressed Concentric Spheres	[71], [158, 159]
5.3.5. Twisting Contact Between a Hemisphere and a Block	[160], [159]

5.2. Small Deformation Benchmark Problems

5.2.1. Cylinder and Flat Rigid Surface

The contact between an infinite half-cylinder and a rigid plane is a classical example of the Hertzian contact commonly used in literature to check the accuracy of the numerical procedures. To satisfy the Hertzian assumption of infinitesimal strains the prescribed displacement or load at the half-cylinder top surface must be very small. The analytical solution to this problem is given in Appendix A1.

The problem is solved using a 2D computational model (see Fig. 5.1) and plane strain assumptions. The cylinder geometry and material properties are chosen in accordance with literature: Young's modulus $E = 200$ Pa, Poisson's ratio $\nu = 0.3$ and radius $R = 8$ m. The top surface of the half-cylinder has prescribed vertical displacement $u_z = 0.02$ m. Two computational meshes are used: a coarse mesh with 1,615 CVs and a fine mesh with 6,460 CVs (fine mesh is obtained by halving each edge of the coarse mesh).
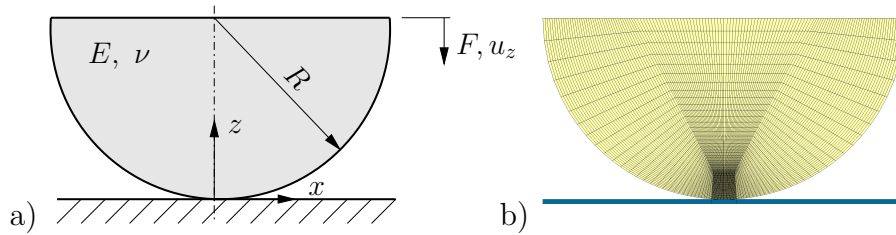

Figure 5.1. Contact between cylinder and flat rigid surface: a) problem geometry, b) fine computational mesh (6,460 CVs)

Figures 5.2 and 5.3 show the pressure distribution at the contact surface and stress components along the z-axis for the coarse and the fine mesh. The diagrams are given in the normalised form in which stress is normalised with the Herzian peak contact pressure and the x-coordinate with the Hertzian contact width. The distribution of the contact pressure is symmetric; therefore, to improve clearness, only the results for $x > 0$ are presented. As shown, for both meshes the peak contact pressure, the contact width and the distribution of stresses match well to the analytical solution. The coarse mesh has slight deviations from the analytical solution as regards the pressure values near the contact edge; the reason for this is the sharp slope of the pressure distribution which is hard to capture with the piecewise linear approximation of the solution field.
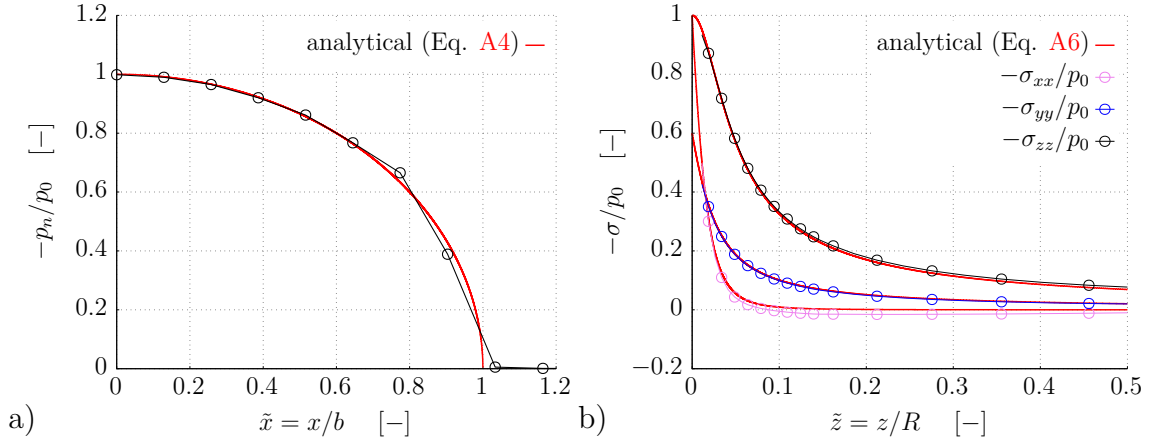
Figure 5.2. Contact between cylinder and flat rigid surface (coarse mesh): a) normalised contact pressure distribution, b) normalised stress distribution along z-axis
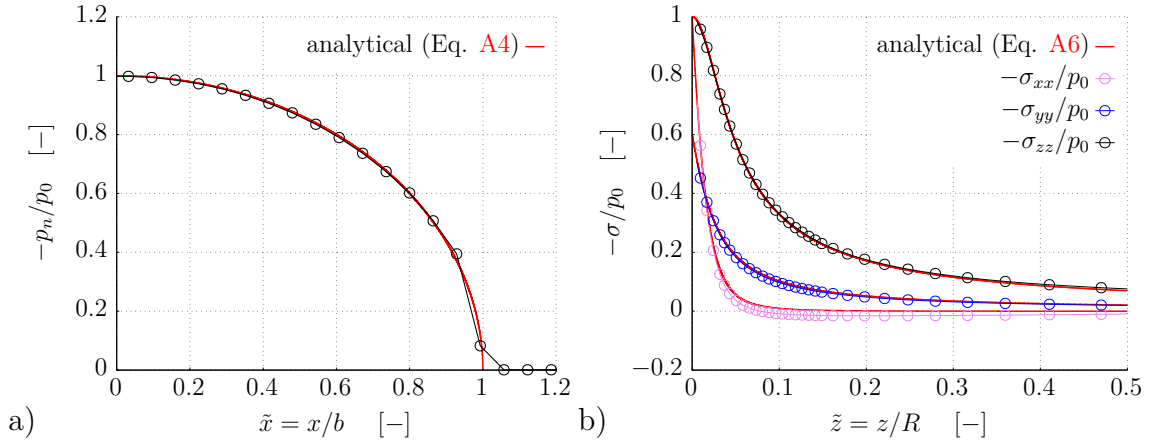

Figure 5.3. Contact between cylinder and flat rigid surface (fine mesh): a) normalised contact pressure distribution, b) normalised stress distribution along z-axis

5.2.2. Cylinders With Parallel Axis

This example considers the contact between two identical infinite half-cylinders with parallel axes. The geometry and material properties of both cylinders are identical as in the previous example ($E = 200$ Pa, $\nu = 0.3$ and $R = 8$ m). Precisely, the example is an extension to the previous example: the rigid surface is replaced with a deformable cylinder. The bottom surface of the lower cylinder is held constant, whereas the top surface of the upper cylinder has a prescribed constant normal and tangential pressure distribution, see Fig 5.4 a). The problem is solved with two load increments using a 2D computational model under the plane strain assumption. In the first load increment, only the normal pressure is prescribed ($p = 0.625$ Pa), and it is held constant for the second increment, in which the tangential pressure (q

= 0.05851 Pa) is subsequently added. Friction is considered and the coefficient of friction is set to $\mu = 0.2$.

Two computational meshes are used: a coarse and a fine mesh, which is obtained by the halving each edge of the coarse mesh. In the coarse mesh, shown in Fig. 5.4 b), the lower cylinder has 12 CVs at the contact zone, while the upper cylinder has 18 CVs. The overall number of CVs in the lower cylinder is 1,120 and in the upper is 2,520 CVs. The non-conformal discretisation was intentionally made to check for possible errors which can arise due to different cylinder curvature discretisation.
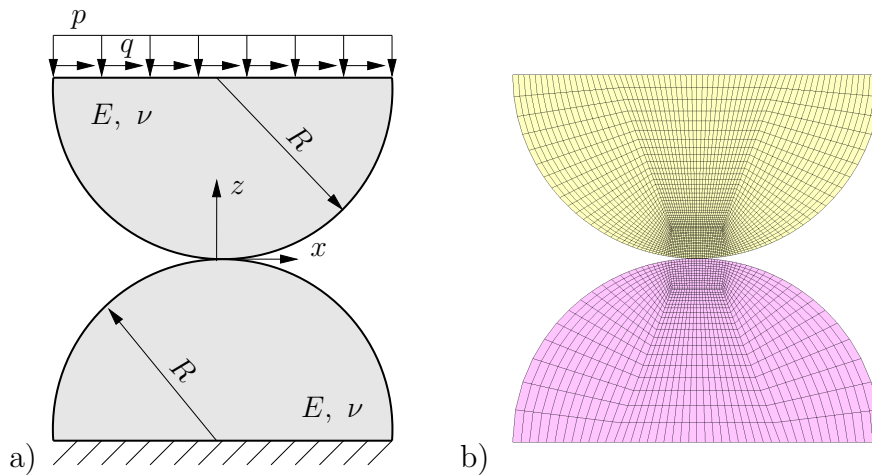

Figure 5.4. Cylinders with parallel axis: a) problem geometry, b) coarse computational mesh (3,640 CVs)

Figure 5.5 shows a comparison of the normal and the tangential contact pressure with the analytical solution; the analytical solution is given in Appendix A1. The pressure distribution is symmetric, thus the diagram gives a distribution for both cylinders: the left-hand side of the diagram shows the pressure distribution for the lower (coarser) cylinder, while the right-hand side shows the distribution for the upper (finer) cylinder. The slip and stick zones of the tangential traction are correctly captured on both meshes and, as expected, the results for the fine mesh agree better with the analytical solution. Nevertheless, even the coarse mesh can produce satisfactory results; the maximum difference to the analytical solution can be seen at the edge of the contact area where the contact width is slightly overpredicted and the slope of the normal contact pressure is high.
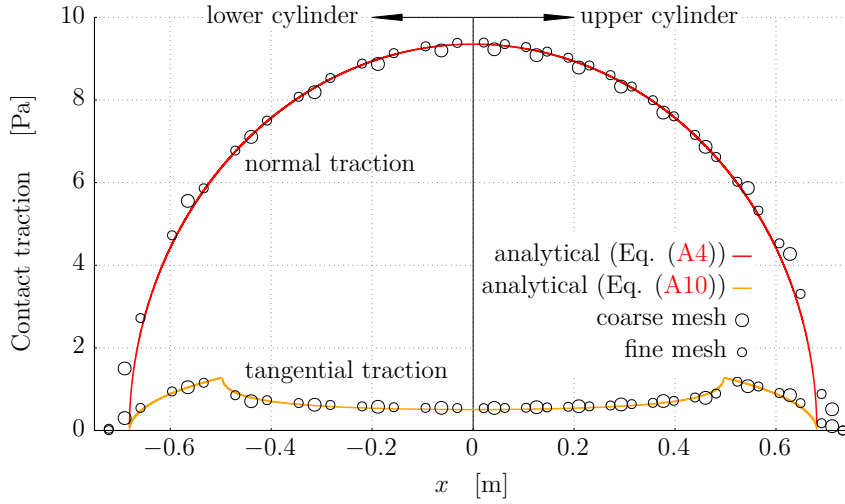
Figure 5.5. Cylinders with parallel axis: normal and tangential pressure distribution at the lower and upper cylinder for coarse and fine mesh

5.2.3. Indentation of Elastic Half-Space With a Flat-Ended Rigid Punch

A rigid flat-ended punch is pressed onto an elastic-half space, as shown in Fig. 5.6. The elastic half-space is modelled as finite, with dimensions $2H \times H$, with Young's modulus $E = 200$ GPa and Poisson's ratio $\nu = 0.3$. The problem is solved by assuming a plane stress state and a 2D model with unit thickness. The top surface of the rigid punch have the prescribed vertical displacement $u_z = 0.0967$ mm corresponding to vertical force $F_z = 10,000$ N. The bottom surface of the finite half-space is held fixed. The contact between the punch and the elastic half-space is modelled as frictionless.
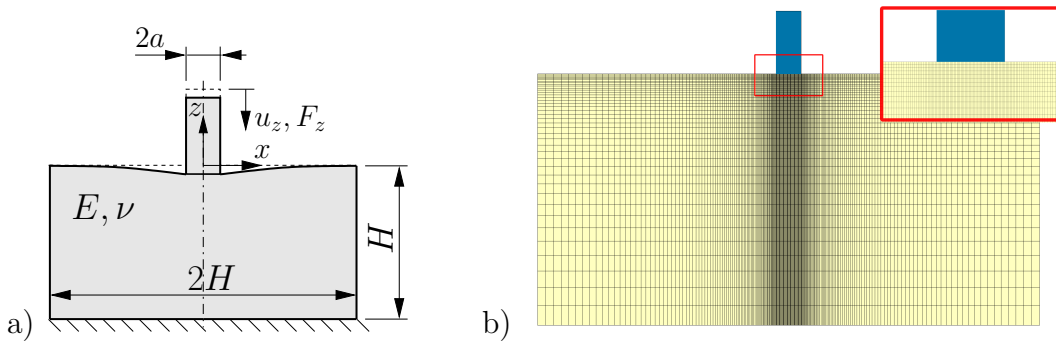

Figure 5.6. Indentation of rigid flat-ended punch: a) problem geometry ($H = 0.2$ m, $a = 0.01$ m), b) computational mesh (9,090 CVs)

Due to geometrical simplicity and the known contact area this example can be

characterised as a simple one; however, it should be noted that the indenter sharp edges cause singular pressure distribution (see the analytical solution presented in Appendix A3). Because of that, the refining of the mesh will always result in a higher pressure peak at the contact edge.

Figure 5.7 shows the distribution of the contact pressure and surface displacement. One can see that the obtained finite pressure distribution curve match well to the analytical solution. Further mesh refinement would lead to a better approximation of the analytical solution; however, it would be preferable and more physical to include the plasticity effects. The curve form of the surface displacement corresponds to the analytical solution; however, the finite pressure distribution leads to a slight offset between them. If the numerically obtained displacement distribution is translated for the average value of the achieved penetration (dotted red line in Fig. 5.7), a better agreement is obtained.
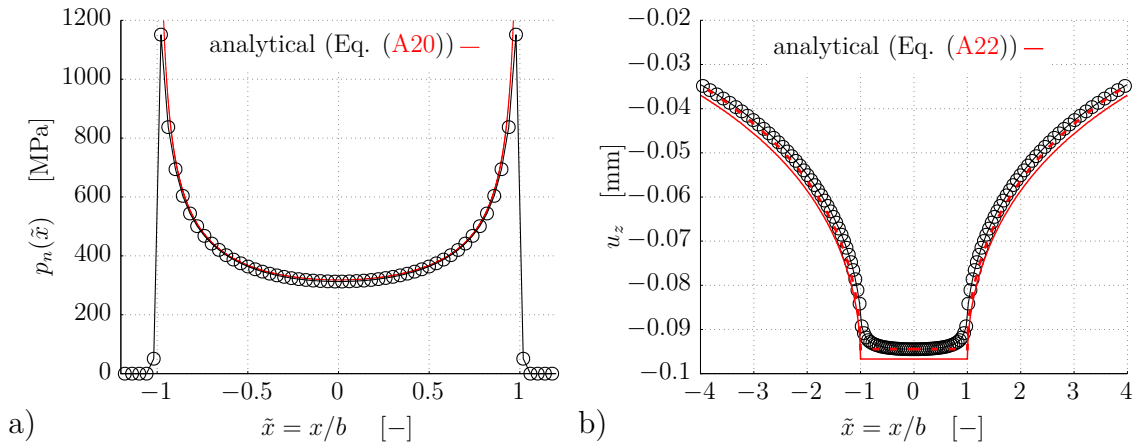

Figure 5.7. Indentation of elastic half-space with flat-ended rigid punch: a) contact pressure distribution, b) vertical displacement of contact surface

5.2.4. Cylindrical Punch With Rounded Edge

In this example, proposed by NAFEMS, a cylindrical punch is pressed into an elastic cylindrical foundation, see Fig. 5.8 a). The material properties of the punch are similar to steel ($E_p = 210$ GPa, $\nu_p = 0.3$), whereas the foundation has aluminium Young's modulus ($E_f = 70$ GPa, $\nu_f = 0.3$). The punch is loaded with uniform pressure distribution at its top surface. The bottom surface of the elastic foundation is held fixed. Friction is considered and the coefficient of friction is set to $\mu = 0.1$. The problem is solved with one load-increment using an axisymmetric computational model. The computational mesh consisting of 3,660 CVs (punch 860 CVs, foundation
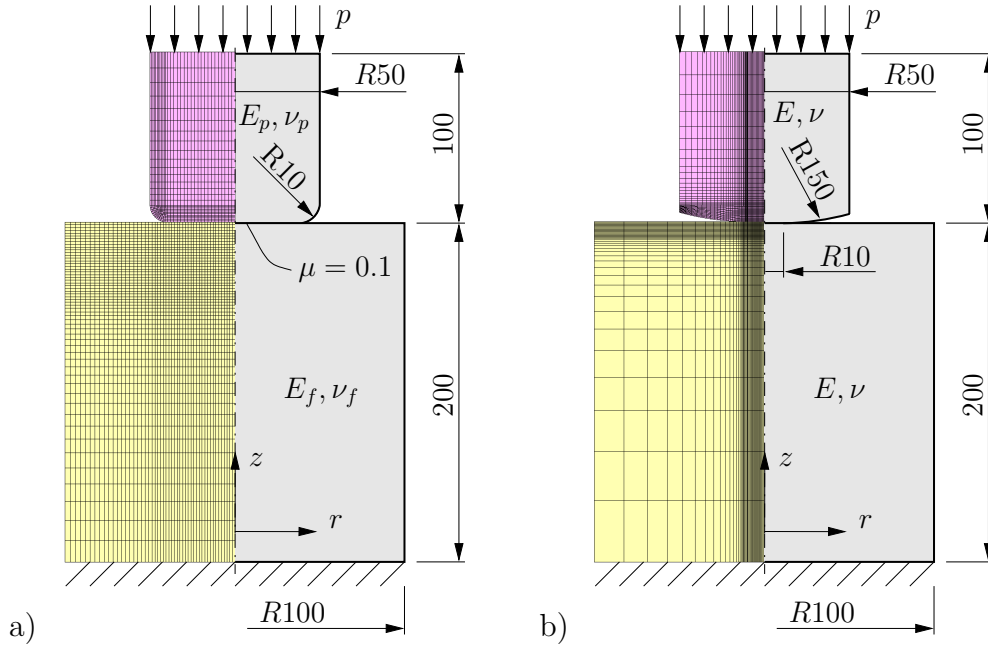
2,800 CVs) is shown in Fig. 5.8 a).

Figure 5.8. Cylindrical punch with rounded edge: a) case proposed by [9], b) modified case. Used computational mesh is included within left-hand side of each figure

The predicted radial and axial displacement distributions on the foundation contact surface are shown in Fig. 5.9 a) and b), respectively. The results are compared with the results from [9], obtained using linear and quadratic finite elements in the MSC Marc software. One can see that the obtained distribution of both displacement fields matches well to the results obtained using quadratic finite elements.
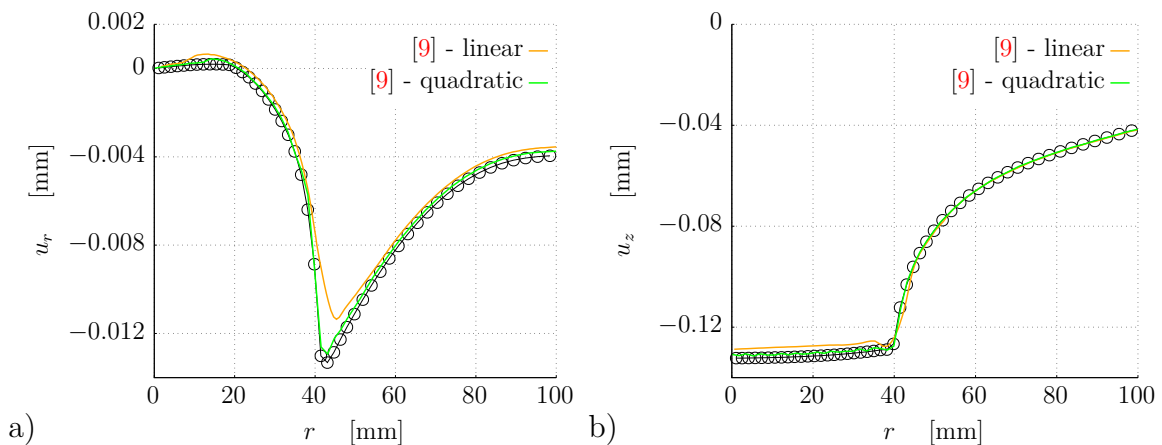

Figure 5.9. Cylindrical punch with rounded edge: a) foundation radial displacement, b) foundation axial displacement

In Fig. 5.10 the normal and tangential contact pressure distributions are com-

pared with the results reported in [9]. One can see that a higher stress peak is predicted at the end of the contact surface. The reason for this lies in the insufficient curvature of the punch fillet. In other words, the radius of the punch flat part is very close to the actual radius of the contact area. In such cases, the stress distribution is singular like in the case of the flat-ended punch [161, 162]. Accordingly, the mesh independent results cannot be achieved and the obtained difference is solely caused by the fact that a finer computational mesh is used.
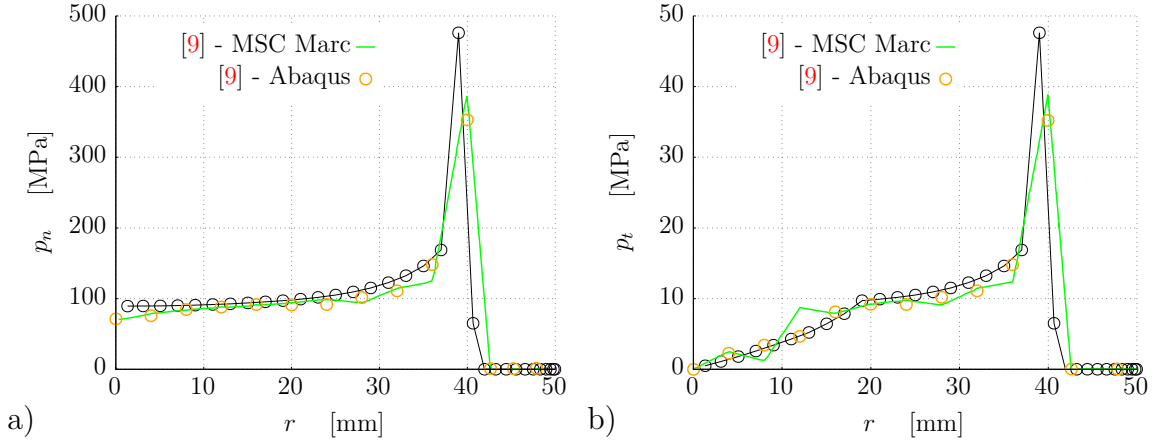

Figure 5.10. Cylindrical punch with rounded edge: a) normal contact pressure distribution, b) tangential contact pressure distribution

To ensure the finite pressure distribution, we propose to introduce the modification shown in Fig. 5.8 b). The fillet radius is much larger and the punch flat section is reduced from 40 to 10 mm. In the case of the same material for both bodies, the analytical solution from [162, 163] is available to verify the obtained distribution of the normal contact pressure. The targeted contact half-width is 12.5 mm, thus the prescribed pressure $p = 37.588$ MPa is calculated by using the analytical expression, see Eq. (A3.1.) in Appendix A3. Frictionless contact is considered and the material properties of both bodies are set to $E = 70$ GPa and $\nu = 0.3$. In the case of friction modelling, the stress distribution would remain the same as for the similar materials there is no relative sliding between bodies [162]. This can be confirmed by analysing Fig 5.11 b) in which one can see that within the contact area the radial displacement of the foundation and the punch match. Nonetheless, the proposed case can be easily modified to consider friction by using different material properties of punch and foundation, respectively.
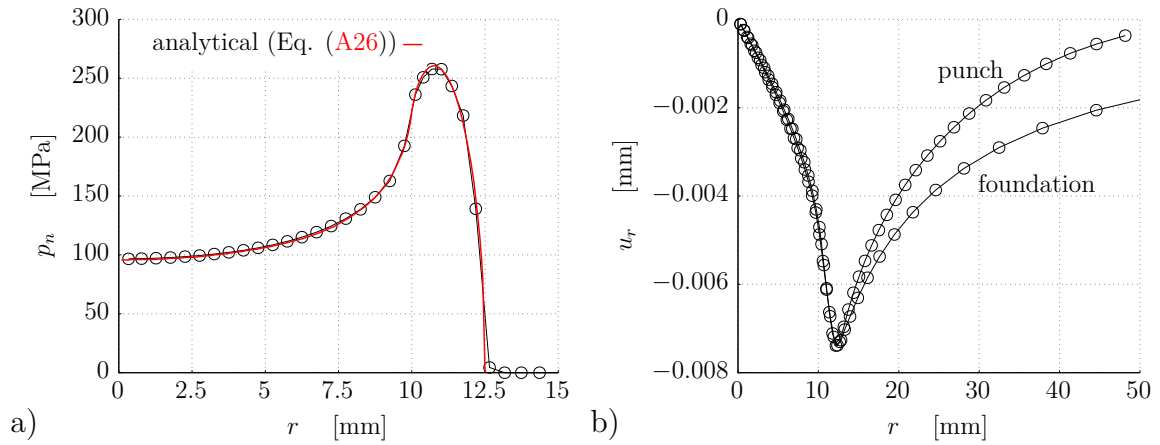
Figure 5.11. Cylindrical punch with rounded edge: a) normal contact pressure distribution, b) radial displacement distribution of punch and foundation contact surface

5.2.5. Contact Between Sphere and Block

This example considers the contact between a half-sphere and a deformable block, as shown in Fig 5.12 a). The predicted interior stress and the contact width are compared with the Hertzian analytical solution given in Appendix A2. The top surface of the half-sphere has prescribed vertical displacement $u_z = 0.01$ m, while the bottom surface of the block is held fixed. The material properties of the half-sphere and the block are $E_s/E_b = 3$ and $\nu_b/\nu_s = 1.1$. The problem is solved in one load increment using a half symmetry computational model (note that the model can be further reduced by using the quarter symmetry model). Two computational meshes are used. In the coarse mesh, shown in Fig. 5.12 b), the sphere is discretized with 23,232 CVs and the block with 13,552 CVs. The fine mesh is obtained by uniform refinement of the coarse mesh and consists of 294,282 CVs.
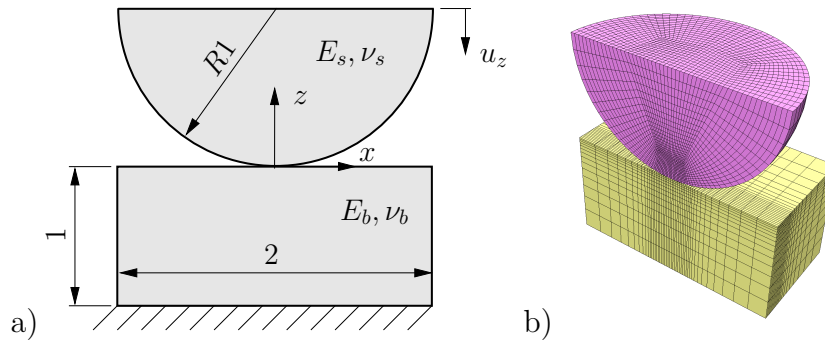

Figure 5.12. Contact between sphere and block: a) problem geometry, b) coarse computational mesh

113

5. Numerical examples

Figure 5.13 a) shows the predicted distribution of half-sphere subsurface stresses along the normalised z-axis (the stress components are normalised with the Hertzian peak pressure p_0). The scenario in which the bottom block is rigid is also considered and shown in Fig 5.13 b). One can see that, for both scenarios, the distribution of the stress field match well to the analytical solution.

Fig 5.14 shows, for both meshes, the distribution of the normalised normal contact pressure at the sphere contact surface; note the white circle representing the contact width obtained analytically. The expected axisymmetric pressure distribution is correctly captured, i.e. the obtained pressure contours are circularly shaped and they are smooth near the contact edge. Although the coarse mesh describes the contact half-width with only 7 uniformly spaced CVs, one can see that the contact width is not visibly overestimated and that the pressure contours match to the pressure contours of the fine mesh.
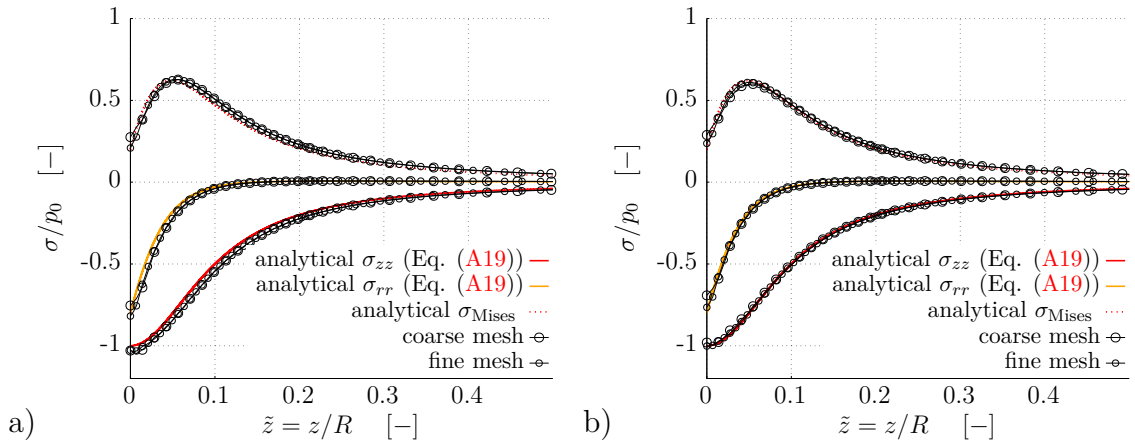

Figure 5.13. Contact between sphere and block: stress distribution along z-axis for: a) deformable block and b) rigid block
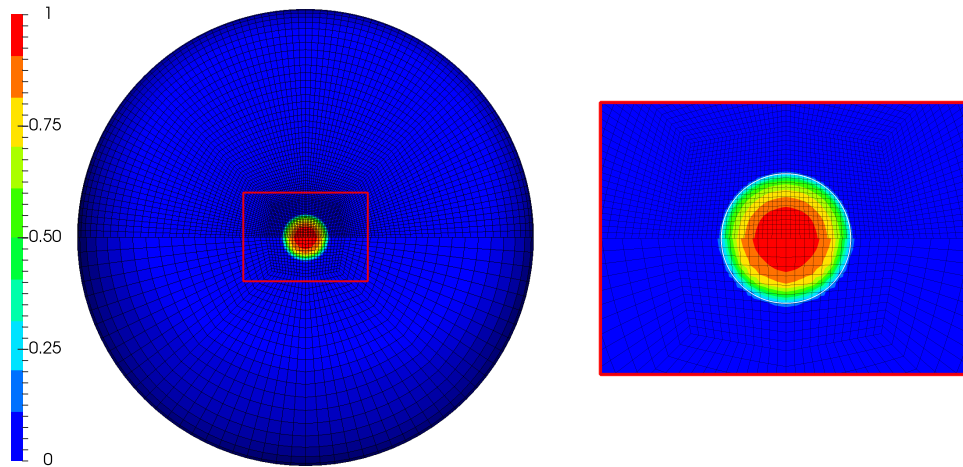
Figure 5.14. Contact between sphere and block: distribution of normalised normal contact pressure $|p_n/p_0|$ at half-sphere contact surface. Upper part of the image shows result from fine mesh and lower part from coarse mesh

5.3. Large Deformation Benchmark Problems

5.3.1. Cylinder Drawn Through a Die

An elastic cylinder is pulled through a rigid die, as shown in Fig. 5.15 a). Due to the axial symmetry of the problem, an axisymmetric model is used. At the left side of the wire mesh, the prescribed displacement of 0.25 m is applied within 100 equal displacement increments. The computed drawing force is compared with the analytical solution for the steady state drawing force [153]:

$$F = \frac{1}{6}\pi E r_2^2 \left[2 \left(\frac{r_2}{r_1} \right)^2 + \left(\frac{r_1}{r_2} \right)^4 - 3 \right]. \tag{5.1}$$

The available analytical solution is valid for an incompressible material; therefore, Poisson's ratio is set to 0.495. To better mimic incompressibility, Poisson's ratio should be higher; however, in that case, unphysical oscillations of force can be expected [164]. The Young modulus is set to 1 MPa. The value of the steady state drawing force shown in Fig 5.15 b) overpredicts the value obtained using the analytical solution by 5%. This difference can be considered to be acceptable and for further investigation, the pressure-displacement formulation would be a more suitable approach (see Section 3.3.2).
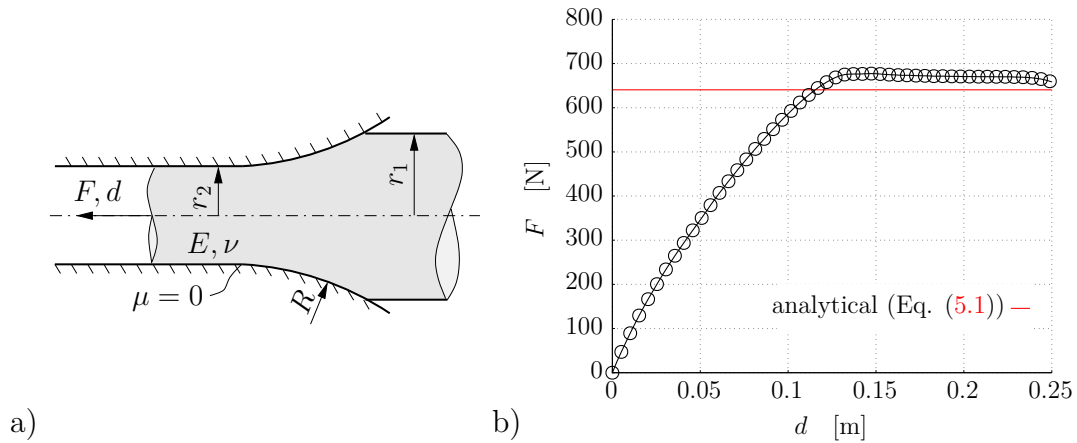
Figure 5.15. Cylinder drawn through a die: a) problem geometry ($r_1 = 0.025$ m, $r_2 = 0.015$ m, $R = 0.15$ m), b) evolution of pulling force

Figure 5.16 shows deformed configurations of the cylinder for different wire displacements. The initial length of the cylinder is 0.07 m. Cylinder is discretised with 20 CVs in axial direction and 3 CVs in radial direction. Although not shown in Fig. 5.16, the cylinder mesh is extended axially using 17 CVs to the die flat part following its curvature. This is done to help draw the cylinder inside the die at the beginning of the simulation. Since this example seems hard to solve, a necessary remedy is to heavily under-relax the displacement field with a low under-relaxation factor (0.7).
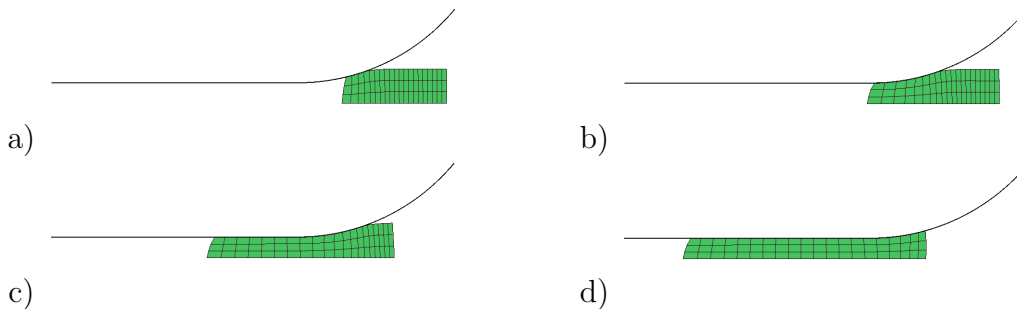
Figure 5.16. Cylinder drawn through a die: deformed configurations of cylinder at: a) $d = 0.0625$ m, b) $d = 0.125$ m, c) $d = 0.1875$ m and d) $d = 0.25$ m

5.3.2. Shallow Ironing

A pressed elastic block with a rounded contact surface slides over an elastic rectangular foundation. The foundation is fixed at the bottom and the block has a prescribed vertical and horizontal displacement. The problem geometry, material data and dimensions are shown in Fig. 5.17. A vertical displacement of the top

surface of the block ($u_y = 1$ mm) is applied from 0 to 1 s and solved using 100 equal displacement increments. A horizontal displacement ($u_x = 10$ mm) is subsequently added between 1 and 2 s and solved using 500 equal displacement increments. The problem is solved under the plane strain assumption. At the contact interface, friction is considered, and the coefficient of friction is set to $\mu = 0.3$.
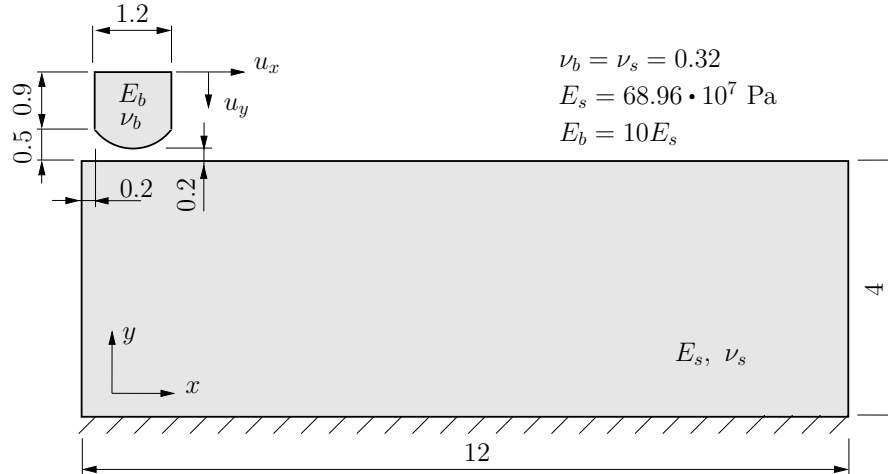
Figure 5.17. Shallow ironing: geometry description and material properties (dimensions in mm) [7]

The calculation is conducted on the mesh in which the block is discretised using 72 CVs and slab using 3,600 CVs (the same as in [88] where linear finite elements are used). The initial mesh configuration and the deformed mesh configuration for different time steps are shown in Figure 5.18.

Figure 5.18. Shallow ironing: deformed mesh configurations at: a) initial configuration, b) $t = 1$ s, c) $t = 1.5$ s and d) $t = 2$ s

5. Numerical examples

The evolution of the vertical and horizontal reaction forces is shown in Fig. 5.19 together with the results from the literature. One can see that there is no unique agreement among results obtained using FEM. Except for contact modelling, the cause for these discrepancies in results can also lie in other aspects of the numerical procedure (for ex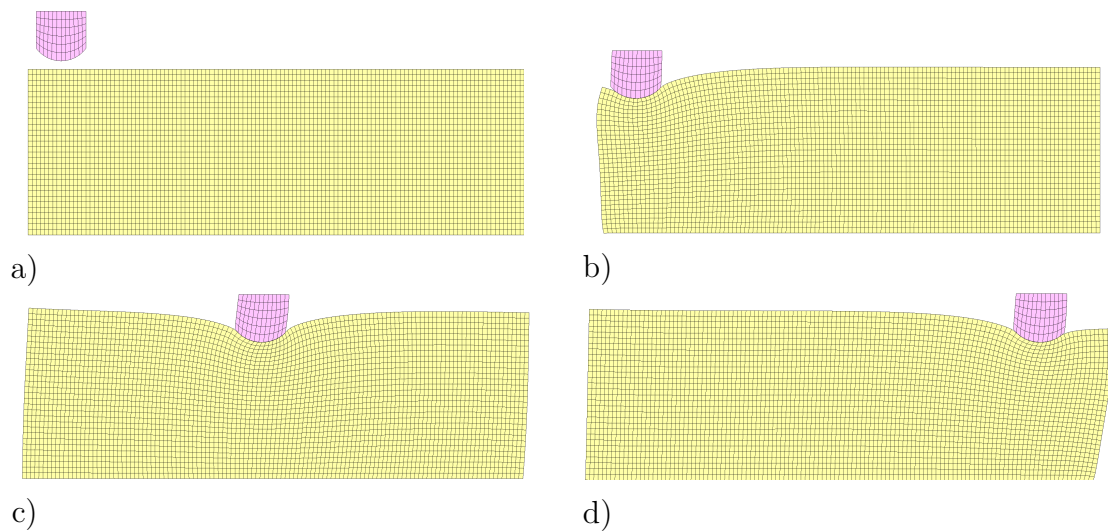ample, applied constitutive relation). Regardless, it is important to note that the evolution of the contact force is smooth during sliding, i.e. there are no visible oscillations. Also, the form of the obtained curves matches well with the results from [156] and [155].
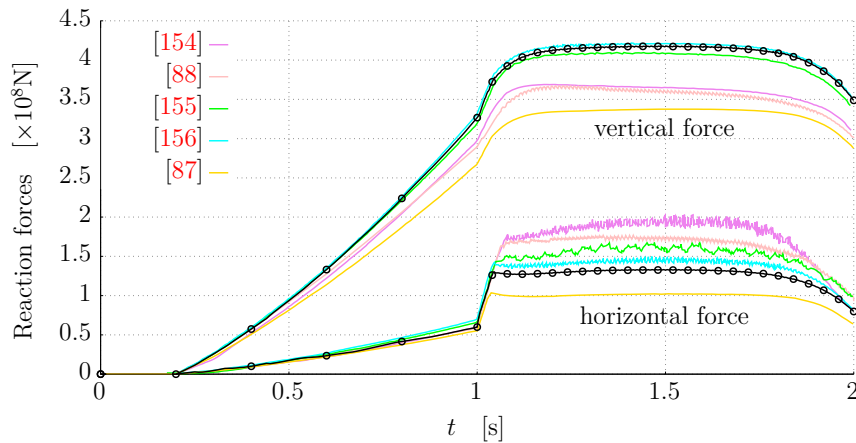

Figure 5.19. Shallow ironing: evolution of vertical and horizontal reaction force

Another parameter to consider, apart from the values of the reaction forces, is the ratio between reaction forces, referred to as the global coefficient of friction. Table 5.2 summarises the values of the global coefficient of friction for the results shown in Fig. 5.19. One can see that the obtained result for the global coefficient of friction is close to the local coefficient of friction and that the results for the global coefficient of friction significantly vary in the literature. In [87] it is reported that there is no notable difference between the local and the global coefficient of friction when the commercial software Ansys is used, which is in agreement with the obtained solution. Generally, the results are closest to the results reported in the documentation of the open-source software Code_Aster [156].

Table 5.2. Shallow ironing: comparison of global coefficients of friction μ_g

	Present	[87]	[156]	[155]	[88]	[154]
$(\mu_g)_{t=1.5}$	0.32	0.30	0.34	0.38	0.47	0.53

5.3.3. Contact Between Curved Beams

This example considers large sliding between two curved beams, see Fig. 5.20 a). The lower beam is fixed, whereas the upper beam has a prescribed horizontal displacement of $u_x = 31.5$ mm. Both beams are modelled as elastoplastic with isotropic hardening using the following material properties: Young's modulus $E = 689.56$ MPa, Poisson's ration $\nu = 0.32$, initial yield strength $\sigma_Y = 31$ MPa and linear hardening coefficient $H^` = 261.2$ MPa. The problem is solved using 63 equal displacement increments, assuming the plane strain conditions. The frictionless and frictional responses are considered. As regards the frictional response, two coefficients of friction are taken $\mu = 0.3$ and $\mu = 0.6$. Figure 5.20 b) shows the computational mesh at initial configuration: the beams are discretised using 50 CVs in the circumferential and 5 CVs in the radial direction.
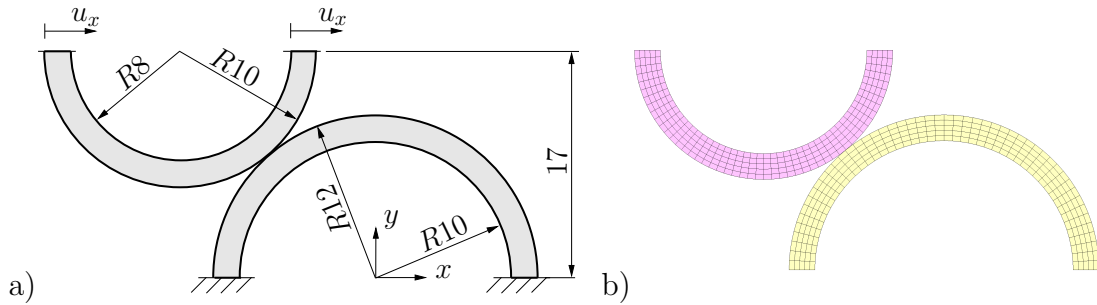
Figure 5.20. Contact between curved beams: a) problem geometry (dimensions in mm) and b) mesh at initial configuration

Figures 5.21 and 5.22 show the evolution of the horizontal and vertical total reaction forces for the considered cases with different choices of the master surface. In all cases, results are insensitive to the choice of the master surface and the form of the curves matches well to the results reported in the literature [10, 158]. Compared to [10], a relative difference in the maximum amplitude of the reaction forces is below 3% for both components of the reaction forces. More importantly, during the sliding, the evolution of the reaction forces is smooth and without visible oscillations even though the discretisation of the contact surface is piecewise linear.

5. Numerical examples
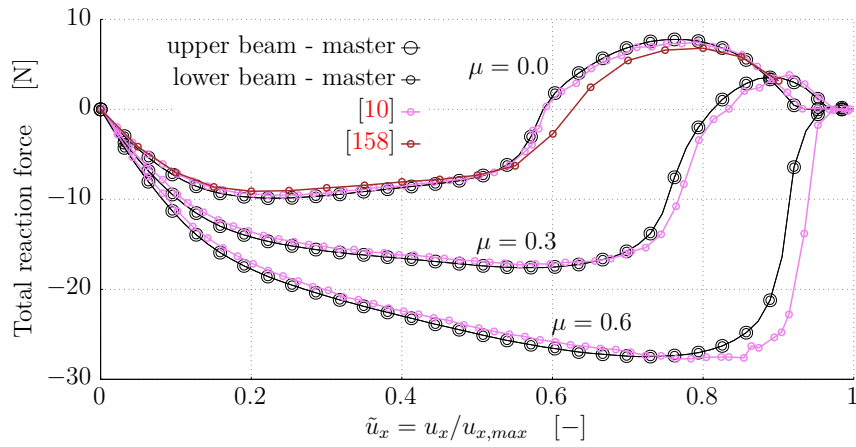
Figure 5.21. Contact between curved beams: total reaction force in x-direction
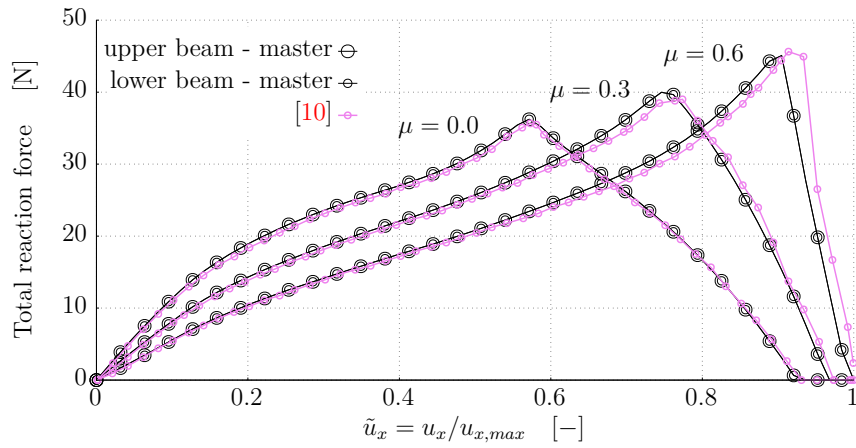
Figure 5.22. Contact between curved beams: total reaction force in y-direction

During the sliding, due to the larger diameter, the lower beam exhibits larger plastic deformation which is reduced by the larger values of the coefficient of friction. This can be seen in Fig. 5.23, where contours of the equivalent plastic strain are shown for the upper beam displacement of $u_x = 15$ mm. The predicted distribution of the equivalent plastic strain is compared side-by-side to the results from [10], which are added to the right-hand side of the Fig. 5.23. One can see that, in all cases, the locations and the magnitude of the plastic deformations agree well.
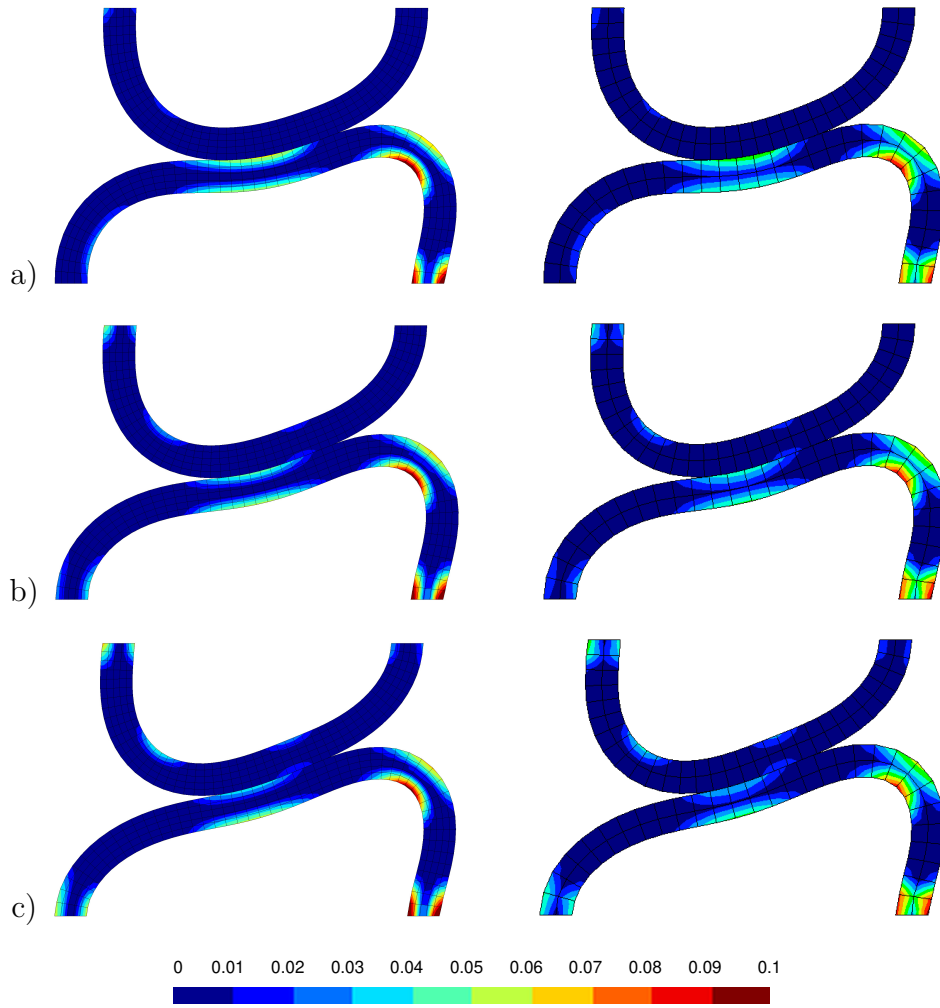
Figure 5.23. Contact between curved beams: contours of equivalent plastic strain $\varepsilon^{p,eq}$ at $u_x = 15$ mm for: a) $\mu = 0$, b) $\mu = 0.3$ and c) $\mu = 0.6$. Solutions from [10] are added to right-hand side

5.3.4. Compressed Concentric Spheres

In this example, two thick-walled hollow concentric half-spheres are compressed between two rigid planes. Due to the geometric and the load symmetry, the computational model is modelled in quarter symmetry. The geometry of the computational model is shown in Fig. 5.24 a). The contact between the lower rigid plane and the inner sphere as well as contact between the upper rigid plane and the outer sphere is modelled as frictionless. For the contact between two spheres, two different coefficients of friction are considered ($\mu = 0$ and $\mu = 0.5$). The displacement of the upper rigid plane $u_z = 9$ mm is prescribed within 90 equal displacement increments. The material property of both spheres is the same ($E = 1$ MPa and $\nu = 0.3$).

5. Numerical examples

The problem is solved using two computational meshes; the fine mesh (shown in Fig. 5.24 c)) and the coarse mesh (Fig 5.24 b)) which is similar by density to the mesh used in [71, 158, 159]. Due to the large compression and coarse discretisation of the curved contact interface, the coarse mesh is considered more challenging to solve from the aspects of the contact algorithm.
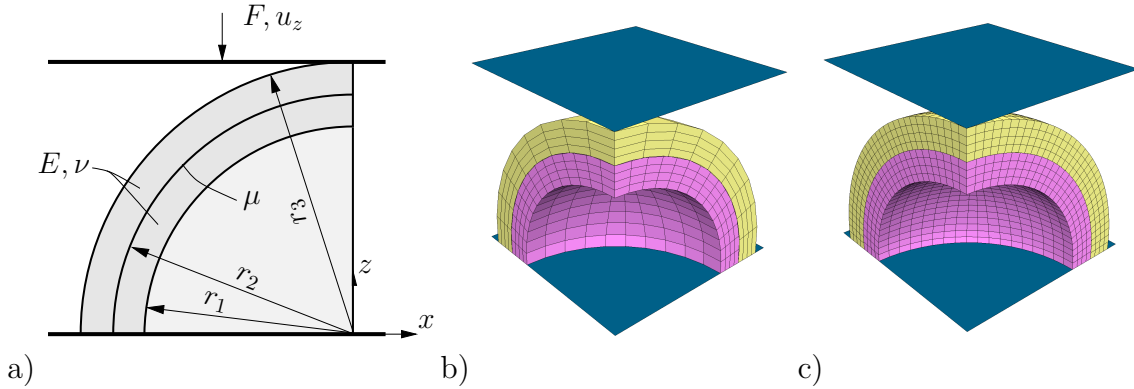

Figure 5.24. Compressed concentric spheres: a) problem geometry ($r_1 = 10$, $r_2 = 12$, $r_3 = 14$ mm), b) coarse mesh (inner sphere 500 CVs, outer sphere 180 CVs), c) fine mesh (inner sphere 2,420 CVs, outer sphere 1,620 CVs)

Figures 5.25 and 5.26 show the evolution of the compression force for the frictionless and the frictional case, respectively. For both scenarios, the compression force before buckling of the inner sphere matches well to the results from the literature. Also, there is no notable difference between the results obtained using a coarse and a fine mesh.

After the start of the inner sphere buckling, there is a visible difference between the evolution of the compression force, however, the reason for this lies in the buckling of the outer sphere. More precisely; when the coarse mesh is used, the outer sphere is less prone to buckling, as shown in Fig. 5.27. For the frictional case, the outer sphere starts to buckle; therefore, the results between two meshes match better. Nonetheless, these differences are not caused by the contact modelling, which can be confirmed comparing the results from the literature as some authors reported the occurrence of buckling of the outer sphere and some did not. After $u_z = 8$ mm the compression force shows some reduction in the force slope, which can be further investigated but in overall, the results on both meshes are in good agreement with the results from the literature.
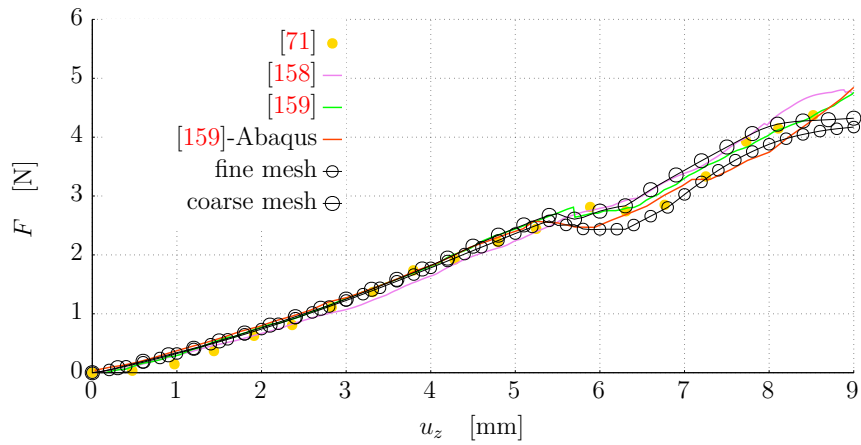
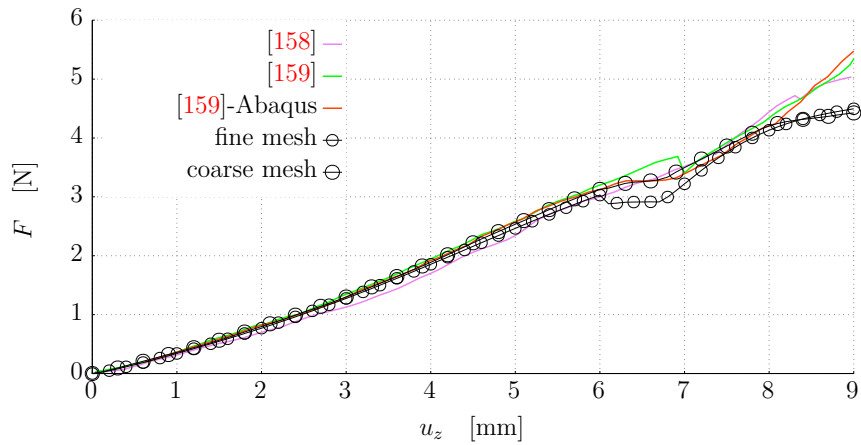Figure 5.25. Compressed concentric spheres: compression force for frictionless case

Figure 5.26. Compressed concentric spheres: compression force for frictional case ($\mu = 0.5$)
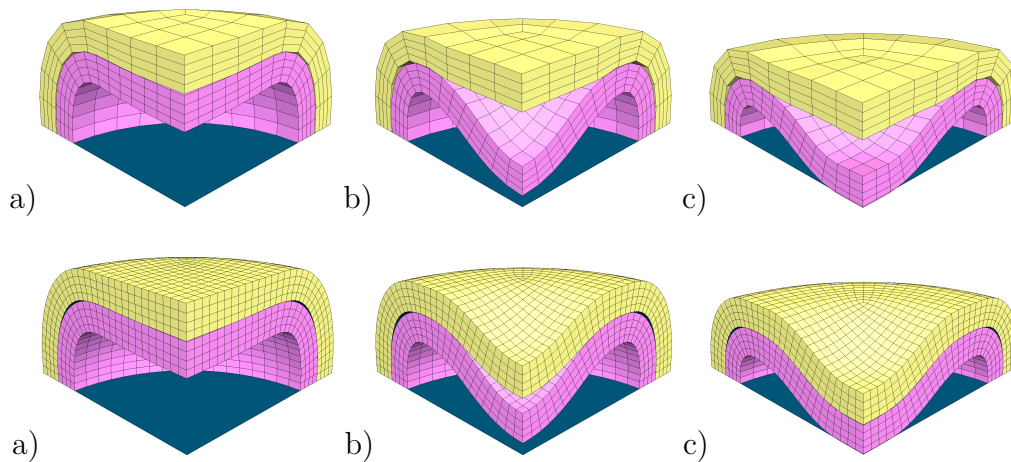
Figure 5.27. Compressed concentric spheres: deformed configurations of coarse and fine mesh for frictionless contact at: a) $u_z = 5$ mm, b) $u_z = 6$ mm, c) $u_z = 7.5$ mm

5.3.5. Twisting Contact Between a Hemisphere and a Block

A thick-walled hollow half-sphere is indented and twisted against a finite deformable block. The half-sphere downward motion is prescribed on its top surface, and it is equal by magnitude to sphere outer radius R_i. After indentation, rigid rotation of the half-sphere top surface is subsequently added. During the half-sphere indentation, frictionless contact is assumed, whereas during the twisting a high value of the coefficient of friction is considered ($\mu = 0.5$). The problem geometry as well as material properties are shown in Fig. 5.28 a).

Sphere indentation is prescribed within 10 equal displacement increments, whereas sphere twisting is prescribed within 90 equal rotation increments. The computational mesh at initial configuration is shown in Fig 5.28 b).
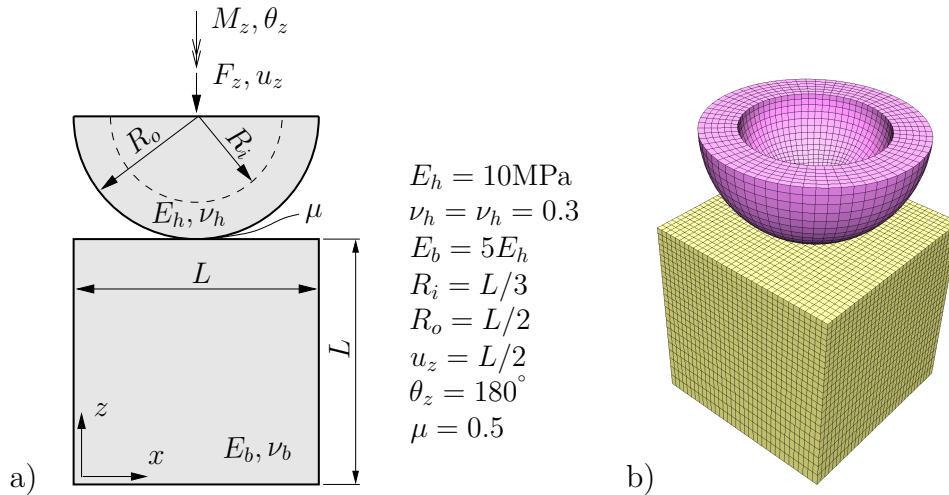
Figure 5.28. Twisting contact between a hemisphere and a block: a) problem geometry ($L = 2$ mm) and material properties, b) computational mesh (block 27,744 CVs, half-sphere 7,680 CVs)

The high value of the coefficient of friction enables stick contact behaviour, which induces twisting of the block, as shown in Fig 5.29. During the twisting, the contact area is gradually transformed to the dominantly slip behaviour, which means that constant torque is transmitted to the block. Consequently, the block exhibits constant deformation for larger half-sphere rotation angles.
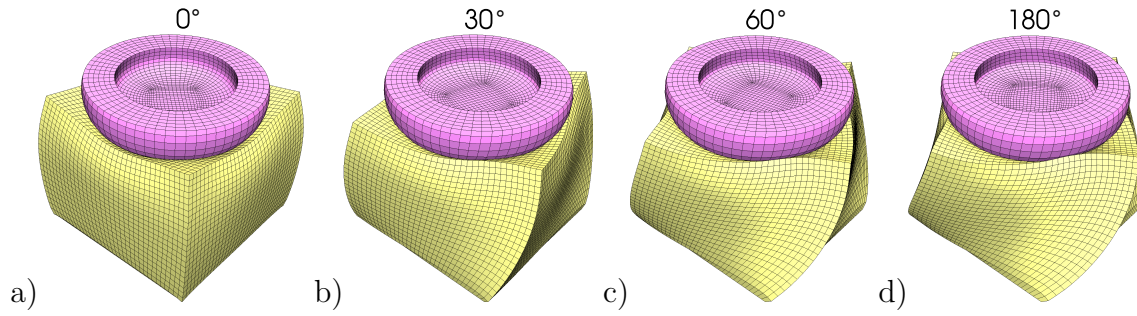
Figure 5.29. Twisting contact between a hemisphere and a block: deformed mesh configurations for twisting angles: a) $\theta = 0°$, b) $\theta = 30°$, c) $\theta = 60°$ and d) $\theta = 180°$

Figure 5.30 shows the evolution of the half-sphere twisting torque. During block deformation, i.e. for the rotation angles smaller than $\theta = 120°$, the torque value gradually increases until its constant evolution is reached. By comparing the obtained results with the results from the literature, one can see that the twisting torque is in good agreement during the transition from stick to slip behaviour. In the regime of the dominantly slip behaviour, the twisting torque is overpredicted by 5%. The reason for this lies in the 2.5% higher value of the normal contact force, as shown in Fig. 5.31. Although the obtained differences can be further investigated, from the aspects of contact modelling it is more important to notice that the form of the curves matches and that there are no visible oscillations in reaction force during large sliding. Figures 5.30 and 5.31 also include results for different choices of the master surface. Obtained results show that there is no discernible difference in the resulting evolution of the twisting moment and the normal contact force. Moreover, Fig. 5.32 shows that besides the accuracy, the efficiency is also unaffected by the choice of the master surface.

Figure 5.33 shows the tangential contact traction magnitude for different rotation angles. One can see that sliding starts to occur at the edge of the contact surface due to a lower normal contact pressure. With increasing the rotation angle, the slip zone progresses towards the rotation centre. To make this clearer, red-coloured contour lines of the constant ratio between the tangential and the normal contact traction are added to Fig. 5.33.
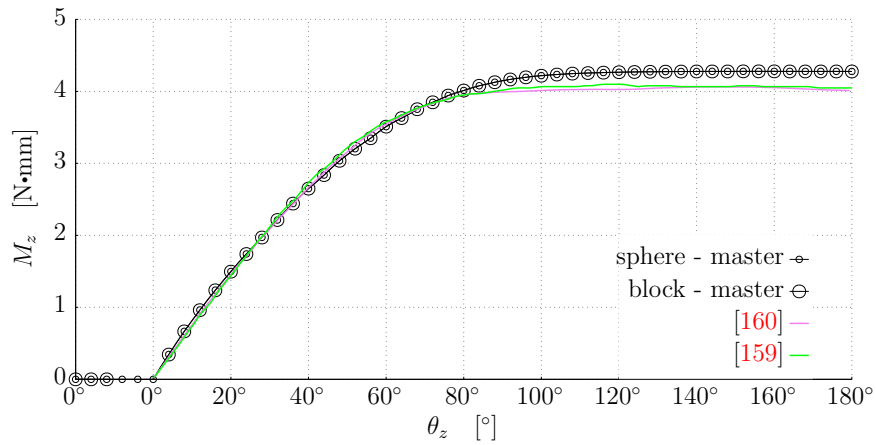
Figure 5.30. Twisting contact between a hemisphere and a block: twisting torque M_z
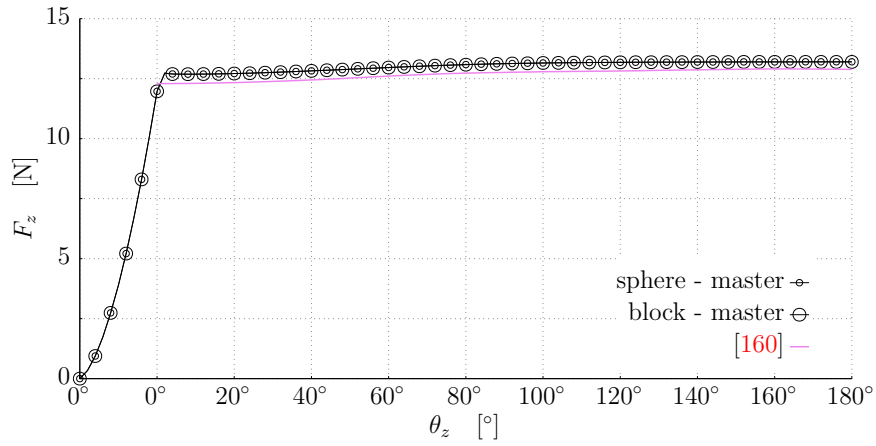
Figure 5.31. Twisting contact between a hemisphere and a block: vertical reaction force F_z
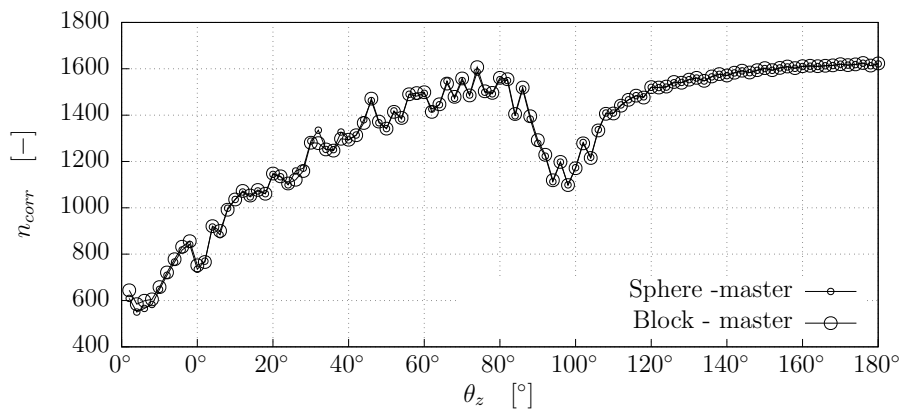
Figure 5.32. Twisting contact between a hemisphere and a block: convergence history

126
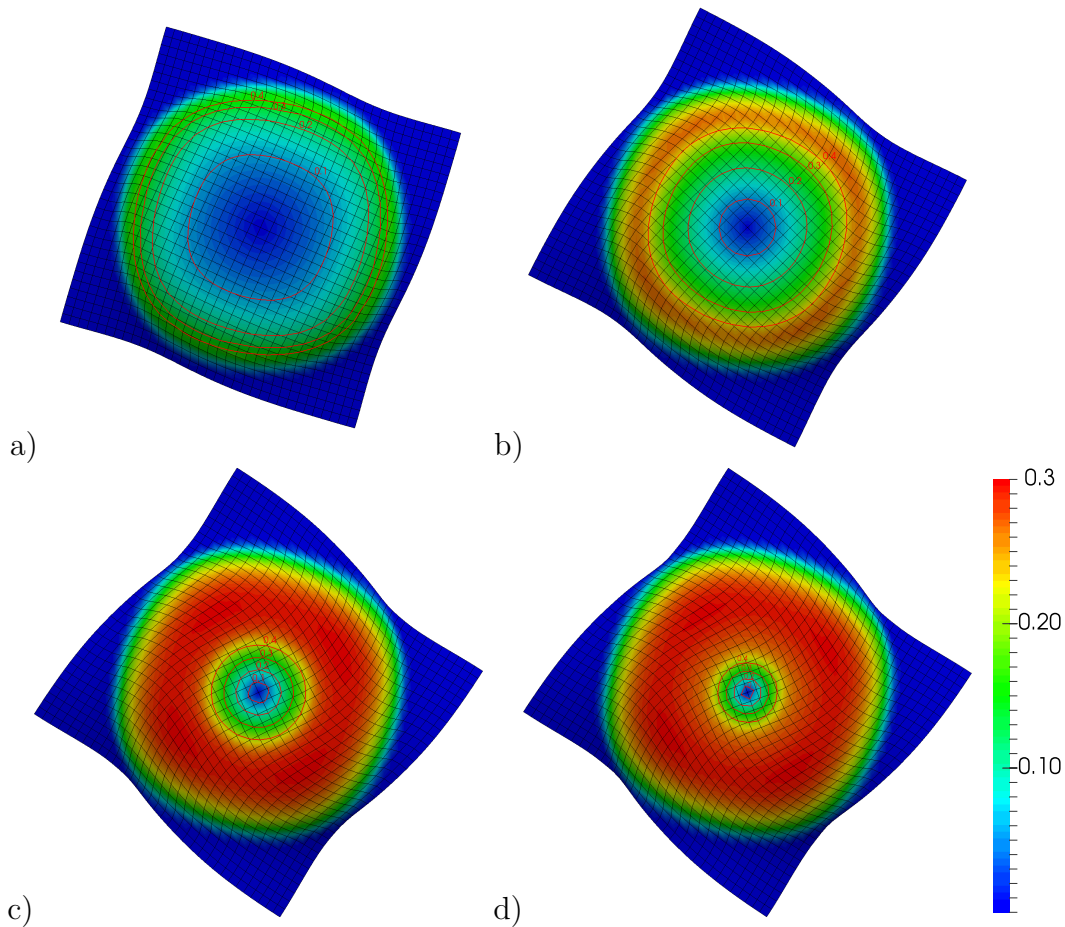
Figure 5.33. Twisting contact between a hemisphere and a block: friction traction magnitude on block contact surface for twisting angles: a) $\theta = 30°$, b) $\theta = 60°$, c) $\theta = 100°$ and d) $\theta = 180°$. Red contour lines represent the ratio between frictional and normal traction

5.4. Industrial Grade Contact Problems

5.4.1. Wire Drawing

Wire drawing is a process in which wire shape, wire size and mechanical properties of the wire are transformed by drawing the wire through a die. As the wire passes through the die, it elongates and reduces its diameter to a desired value. Such a process is widely used in industry due to a wide variety of applications. Moreover, it is a very versatile process since the reduced cross section shape is not restricted only to the circular shape.

Here, we will consider the generic case of isothermal wire drawing, see Fig. 5.34. The considered diameter reduction is 14%, which corresponds to an area reduction

of 26%. The wire initial diameter is $2r_1 = 2$ mm, while the diameter of the die bearing zone is $2r_d = 1.72$ mm. The die approach (entrance) angle is $2\alpha_1 = 12°$, while the back relief angle is $2\alpha_2 = 30°$. The length of the bearing zone or the so-called land length is set to $l_b = 0.45$ mm. To mimic the semi-dry friction conditions, caused by the usage of lubricant, the coefficient of friction is set to $\mu = 0.08$. The material properties of the wire and the die are depicted in Table 5.3. The computational model considers one-quarter of the geometry and wire length of a 12 mm, as shown in Fig. 5.34. The wire downstream cross section surface has a prescribed displacement while the upstream cross section has a prescribed counter pressure of 10 MPa. The chosen drawing speed is 50 m/ min and the simulated time of 0.1 s is resolved using 100 equally spaced time increments (8.3 mm drawn length).
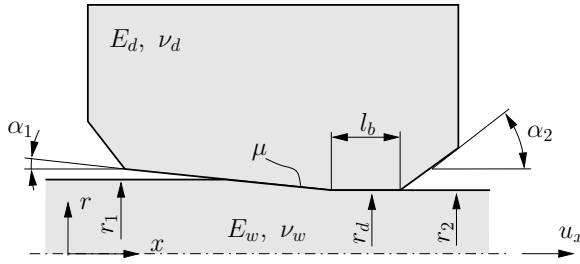
Figure 5.34. Wire drawing: problem geometry

Table 5.3. Wire drawing: material properties

	Wire	Die
Young's modulus E (in GPa)	190	600
Poisson's ratio ν (in GPa)	0.3	0.24
Density ρ (in kg/m³)	7800	7800
Initial yield stress σ_Y (in GPa)	0.85	∞
Hardening coefficient H^{ι} (in GPa)	0.5	–

From the aspect of contact modelling, it is interesting to see how different wire discretisations affect the evolution of the drawing force. Therefore, four computational meshes are used and the wire is discretised by refining the coarse mesh shown in Fig. 5.35. The coarse mesh consists of 2,400 CVs, while the subsequently refined meshes consist of 19,200 (medium), 153,600 (fine) and 1,228,800 CVs (very fine). The die discretisation is held the same in all cases. The die mesh consists of 33,340 CVs and the circumferential curvature is resolved using 70 CVs.

To start the simulation, there are different possibilities regarding the initial wire spatial position. If the wire is set to touch the die, due to the sharp edge of the wire, a high number of iterations can be expected until the wire is drawn through the die. To avoid this, the wire can be filleted at the beginning; such an approach is

reported in [165]. Another possible remedy is to set the wire over the die and within a few time steps gradually scale the die to its original diameter. Such a technique is used here and wire compression is done in the initial time step as the solver is robust enough to handle such a large deformation in the single time step.
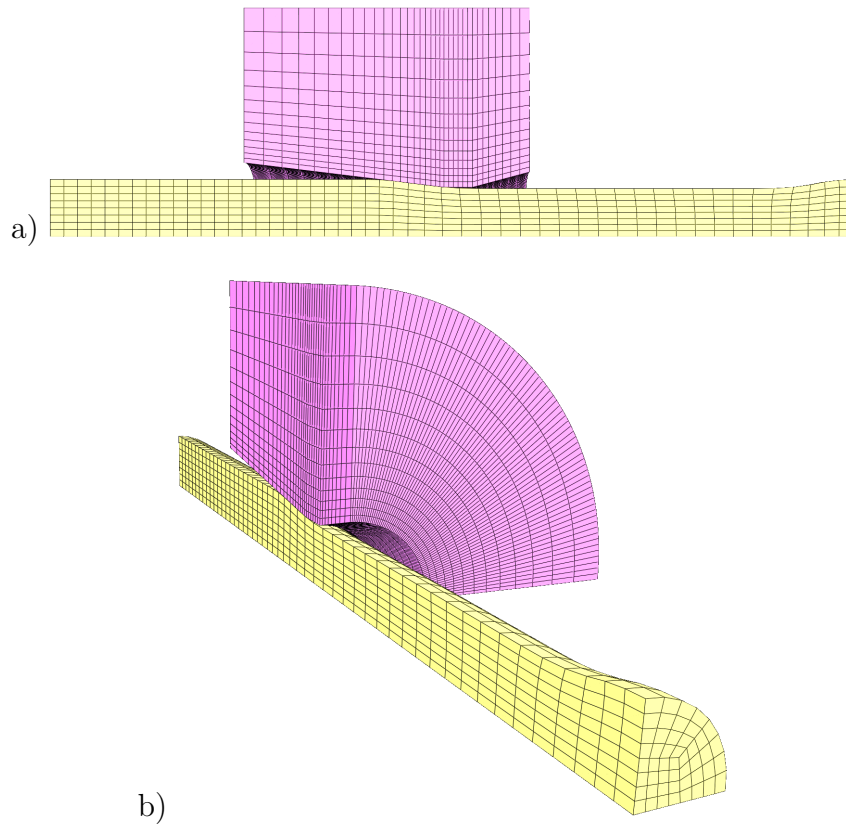
Figure 5.35. Wire drawing: coarse computational mesh at time $t = 0.069$ s: a) symmetry plane view, b) 3D view

Figure 5.36 shows the evolution of drawing force F_x calculated at the one-quarter of the downstream patch where the wire axial displacement is prescribed. As shown, the quasi-static solution is reached quite fast, after the first ten time steps. The averaged values of the drawing force for all meshes are 312.7 (coarse), 301.1 (medium), 291.7 (fine) and 285.0 N (very fine). By applying Richardson's extrapolation [166] the final solution for the drawing force is 282.8 N. The drawing force for the coarse mesh exhibits visible oscillations which deviate from the averaged value by less than 1.8%. These oscillations arise as the control volumes slide over the die transitional edge between the entrance and the bearing zone.
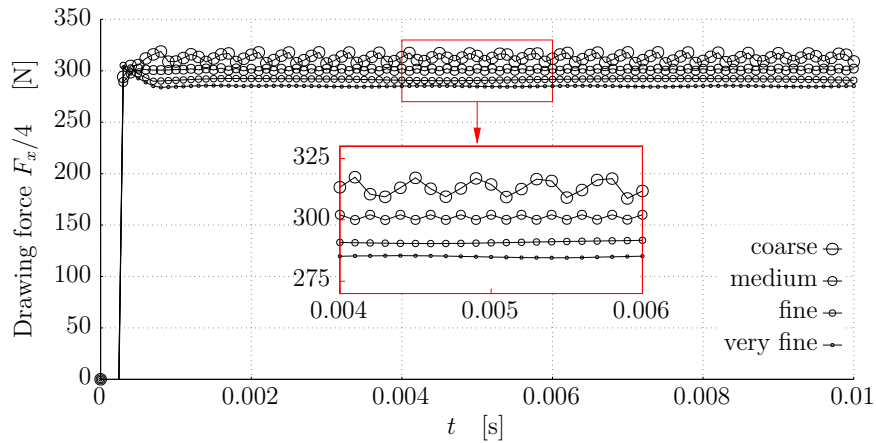
Figure 5.36. Wire drawing: evolution of the drawing force

Figure 5.37 shows the number of outer correctors required to reach the prescribed relative tolerance on all meshes using the same simulation settings. On average, the coarse mesh requires 444, the medium mesh 519, the fine mesh 647 and the very fine mesh requires 910 outer correctors. The finest wire mesh has a 64 times larger number of CVs than the coarsest mesh resulting in twice the number of correctors.
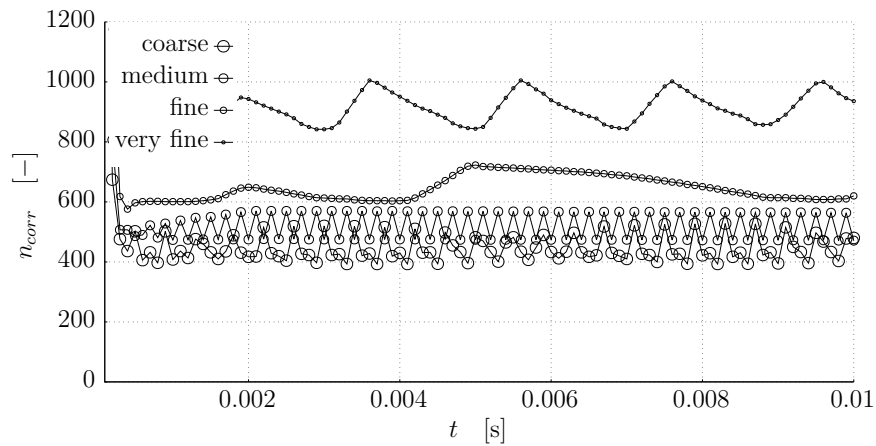
Figure 5.37. Wire drawing: number of outer correctors versus time step

5.4.2. Wire Rolling

Wire rolling is a process in which the workpiece is deformed between one or more pairs of rotating rolls. Figure 5.38 a) schematically shows the considered case in which the circular wire is formed by passing through one pair of rolls with the flat surface. There are many types of rolling processes, for example, if rolls have a specific shape across their width, such a process is called profile rolling.

In the presented example, the cylindrical rollers have a width of 10 mm and their outer and inner radius is 89 mm and 60 mm, respectively. The wire is 80 mm long with an initial diameter of $\phi_w = 2.7$ mm. The prescribed rotational speed of the rollers is $\omega_r = 6.28$ rad/s which corresponds to 60 revolutions per minute. The coefficient of friction between the rollers and the wire is set to $\mu = 0.1$. The gap between the rollers is $\delta = 1.89$ mm, therefore the wire has a 30% linearwise reduction in the initial diameter. Mechanical properties of the wire are given in Table 5.4. The rollers are modelled with the same material properties as the wire, but as hyperelastic.
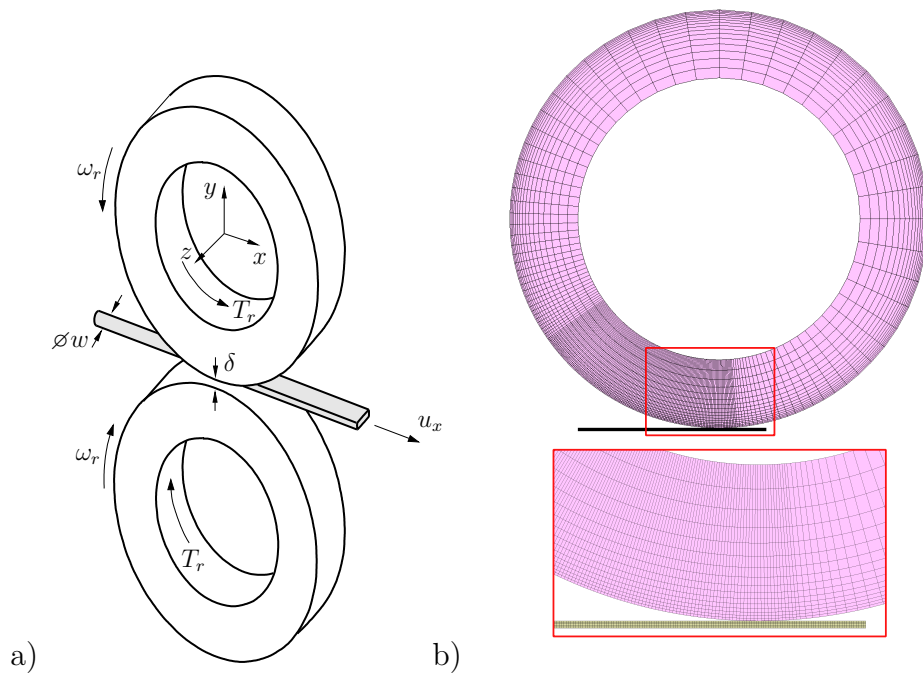
Figure 5.38. Wire rolling: a) problem geometry, b) coarse computational mesh at initial configuration

The computational model considers the quarter symmetry of the geometry as shown in Fig. 5.38 b). The chosen time step corresponds to the roller rotation of 0.126° and the simulation is executed until the steady state is achieved. To ensure convergence and stability, the initial distance between the rollers is equal to the wire diameter and the prescribed gap is reached after the first 40 time steps. Also, during the wire compression phase, the rotational speed of the rollers is gradually increased to the prescribed value. The roller is discretised using 44,096 CVs with an increased mesh density in the circumferential direction at the zone that will be in contact during the simulation time. The wire is discretised using different mesh

densities, i.e. the coarse mesh has 19,440, the medium mesh 65,610, the fine mesh
524,880 and the very fine mesh 1,771,470 CVs.

Table 5.4. Wire rolling: wire mechanical properties

Young's modulus	Poisson's ratio	Density	Initial yield stress	Hardening					
E (in GPa)	ν $(-)$	ρ (in kg/m³)	σ_Y (in GPa)	Plastic strain	0	0.01	0.1	0.5	0.88
210	0.3	7800	1.3	Yield stress (in GPa)	1.3	1.5	1.69	1.64	2.11
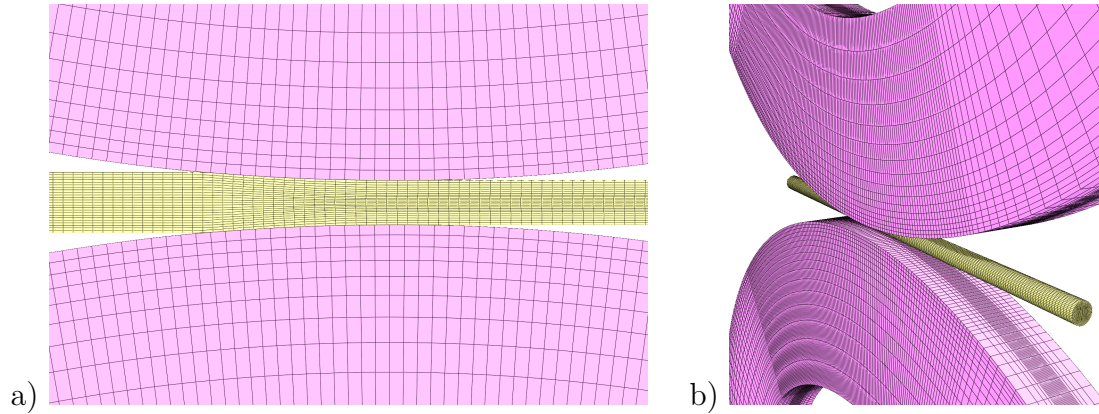
Figure 5.39. Wire rolling: a) roll bite (symmetry plane view), b) 3D view. Images
are generated by mirroring coarse mesh

Fig. 5.39 shows the roll bite in the symmetry plane and the deformed mesh in 3D.
Fig. 5.40 a) shows the evolution of the vertical reaction force at the roller support,
while Fig. 5.40 b) shows the evolution of the resulting roller torque. In steady state,
the difference between the vertical reaction force for the fine and the very fine mesh
is 0.5%, while for the coarse and the very fine mesh is 2.7%. The evolution of the
torque exhibits a less smooth behaviour than the evolution of the vertical reaction
force. The increased wire mesh density did not suppress oscillatory behaviour which
means that for a smoother torque evolution a finer roller mesh would be preferable.
Nevertheless, the very fine mesh proves that the contact algorithm can handle the
large difference in the cells in contact since the axial length of the wire cells is 7
times smaller than the axial length of the roller cells. Finally, the difference in the
averaged roller torque for the very fine mesh and the coarse mesh is 5.2%, which
can be considered satisfactory for the problem of this complexity.
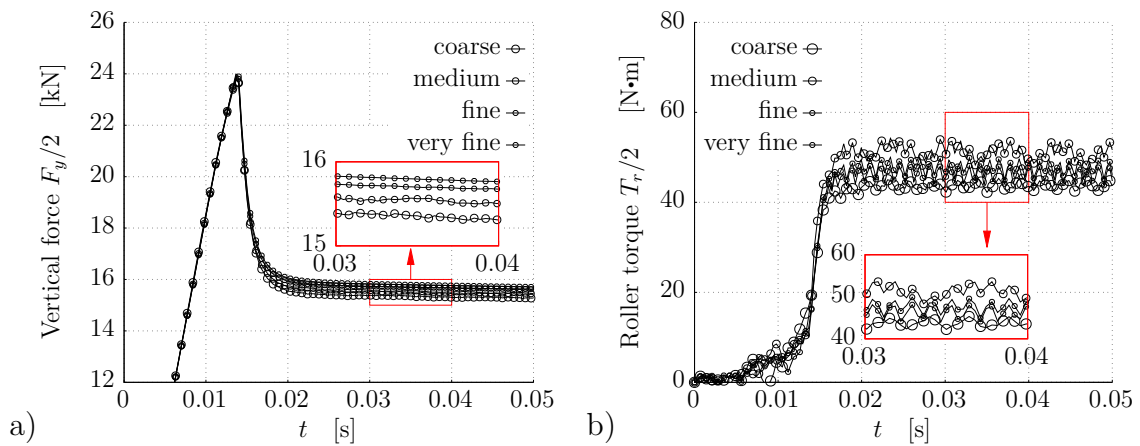
Figure 5.40. Wire rolling: a) evolution of vertical reaction force, b) evolution of roller torque

Fig. 5.41 a) shows the pressure distribution between the roller and the wire on the coarse mesh. One can see that the expected horse-shoe shaped distribution of the contact pressure is obtained [167]. Fig. 5.41 b) shows the predicted steady state deformed cross section, coloured with equivalent plastic strain. The distribution of the equivalent plastic strain has a characteristic X pattern, which is known as the blacksmith's cross [168]. Although finer meshes can better capture local high values of plasticity, the difference in the final shape of the wire can be considered negligible between the meshes.
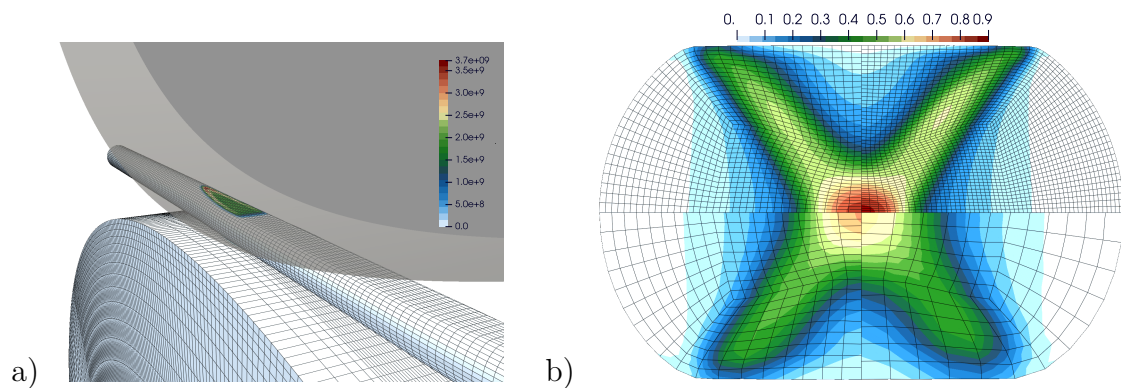
Figure 5.41. Wire rolling: a) horse-shoe shaped distribution of contact pressure on coarse mesh, b) distribution of equivalent plastic strain at wire deformed cross section

5.4.3. Wire Strand Compacting

In this example, compacting of a single-layered wire strand is considered. Here, compacting is performed similarly to the drawing process presented in subsection 5.4.1, with the main difference being a more complex workpiece geometry. After compacting, the contact area between the wires is increased and the amount of gap in the cross-sectional area is reduced, i.e. a better material coverage of the resulting cross-sectional area is obtained. In practice, this leads to an increase in the strength and life span of wire strands [169].

The computational model consists of a straight cylindrical core wire and one layer of six helical wires (1+6 configuration). The lay length is set to 250 mm and the diameter of all wires is set to 5 mm. The die is modelled as rigid, with the following geometric properties: $2\alpha_1 = 14°$, $l_b = 4$ mm and $2\alpha_2 = 30°$, see Fig 5.34 a). The diameter of the bearing zone is 13.5 mm, which corresponds to a 10% reduction in the initial strand diameter. Mechanical properties of the wires are given in Table 5.5. Since the drawing process is heavily lubricated, frictionless contact is assumed, allowing wires to slide freely during compaction.

Table 5.5. Wire strand compacting: mechanical properties

Young's modulus	Poisson's ratio	Density	Initial yield stress	Hardening					
E (in GPa)	ν (−)	ρ (in kg/m^3)	σ_Y (in GPa)	Plastic strain	0	0.05	0.1	0.25	0.5
200	0.3	7800	1.6	Yield stress (in GPa)	1.6	1.64	1.67	1.73	1.75

The computational mesh is shown in Fig. 5.42. The overall mesh size is 651,600 CVs; the core wire is discretised with 136,800 CVs and the outer wires with 85,800 CVs. The considered strand length of 170 mm is discredited with 200 CVs in the axial direction. Complete lay length is not modelled, as the required drawing length to reach a quasi-static solution is relatively short. Also, after reaching a quasi-static solution, the remaining length between the die and the wire downstream cross section is long enough to eliminate the possible effects of the downstream boundary. The wire downstream boundary has a prescribed displacement, whereas the upstream boundary is allowed to elongate in the axial direction. Compacting is performed gradually at the beginning of the simulations at the distance of 20 mm from the downstream wire faces, to rule out the effects of the fixed boundaries.

Fig. 5.43 shows the deformed cross section after compacting. The shape of the outer helical wires takes the form of an isosceles trapezoid, whereas the core wire is slightly formed to a hexagonal shape. As can be seen, the gaps between the wires are reduced and the contact area between the wires is increased, which will result

in lower contact pressures during exploitation. The distribution of the equivalent plastic strain is slightly asymmetric for the outer wires due to their geometrical asymmetry, i.e. twisting direction.

a)
b)
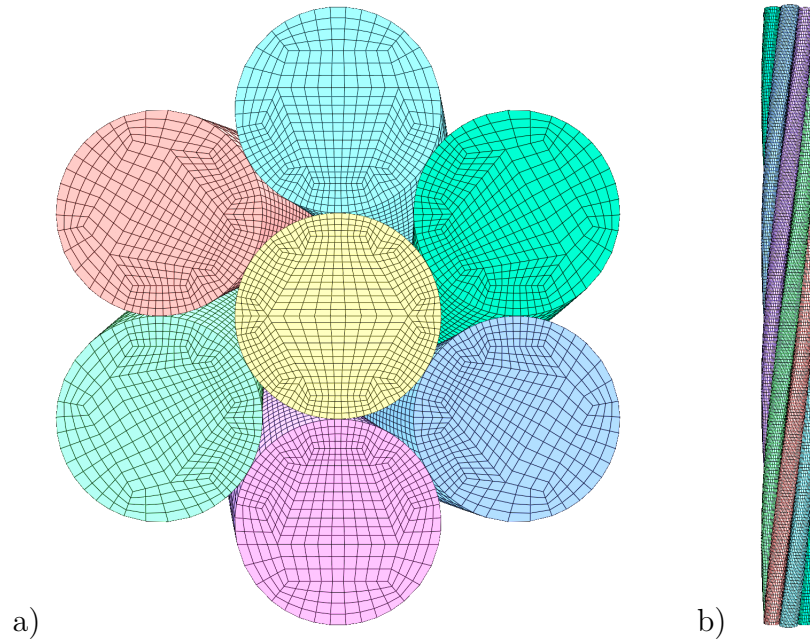
Figure 5.42. Wire strand compacting — computational mesh: a) cross section view, b) plan view
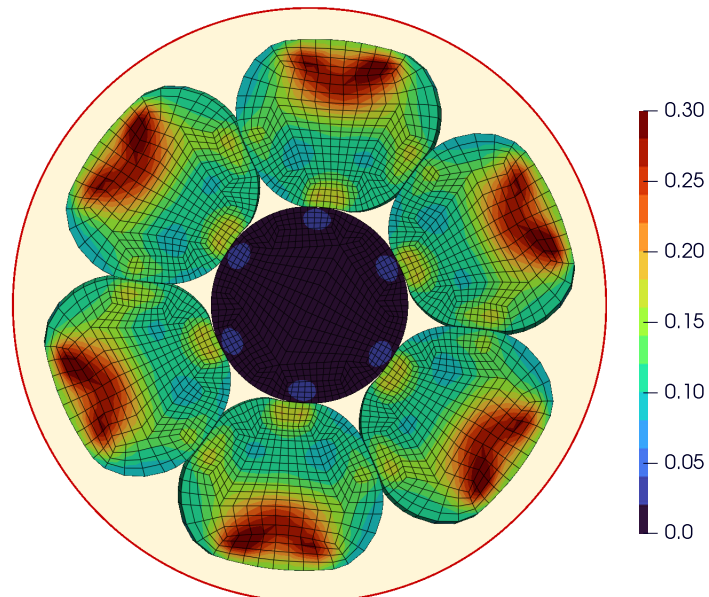
Figure 5.43. Wire strand compacting: deformed cross section, coloured with equivalent plastic strain $\varepsilon^{p,eq}$. Circle filled with light yellow colour denotes cross section area before compacting

5. Numerical examples

Fig. 5.44 shows a 3D view of the compacted wire strand (coloured with equivalent plastic strain) and the rigid die (coloured in blue). After the compaction, outer wires have the largest plastic deformation at the area at which contact with die occur. The distribution of the equivalent plastic strain is uniform in axial direction, which is expected as constant evolution of the drawing force is obtained.
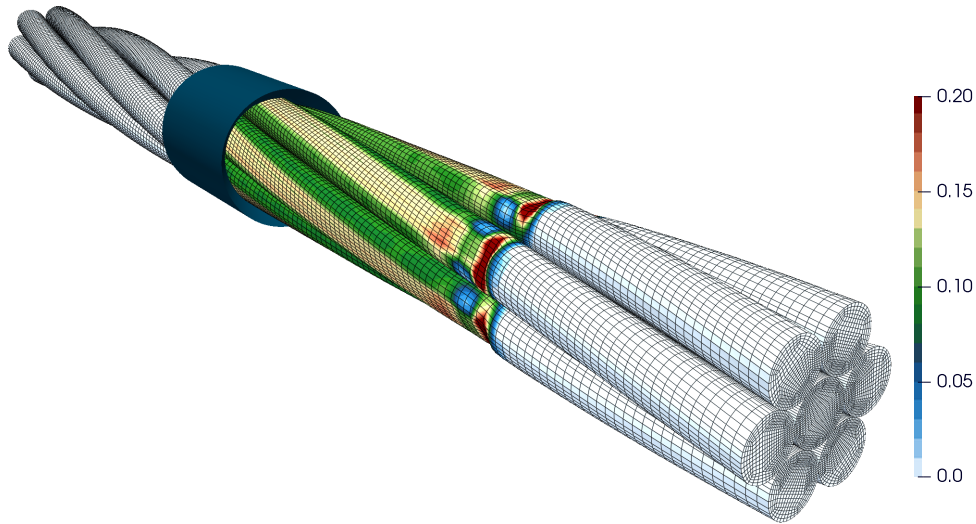
Figure 5.44. Wire strand compacting: compacted wire strand and rigid die in 3D view (coloured with equivalent plastic strain $\varepsilon^{p,eq}$)

Due to the complexity, this numerical example is excellent for testing the implementation of the contact algorithm by checking the evolution of the drawing force. Due to the geometrical and load similarity, the identical drawing force is expected for the outer wires, which is confirmed in Fig 5.45.
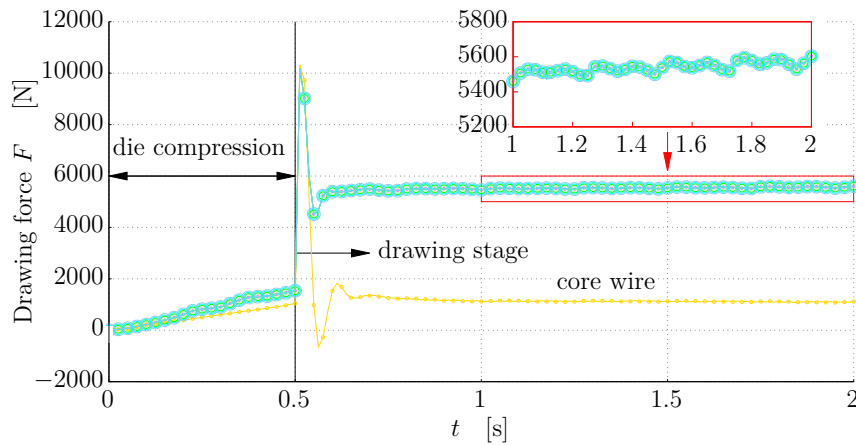
Figure 5.45. Wire strand compacting: evolution of drawing force for core wire and outer wires

136

6. Conclusions and Future Work

The main objective of this thesis was to develop a segment-to-segment contact force calculation algorithm for simulations of large deformation frictional contact problems using the finite volume method.

The thesis relies on the advance finite volume structural solver [2] to describe the hyperelastic and hyperelastoplastic deformation of bodies in contact. The strong integral form of the momentum equation in the updated Lagrangian form is discretised in space with second-order accuracy using the unstructured finite volume method. The system of equations is solved in a segregated manner, resolving nonlinearities and inter-equation coupling using Picard iterations. The movement of the computational mesh is performed using the least-square method.

Contact boundaries are treated using the explicit Neumann-Neumann coupling procedure based on the penalty method in which the contact force is updated based on the achieved penetration and relative sliding between bodies. Such a coupling procedure is proposed with the introduction of the pointwise contact algorithm [1], and it is proven to be stable and accurate in solving challenging contact problems [2]. The same coupling scheme is adopted in this thesis with the aim of developing a more accurate contact-force calculation algorithm that will minimise or eliminate issues of the pointwise contact algorithm.

A newly developed segment-to-segment contact algorithm is extensively tested, and the following conclusions can be outlined:

- The algorithm has passed the patch test; the pressure was transmitted correctly at the flat non-conformal contact interface subjected to uniform pressure distribution.

- The contact is treated symmetrically; the same accuracy and efficiency are obtained irrespective of the choice of the master surface.

- The algorithm is suitable for solving contact problems involving material and geometric nonlinearities. Moreover, it can successfully handle large values of the friction coefficient.

- Problems related to a non-detected penetration or undefined solution of the orthogonal projection are avoided, thus robustness is enhanced.

- The integration of the contact pressure allows accurate prediction of the contact stress at the edge of the contact boundary. Also, during large sliding, contact force is less prone to exhibiting oscillatory behaviour due to a smooth change in penetration.

The main drawbacks of the segment-to-segment algorithm are related to the penalty method and explicit implementation of the contact coupling. Although the usage of the penalty method is strongly beneficial in terms of overall procedure robustness, it results in a finite amount of penetration and elastic slip between bodies due to the approximate enforcement of the contact conditions. The explicit implementation negatively affects the overall efficiency of the solution procedure, significantly extending the computation time. To overcome this, an implicit version of the contact coupling is derived, and the effects of the coupling are investigated. The implicit equations for the boundary gradient and boundary displacement are obtained by performing the Picard linearization of the normal contact traction. The results show that the implicit coupling can offer a significant efficiency improvement while maintaining the same level of accuracy as the explicit coupling. Further research is required into contact problems with material and geometric nonlinearities to achieve the same level of robustness as the explicit coupling procedure.

6.1. Future Work

Future development towards more advance contact treatment procedures can take several directions of which all can be characterized as equally important:

- The huge potential of efficiency improvement lies in the implicit version of the Neumann-Neumann coupling procedure. In further research, the main focus must be kept on increasing robustness and linearisation of the frictional component of the contact traction.

- To be able to simulate self-contact problems it is necessary to implement a suitable contact detection procedure.

- The Nagata surface smoothing technique should be integrated in the segment-to-segment contact-force calculation procedure. Such implementation is undoubtedly a challenging task, however it can be extremely beneficial for some types of contact problems.

- It is necessary to extend investigations into modelling more complex contact interface effects, such as heat dissipation and transmission or surface degradation due to wear.

Besides the listed considerations, future work should be also focused on the more efficient implementation of the underlying numerical algorithms and parallel scalability of the contact algorithm.

Appendices

A Analytical Solutions

The analytical solutions presented in this thesis are used in Chapter 5 to validate the numerical results. Although the presented analytical solutions are widely used in engineering practice, they must be used with caution due to various assumptions on which their derivation is based. Accordingly, when comparing results from numerical procedures, if the computational model does not satisfy their assumptions — at least to an acceptable degree — a notable mismatch in results can be obtained. The mathematical background, assumptions and limitations of the presented analytical solutions can be found in reputable contact mechanics literature listed in the penultimate part of the description of each problem.

A1. Contact Between Cylinders With Parallel Axis

Contact half-width b of two circular cylinders compressed with force F is:

$$b = 2\sqrt{\frac{FR_r}{\pi E_c}}, \qquad (A1)$$

where F is defined per unit length. R_r and E_c are the relative radius of curvature and the contact modulus, which are defined as:

$$\frac{1}{R_r} = \frac{1}{R_1} + \frac{1}{R_2}, \qquad (A2)$$

$$\frac{1}{E_c} = \frac{1 - \nu_1^2}{E_1} + \frac{1 - \nu_2^2}{E_2}. \qquad (A3)$$

At contact surface, the distribution of the normal contact pressure has the semi-elliptical form:

$$\sigma_{zz} = -|p_n(x)| = -p_0\sqrt{1 - \frac{x^2}{b^2}} \qquad \text{for } |x| \leq b, \quad (A4)$$
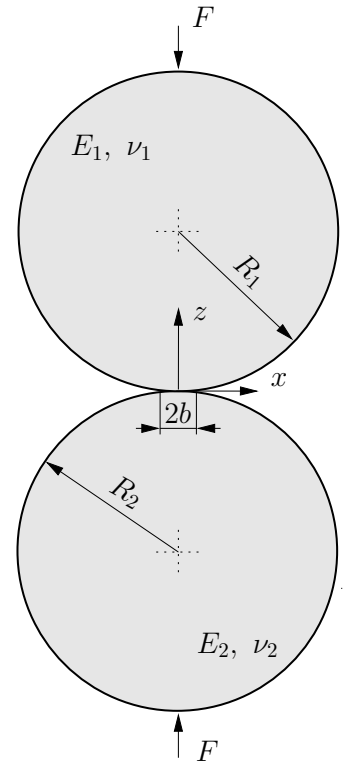
Figure A1. Contact between cylinders with parallel axis

A Analytical Solutions

where p_0 is the peak pressure defined as:

$$p_0 = \sqrt{\frac{F E_c}{\pi R_r}}. \tag{A5}$$

Along the z-axis the components of stress are:

$$\sigma_{xx} = -p_0 \left(\frac{1 + 2\,(z/b)^2}{\sqrt{1 + (z/b)^2}} - 2(z/b) \right),$$

$$\sigma_{zz} = -p_0 \left(1 + \frac{z^2}{b^2} \right)^{-\frac{1}{2}}, \tag{A6}$$

$$\sigma_{xz} = \sigma_{zx} = 0, \quad \sigma_{yy} = \nu(\sigma_{xx} + \sigma_{zz}).$$

Eq. (A6) is derived from the following equations, which allows for the calculation of subsurface stresses at any point:

$$m = \pm\sqrt{\frac{1}{2} \left(\sqrt{(b^2 - x^2 + z^2)^2 + 4x^2 z^2} + b^2 - x^2 + z^2 \right)},$$

$$n = \pm\sqrt{\frac{1}{2} \left(\sqrt{(b^2 - x^2 + z^2)^2 + 4x^2 z^2} - (b^2 - x^2 + z^2) \right)}, \tag{A7}$$

where sign of n and m are $\text{sign}(z)\sqrt{z^2}$ and $\text{sign}(x)\sqrt{x^2}$, respectively.

$$\sigma_{xx} = -\frac{p_0}{b} \left[m \left(1 + \frac{z^2 + n^2}{m^2 + n^2} \right) - 2z \right],$$

$$\sigma_{zz} = -\frac{p_0}{b} m \left(1 - \frac{z^2 + n^2}{m^2 + n^2} \right), \tag{A8}$$

$$\sigma_{xz} = \sigma_{zx} = \frac{p_0}{b} n \left(\frac{m^2 - z^2}{m^2 + n^2} \right).$$

Note that the stresses are independent of Poisson's ratio; however, dependence exists in the case of a plane strain state, i.e. $\sigma_{yy} = \nu(\sigma_{xx} + \sigma_{zz})$. In the case of a contact with a flat surface ($R \to \infty$) the relative radius of curvature is the same as the radius of cylinder $R_c = R$, whereas in the case of a contact with a rigid body ($E \to \infty$), Eq. (A3) is transformed into:

$$\frac{1}{E_c} = \frac{1 - \nu_1^2}{E}. \tag{A9}$$

In the case of existence of friction, the above equations for frictionless contact (Hertzian contact) are still valid if cylinders have the same material properties. Tangential contact pressure p_t, modelled using the Coulomb friction law is added to

the contact surface:

$$
\begin{aligned}
p_t(x) &= \frac{\mu p_0}{b}\left(\sqrt{b^2 - x^2} - \sqrt{c^2 - x^2}\right) && \text{if } |x| \leq c \text{ (stick zone)}, \\
p_t(x) &= \frac{\mu p_0}{b}\left(\sqrt{b^2 - x^2}\right) && \text{if } c < |x| \leq b \text{ (slip zone)},
\end{aligned}
\tag{A10}
$$

where c is the characteristic parameter which defines the boundary between stick and slip zones:

$$
c = b\sqrt{1 - \upsilon/\mu}.
\tag{A11}
$$

Term υ stands for the ratio between the tangential and the normal load. Subsurface stresses are different in the case of frictional contact, therefore Eq. (A8) in its general form is as follows:

$$
\begin{aligned}
\sigma_{xx} &= -\frac{p_0}{b}\left[m\left(1 + \frac{z^2 + n^2}{m^2 + n^2}\right) - 2z + 2\mu(x - n) + \mu n \frac{z^2 - m^2}{m^2 + n^2}\right], \\
\sigma_{zz} &= -\frac{p_0}{b}m\left[1 - \frac{z^2 + n^2}{m^2 + n^2} - \mu\frac{n}{m}\left(\frac{z^2 - m^2}{m^2 + n^2}\right)\right], \\
\sigma_{xz} &= \sigma_{zx} = -\frac{p_0}{b}\left[\mu(m - 2z) - n\frac{z^2 - n^2}{m^2 + n^2} + \mu m \frac{z^2 + n^2}{m^2 + n^2}\right].
\end{aligned}
\tag{A12}
$$

In the case of dissimilar materials, the analytical solution for frictional contact can also be calculated by employing the procedures proposed in [170]. Also, it is worth mentioning that in the case of internal contact, the radius of the cylindrical groove has a negative sign in Eq. (A2).

All the above presented equations, their derivations, limitations and assumptions can be found in the literature on contact mechanics [14, 171, 172].

A2. Contact Between Spheres

In the case of a contact between two spheres, the contact area has a circular shape with radius b:

$$
b = \sqrt[3]{\frac{3FR_r}{4E_c}} = \sqrt{R_c d},
\tag{A13}
$$

where relative radius of curvature R_r and contact modulus E_c are defined using Eq. (A2) and (A3), respectively. The relationship between the applied load and the displacement of sphere centre d is:

$$
F = \frac{4}{3}E_c\sqrt{R_r d^3}.
\tag{A14}
$$

A Analytical Solutions

The normal contact pressure at contact surface has the following parabolic form:

$$\sigma_{zz} = -|p_n(r)| = -p_0\sqrt{1 - \tilde{r}^2} \qquad \text{for } |\tilde{r}| \le 1, \tag{A15}$$

where $\tilde{r}$ is the normalised radial coordinate (normalised with contact radius b). The peak contact pressure (maximum Hertzian pressure) is:

$$p_0 = \frac{3F}{2\pi b^2}. \tag{A16}$$

In the case of frictionless contact i.e. pure indentation, the subsurface stresses written in cylindrical coordinates are:

$$\frac{\sigma_{rr}}{p_0} = \frac{1 - 2\nu}{3}\frac{1}{\tilde{r}^2}\left[1 - \left(\frac{z}{\sqrt{u}}\right)^3\right] + \left(\frac{\tilde{z}}{\sqrt{u}}\right)^3\frac{u}{u^2 + \tilde{z}^2}$$
$$+ \frac{\tilde{z}}{\sqrt{u}}\left[\frac{(1-\nu)u}{1+u} + (1+\nu)\sqrt{u}\tan^{-1}\left(\frac{1}{\sqrt{u}}\right) - 2\right],$$

$$\frac{\sigma_{\theta\theta}}{-p_0} = \frac{1 - 2\nu}{3}\frac{1}{\tilde{r}^2}\left[1 - \left(\frac{\tilde{z}}{\sqrt{u}}\right)^3\right]$$
$$+ \frac{\tilde{z}}{\sqrt{u}}\left[2\nu + \frac{(1-\nu)u}{1+u} - (1+\nu)\sqrt{u}\tan^{-1}\left(\frac{1}{\sqrt{u}}\right)\right], \tag{A17}$$

$$\frac{\sigma_{zz}}{-p_0} = \left(\frac{\tilde{z}}{\sqrt{u}}\right)^3\frac{u}{u^2 + \tilde{z}^2},$$

$$\frac{\tau_{rz}}{-p_0} = \frac{\tilde{r}\tilde{z}^2}{u^2 + \tilde{z}^2}\frac{\sqrt{u}}{1 + u},$$

$$\tau_{z\theta} = \tau_{r\theta} = 0,$$

where term u is calculated as:

$$u = \frac{1}{2}\left(\tilde{r}^2 + \tilde{z}^2 - 1 + \sqrt{(\tilde{r}^2 + \tilde{z}^2 - 1)^2 + 4\tilde{z}^2}\right). \tag{A18}$$

Note that axial coordinate z is also normalised with contact radius b. From Eq. (A17), the stress distribution along axis of symmetry ($r = 0$) is:

$$\frac{\sigma_{\theta\theta}}{-p_0} = \frac{\sigma_{rr}}{-p_0} = (1+\nu)\left[1 - \tilde{z}\tan^{-1}\left(\frac{1}{\tilde{z}}\right)\right] - \frac{1}{2(1 + \tilde{z}^2)},$$

$$\frac{\sigma_{zz}}{-p_0} = \frac{1}{1 + \tilde{z}^2}, \tag{A19}$$

$$\tau_{rz} = \tau_{z\theta} = \tau_{r\theta} = 0.$$

In case of an external contact (contact between sphere and spherical groove), equations are still valid but the spherical groove has a negative sign in the equation for relative radius R_r. Also, the equations are valid in the case of a contact with a flat surface ($R_r = R$) or if a second body is a rigid one, see (A9).

The presented equations are special case of equations for the elliptical contact in the case of a circular contact area. Accordingly, they are also valid in the case of a contact between two crossed cylinders with the same radius.

For more details regarding equations derivations and their assumptions and limitations, interested readers are referred to [82, 161, 173, 174].

A3. Contact Between Flat-Ended Rigid Indenter and Deformable Half-Space

In the case of indentation of an elastic half-space with a flat-ended rigid indenter, the pressure distribution has the following distribution:

$$p_n(x) = \frac{F}{\pi\sqrt{b^2 - x^2}} \qquad \text{for } |x| \leq b, \quad \text{(A20)}$$

which is singular at the edge of the fixed contact zone. The indentation depth, i.e. indenter displacement can be obtained using:

$$d = \frac{F}{\pi E}\left[2\ln\left(\frac{2L}{b}\right) - (1 + \nu)\right], \qquad \text{(A21)}$$

Figure A2. Indentation of a flat-ended indenter into deformable half-space

where L is the thickness of the elastic half-space. The boundary displacement outside the contact zone can be calculated using the following equation:

$$u_z(x) = d + \frac{2(1 - \nu^2)F}{\pi E}\ln\left[\frac{x}{b} + \sqrt{\frac{x^2}{b^2} - 1}\right] \qquad \text{for } |x| \geq b. \qquad \text{(A22)}$$

Note that Eqs. (A20), (A21) and (A22) are derived for the case of thin elastic-half space, i.e. under the plane stress assumption. In the case of an axisymmetric contact, i.e. a circularly shaped indenter, the pressure distribution has the following

form:

$$p_n(x) = \frac{F}{2\pi b\sqrt{b^2 - r^2}} = \frac{Ed}{\pi\sqrt{b^2 - r^2}} \qquad \text{for } |x| \leq b, \tag{A23}$$

which is, the same as above, singular at the edge of the circularly shaped (fixed) contact zone. The correlation between the indentation depth and the applied force is:

$$d = \frac{F(1 - \nu^2)}{2bE}\left(1 - \frac{b}{R}\right). \tag{A24}$$

The above-presented equations and further details can be found in [14, 82, 161].

It should be noted that analytical solutions for more complex shapes of indenter exist in literature [161]. However, the shape complexity typically results in semi-analytical expressions for which numerical integration must be used and they are therefore somehow impractical for benchmarking.

A3.1.   Cylindrical Flat-Ended Indenter With Rounded Edge

In the case of the flat-ended indenter, a possible remedy to avoid pressure singularity is to fillet its edges, see Fig A3. The indentation force and the indentation depth as a function of unknown contact radius b is:

$$F = \frac{E}{3R}\left[\sqrt{b^2 - a^2}(4b^2 - a^2) - 3ab^2\cos^{-1}\left(\frac{a}{b}\right)\right],$$

$$d = \frac{b}{R}\left[\sqrt{b^2 - a^2} - b\,\cos^{-1}\left(\frac{a}{b}\right)\right]. \tag{A25}$$

The normal contact pressure at the contact surface is given in a semi-analytical form [161–163, 175]:

Figure A3.   Flat-ended cylindrical indenter in contact with deformable body

$$\sigma_{zz} = -\frac{E}{\pi R}\begin{cases} \int_a^b \left(2\sqrt{x^2 - a^2} - a\cos^-1\left(\frac{a}{x}\right)\right)\frac{\mathrm{d}x}{\sqrt{x^2 - r^2}}, & \text{for } r \leq a, \\ \int_r^b \left(2\sqrt{x^2 - a^2} - a\cos^-1\left(\frac{a}{x}\right)\right)\frac{\mathrm{d}x}{\sqrt{x^2 - r^2}}, & \text{for } a < r \leq b. \end{cases} \tag{A26}$$

The integration of Eq. (A26) can be avoided by using the non-dimensional solution graph for various ratios a/b, reported in [162].

Bibliography

[1] P. Cardiff, A. Karač, and A. Ivanković, "Development of a finite volume contact solver based on the penalty method," *Computational Materials Science*, vol. 64, pp. 283 – 284, 2012.

[2] P. Cardiff, Ž. Tuković, P. D. Jaeger, M. Clancy, and A. Ivanković, "A Lagrangian cell-centred finite volume method for metal forming simulation," *International Journal for Numerical Methods in Engineering*, vol. 109, no. 13, pp. 1777–1803, 2017.

[3] M. Beaudoin and H. Jasak, "Development of a generalized grid interface for turbomachinery simulations with OpenFOAM," in *Open source CFD International conference*, vol. 2, Berlin, 2008.

[4] P. Cardiff and I. Demirdžić, "Thirty Years of the Finite Volume Method for Solid Mechanics," *Archives of Computational Methods in Engineering (2021)*, Feb 2021.

[5] V. A. Yastrebov, *Numerical methods in contact mechanics.* John Wiley & Sons, 2013.

[6] D. M. S. Neto, *Numerical simulation of frictional contact problems using nagata patches in surface smoothing.* PhD thesis, Faculty of Sciences and Technology of the University of Coimbra, 2014.

[7] I. Batistić, P. Cardiff, and Ž. Tuković, "A finite volume penalty based segment-to-segment method for frictional contact problems," *Applied Mathematical Modelling*, vol. 101, pp. 673–693, 2022.

[8] I. E. Sutherland and G. W. Hodgman, "Reentrant Polygon Clipping," *Commun. ACM*, vol. 17, pp. 32–42, Jan. 1974.

[9] A. Konter, FENet (Project), and National Agency for Finite Element Methods & Standards (Great Britain), *Advanced Finite Element Contact Benchmarks.* NAFEMS Limited, 2006.

[10] D. Neto, M. Oliveira, L. Menezes, and J. Alves, "A contact smoothing method for arbitrary surface meshes using Nagata patches," *Computer Methods in Applied Mechanics and Engineering*, vol. 299, pp. 283 – 315, 2016.

[11] I. Demirdžić, P. Martinović, and A. Ivanković, "Numerical simulation of thermal deformation in welded workpiece," *Zavarivanje*, vol. 31, no. 5, pp. 209–219, 1988.

[12] I. Demirdžić and S. Muzaferija, "Finite volume method for stress analysis in complex domains," *International Journal for Numerical Methods in Engineering*, vol. 37, no. 21, pp. 3751–3766, 1994.

[13] H. Hertz, "Über die berührung fester elastischer körper," *Journal für die reine und angewandte Mathematik*, vol. 1882, no. 92, pp. 156–171, 1882.

[14] K. L. Johnson, *Contact Mechanics.* Cambridge University Press, 1985.

[15] I. Demirdžić, "Finite volumes vs finite elements. There is a choice," *Coupled systems mechanics*, vol. 9, no. 1, pp. 5–28, 2020.

[16] I. Demirdžić and S. Muzaferija, "Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology," *Computer Methods in Applied Mechanics and Engineering*, vol. 125, no. 1, pp. 235–255, 1995.

[17] M. A. Wheel, "A geometrically versatile finite volume formulation for plane elastostatic stress analysis," *The Journal of Strain Analysis for Engineering Design*, vol. 31, no. 2, pp. 111–116, 1996.

[18] H. Jasak and H. G. Weller, "Application of the finite volume method and unstructured meshes to linear elasticity," *International Journal for Numerical Methods in Engineering*, vol. 48, no. 2, pp. 267–287, 2000.

[19] I. Bijelonja, I. Demirdžić, and S. Muzaferija, "A finite volume method for large strain analysis of incompressible hyperelastic materials," *International Journal for Numerical Methods in Engineering*, vol. 64, no. 12, pp. 1594–1609, 2005.

[20] I. Bijelonja, I. Demirdžić, and S. Muzaferija, "Mixed finite volume method for linear thermoelasticity at all Poisson's ratios," *Numerical Heat Transfer, Part A: Applications*, vol. 72, no. 3, pp. 215–235, 2017.

[21] J. Fainberg and H.-J. Leister, "Finite volume multigrid solver for thermo-elastic stress analysis in anisotropic materials," *Computer Methods in Applied Mechanics and Engineering*, vol. 137, no. 2, pp. 167–174, 1996.

[22] I. Demirdžić, I. Horman, and D. Martinović, "Finite volume analysis of stress and deformation in hygro-thermo-elastic orthotropic body," *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 8, pp. 1221–1232, 2000.

[23] P. Cardiff, A. Karač, and A. Ivanković, "A large strain finite volume method for orthotropic bodies with general material orientations," *Computer Methods in Applied Mechanics and Engineering*, vol. 268, pp. 318–335, 2014.

[24] Ž. Tuković, A. Ivanković, and A. Karač, "Finite-volume stress analysis in multi-material linear elastic body," *International Journal for Numerical Methods in Engineering*, vol. 93, no. 4, pp. 400–419, 2013.

[25] I. Demirdžić and D. Martinović, "Finite volume method for thermo-elasto-plastic stress analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 109, no. 3, pp. 331 – 349, 1993.

[26] V. Kanyanta, A. Ivanković, and A. Karač, "Validation of a fluid-structure interaction numerical model for predicting flow transients in arteries," *Journal of Biomechanics*, vol. 42, no. 11, pp. 1705 – 1712, 2009.

[27] A. Karač and A. Ivanković, "Investigating the behaviour of fluid-filled polyethylene containers under base drop impact: A combined experimental/numerical approach," *International Journal of Impact Engineering*, vol. 36, no. 4, pp. 621 – 631, 2009.

[28] S. Kelly and M. O'Rourke, "Fluid, solid and fluid-structure interaction simulations on patient-based abdominal aortic aneurysm models," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 226, no. 4, pp. 288–304, 2012.

[29] A. Ivanković, S. Muzaferija, and I. Demirdžić, "Finite volume method and multigrid acceleration in modelling of rapid crack propagation in full-scale pipe test," *Computational Mechanics*, vol. 20, no. 1, pp. 46–52, 1997.

Bibliography

[30] A. Ivanković, K. Pandya, and J. Williams, "Crack growth predictions in polyethylene using measured traction-separation curves," *Engineering Fracture Mechanics*, vol. 71, no. 4, pp. 657 – 668, 2004.

[31] A. Karač, B. Blackman, V. Cooper, A. Kinloch, S. Rodriguez Sanchez, W. Teo, and A. Ivanković, "Modelling the fracture behaviour of adhesively-bonded joints as a function of test rate," *Engineering Fracture Mechanics*, vol. 78, no. 6, pp. 973 – 989, 2011.

[32] N. Murphy and A. Ivanković, "The prediction of dynamic fracture evolution in PMMA using a cohesive zone model," *Engineering Fracture Mechanics*, vol. 72, no. 6, pp. 861 – 875, 2005.

[33] N. Murphy, M. Ali, and A. Ivanković, "Dynamic crack bifurcation in PMMA," *Engineering Fracture Mechanics*, vol. 73, no. 16, pp. 2569 – 2587, 2006.

[34] Y. Fryer, C. Bailey, M. Cross, and C.-H. Lai, "A control volume procedure for solving the elastic stress-strain equations on an unstructured mesh," *Applied Mathematical Modelling*, vol. 15, no. 11, pp. 639–645, 1991.

[35] C. Bailey and M. Cross, "A finite volume procedure to solve elastic solid mechanics problems in three dimensions on an unstructured mesh," *International Journal for Numerical Methods in Engineering*, vol. 38, no. 10, pp. 1757–1776, 1995.

[36] G. Taylor, C. Bailey, and M. Cross, "Solution of the elastic/visco-plastic constitutive equations: A finite volume approach," *Applied Mathematical Modelling*, vol. 19, no. 12, pp. 746–760, 1995.

[37] A. Slone, C. Bailey, and M. Cross, "Dynamic solid mechanics using finite volume methods," *Applied Mathematical Modelling*, vol. 27, no. 2, pp. 69–87, 2003.

[38] R. Suliman, O. Oxtoby, A. Malan, and S. Kok, "An enhanced finite volume method to model 2D linear elastic structures," *Applied Mathematical Modelling*, vol. 38, no. 7, pp. 2265–2279, 2014.

[39] G. A. Taylor, C. Bailey, and M. Cross, "A vertex-based finite volume method applied to non-linear material problems in computational solid mechanics,"

International Journal for Numerical Methods in Engineering, vol. 56, no. 4, pp. 507–529, 2003.

[40] L. M. Vieira, M. Giacomini, R. Sevilla, and A. Huerta, "A second-order face-centred finite volume method for elliptic problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 358, p. 112655, 2020.

[41] S. Beale and S. Elias, "Stress distribution in a plate subject to uniaxial loading," *PHOENICS J. Comput. Fluid Dyn. Appl*, vol. 3, no. 3, pp. 255–287, 1990.

[42] P. Cardiff, Ž. Tuković, H. Jasak, and A. Ivanković, "A block-coupled finite volume methodology for linear elasticity and unstructured meshes," *Computers & Structures*, vol. 175, pp. 100 – 122, 2016.

[43] S. Das, S. R. Mathur, and J. Y. Murthy, "An Unstructured Finite-Volume Method for Structure–Electrostatics Interactions in MEMS," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 60, no. 6, pp. 425–451, 2011.

[44] L. Ribeiro de Azevedo, *Development of a block-coupled finite volume methodology for non linear elasticity.* PhD thesis, Technische Universität Darmstadt, 2021.

[45] I. Demirdžić, S. Muzaferija, and M. Perić, "Benchmark solutions of some structural analysis problems using finite-volume method and multigrid acceleration," *International Journal for Numerical Methods in Engineering*, vol. 40, no. 10, pp. 1893–1908, 1997.

[46] Ž. Tuković, P. Cardiff, A. Karač, H. Jasak, and A. Ivanković, "OpenFOAM library for fluid structure interaction," in *9th OpenFOAM Workshop*, vol. 710, 2014.

[47] T. Tang, O. Hededal, and P. Cardiff, "On finite volume method implementation of poro-elasto-plasticity soil model," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 39, no. 13, pp. 1410–1430, 2015.

[48] I. Demirdžić, "A fourth-order finite volume method for structural analysis," *Applied Mathematical Modelling*, vol. 40, no. 4, pp. 3104–3114, 2016.

[49] I. Georgiou, A. Ivanković, A. Kinloch, and V. Tropša, "Rate Dependent Fracture Behaviour of Adhesively Bonded Joints," vol. 32 of *European Structural Integrity Society*, pp. 317–328, Elsevier, 2003.

[50] P. Cardiff, *Development of the finite volume method for hip joint stress analysis.* PhD thesis, University College Dublin. School of Mechanical & Materials Engineering., 2012.

[51] P. Cardiff, A. Karač, D. FitzPatrick, R. Flavin, and A. Ivanković, "Development of a hip joint model for finite volume simulations," *Journal of biomechanical engineering*, vol. 136, no. 1, p. 011006, 2014.

[52] H. Jasak and H. Weller, "Finite volume methodology for contact problems of linear elastic solids," in *Proceedings of 3rd International Conference of Croatian Society of Mechanics, Cavtat/Dubrovnik*, pp. 253–260, 2000.

[53] I. Georgiou, *The fracture of adhesively-bonded aluminium joints for automotive structures.* PhD thesis, Imperial College London, 2003.

[54] V. Tropša, I. Georgiou, A. Ivanković, A. Kinloch, and J. Williams, "OpenFOAM in non-linear stress analysis: modelling of adhesive joints," in *First OpenFOAM Workshop, Zagreb, Croatia*, 2006.

[55] H. Elsafti, *Modelling and Analysis of Wave-Structure-Foundation Interaction for Monolithic Breakwaters.* PhD thesis, Technical University of Braunschweig, 2015.

[56] V. Škurić, P. D. Jaeger, and H. Jasak, "Lubricated elastoplastic contact model for metal forming processes in OpenFOAM," *Computers & Fluids*, vol. 172, pp. 226 – 240, 2018.

[57] P. Cardiff, A. Karač, P. De Jaeger, H. Jasak, J. Nagy, A. Ivanković, and Ž. Tuković, "An open-source finite volume toolbox for solid mechanics and fluid-solid interaction simulations," *arXiv preprint arXiv:1808.10736*, 2018.

[58] A. Scolaro, C. Fiorina, I. Clifford, and A. Pautz, "Development of a Semi-Implicit Contact Methodology for Finite Volume Stress Solvers," *International Journal for Numerical Methods in Engineering*, vol. 123, no. 2, pp. 309–338.

[59] P. Farrell and J. Maddison, "Conservative interpolation between volume meshes by local Galerkin projection," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 1-4, pp. 89–100, 2011.

[60] G. Taylor, V. Breiguine, C. Bailey, and M. Cross, "An augmented Lagrangian contact algorithm employing a vertex-based finite volume method," in *The 8th Annual Conference of the Association for Computational Mechanics in Engineering (ACME 2000)*, pp. 163–166, 2000.

[61] N. Bessonov, S. Golovashchenko, and V. Volpert, "Numerical modelling of contact elastic-plastic flows," *Mathematical Modelling of Natural Phenomena*, vol. 4, no. 1, pp. 44–87, 2009.

[62] J. Haider, C. H. Lee, A. J. Gil, A. Huerta, and J. Bonet, "An upwind cell centred Total Lagrangian finite volume algorithm for nearly incompressible explicit fast solid dynamic applications," *Computer Methods in Applied Mechanics and Engineering*, vol. 340, pp. 684–727, 2018.

[63] P. Wriggers, *Computational contact mechanics.* Berlin, Springer-Verlag, 2006.

[64] T. A. Laursen, *Computational contact and impact mechanics: fundamentals of modeling interfacial phenomena in nonlinear finite element analysis.* Springer Science & Business Media, 2013.

[65] J. Hallquist, G. Goudreau, and D. Benson, "Sliding interfaces with contact-impact in large-scale Lagrangian computations," *Computer Methods in Applied Mechanics and Engineering*, vol. 51, no. 1, pp. 107–137, 1985.

[66] P. Wriggers, T. Vu Van, and E. Stein, "Finite element formulation of large deformation impact-contact problems with friction," *Computers & Structures*, vol. 37, no. 3, pp. 319–331, 1990.

[67] N. El-Abbasi and K. J. Bathe, "Stability and patch test performance of contact discretizations and a new solution algorithm," *Computers & Structures*, vol. 79, no. 16, pp. 1473–1486, 2001.

[68] J. C. Simo, P. Wriggers, and R. L. Taylor, "A perturbed lagrangian formulation for the finite element solution of contact problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 50, no. 2, pp. 163–180, 1985.

[69] G. Zavarise and P. Wriggers, "A segment-to-segment contact strategy," *Mathematical and Computer Modelling*, vol. 28, no. 4, pp. 497–515, 1998. Recent Advances in Contact Mechanics.

[70] M. A. Puso and T. A. Laursen, "A mortar segment-to-segment frictional contact method for large deformations," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, no. 45, pp. 4891–4913, 2004.

[71] M. A. Puso and T. A. Laursen, "A mortar segment-to-segment contact method for large deformation solid mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, no. 6, pp. 601–629, 2004.

[72] B. Yang and T. A. Laursen, "A contact searching algorithm including bounding volume trees applied to finite sliding mortar formulations," *Computational Mechanics*, vol. 41, no. 2, pp. 189–205, 2008.

[73] T. Nagata, "Simple local interpolation of surfaces using normal vectors," *Computer Aided Geometric Design*, vol. 22, no. 4, pp. 327–347, 2005.

[74] D. Neto, M. Oliveira, L. Menezes, and J. Alves, "Applying Nagata patches to smooth discretized surfaces used in 3D frictional contact problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 271, pp. 296–320, 2014.

[75] H. Jasak, A. Jemcov, and Ž. Tuković, "OpenFOAM: A C++ library for complex physics simulations," in *International workshop on coupled methods in numerical dynamics*, vol. 1000, pp. 1–20, IUC Dubrovnik Croatia, 2007.

[76] J. C. Simo and T. J. Hughes, *Computational inelasticity*, vol. 7. Springer-Verlag, 1998.

[77] G. A. Holzapfel, *Nonlinear Solid Mechanics: A Continuum Approach for Engineering Science*. Wiley, New York, 1 ed., 2002.

[78] E. W. Chaves, *Notes on continuum mechanics*. Springer Science & Business Media, 1 ed., 2013.

[79] T. Belytschko, W. K. Liu, B. Moran, and K. Elkhodary, *Nonlinear finite elements for continua and structures*. John wiley & sons, 2 ed., 2014.

[80] K. J. Bathe, *Finite element procedures*. Prentice-Hall, New Jersey, 2 ed., 2006.

[81] J. Bonet and R. D. Wood, *Nonlinear Continuum Mechanics for Finite Element Analysis.* Cambridge University Press, 2 ed., 2008.

[82] D. Hills, D. Nowell, and A. Sackfield, *Mechanics of Elastic Contacts.* Butterworth-Heinemann, 1993.

[83] A. Konyukhov and K. Schweizerhof, "On the solvability of closest point projection procedures in contact analysis: Analysis and solution strategy for surfaces of arbitrary geometry," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 33, pp. 3045–3056, 2008.

[84] B. Yang, T. A. Laursen, and X. Meng, "Two dimensional mortar contact methods for large deformation frictional sliding," *International Journal for Numerical Methods in Engineering*, vol. 62, no. 9, pp. 1183–1225, 2005.

[85] B. R. Akula, *Extended mortar method for contact and mesh-tying applications.* PhD thesis, Université Paris sciences et lettres, 2019.

[86] A. Popp, *Mortar methods for computational contact mechanics and general interface problems.* PhD thesis, Technische Universität München, 2012.

[87] K. Poulios and Y. Renard, "An unconstrained integral approximation of large sliding frictional contact between deformable solids," *Computers & Structures*, vol. 153, pp. 75–90, 2015.

[88] V. Yastrebov, *Computational contact mechanics: geometry, detection and numerical techniques.* PhD thesis, École Nationale Supérieure des Mines de Paris, 2011.

[89] A. Heege and P. Alart, "A Frictional Contact Element for Strongly Curved Contact Problems," *International Journal for Numerical Methods in Engineering*, vol. 39, pp. 165–184, 1996.

[90] P. Wriggers and G. Zavarise, "Computational contact mechanics," *Encyclopedia of computational mechanics*, 2004.

[91] Ž. Tuković, A. Karač, P. Cardiff, H. Jasak, and A. Ivanković, "OpenFOAM finite volume solver for fluid-solid interaction," *Transactions of FAMENA*, vol. 42, no. 3, pp. 1–31, 2018.

[92] H. Jasak, *Error analysis and estimation for the finite volume method with applications to fluid flows.* PhD thesis, Imperial College London (University of London), 1996.

[93] I. Demirdžić, "On the discretization of the diffusion term in finite-volume continuum mechanics," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 68, no. 1, pp. 1–10, 2015.

[94] C. M. Rhie and W. L. Chow, "Numerical study of the turbulent flow past an airfoil with trailing edge separation," *AIAA Journal*, vol. 21, no. 11, pp. 1525–1532, 1983.

[95] A. Jameson, W. Schmidt, and E. Turkel, "Numerical solution of the euler equations by finite volume methods using runge kutta time stepping schemes," *In: American Institute of Aeronautics and Astronautics 5th computational fluid dynamics conference*, vol. 81, p. 1259, 1981.

[96] M. Aguirre, A. Gil, J. Bonet, and C. H. Lee, "An upwind vertex centred finite volume solver for lagrangian solid dynamics," *Journal of Computational Physics*, vol. 300, pp. 387–422, 2015.

[97] P. Cardiff, A. Ivanković, A. Karač, *et al.*, "Development of a finite volume based structural solver for large rotation of non-orthogonal meshes," in *7th OpenFOAM Workshop, Technische Universität Darmstadt Germany 25-28 June 2012 Darmstadt*, OpenFOAM Workshop, 2012.

[98] D. S. Kershaw, "The incomplete Cholesky—conjugate gradient method for the iterative solution of systems of linear equations," *Journal of Computational Physics*, vol. 26, no. 1, pp. 43–65, 1978.

[99] D. Jacobs, *Preconditioned conjugate gradient methods for solving systems of algebraic equations.* CERL, Central Electricity Research Laboratories, 1981.

[100] K. Maneeratana, *Development of the finite volume method for non-linear structural applications.* PhD thesis, Imperial College London (University of London), 2000.

[101] K. Maneeratana and A. Ivanković, "Finite volume method for structural applications involving material and geometrical non-linearities," in *Proceedings of ECCM*, vol. 99, 1999.

[102] K. Maneeratana and A. Ivanković, "Finite volume method for geometrically nonlinear stress analysis applications," in *Proc. ACME'99, The Seventh Annual Conference*, pp. 117–120, 1999.

[103] Ž. Tuković and H. Jasak, "Updated lagrangian finite volume solver for large deformation dynamic response of elastic body," *Transactions of FAMENA*, vol. 31, no. 1, pp. 55–70, 2007.

[104] H. Bašić, I. Demirdžić, and S. Muzaferija, "Finite volume method for simulation of extrusion processes," *International Journal for Numerical Methods in Engineering*, vol. 62, no. 4, pp. 475–494, 2005.

[105] I. Bijelonja, I. Demirdžić, and S. Muzaferija, "A finite volume method for incompressible linear elasticity," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 44, pp. 6378 – 6390, 2006.

[106] I. Oliveira, Ž. Tuković, J. Gasche, and P. Cardiff, "Semicoupled numerical strategy to solve large-strain motion of incompressible hyperelastic materials," in *The 16th OpenFOAM Workshop, Dublin, Ireland*, 2021.

[107] Ž. Tuković, I. Batistić, P. Cardiff, H. Jasak, and A. Ivanković, "Block-coupled finite volume solver for incompressible linear elasticity," in *OpenFOAM Workshop 2019,Duisburg, Germany*, 2019.

[108] N. Falah, "A method for calculation of face gradients in two-dimensional, cell centred, finite volume formulation for stress analysis in solid problems," vol. 15, pp. 286–294, 2008.

[109] J. M. Nordbotten, "Cell-centered finite volume discretizations for deformable porous media," *International Journal for Numerical Methods in Engineering*, vol. 100, no. 6, pp. 399–418, 2014.

[110] Ž. Tuković and H. Jasak, "A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow," *Computers & Fluids*, vol. 55, pp. 70–84, 2012.

[111] I. Demirdžić and A. Ivanković, "Finite volume approach to modelling of plates," in *Proceedings of 2nd Congress of Croatian Society of mechanics, Brač, Croatia*, pp. 101–108, 1997.

[112] A. Golubović, I. Demirdžić, and S. Muzaferija, "Finite volume analysis of laminated composite plates," *International Journal for Numerical Methods in Engineering*, vol. 109, no. 11, pp. 1607–1620, 2017.

[113] S. Das, S. R. Mathur, and J. Y. Murthy, "Finite-volume method for creep analysis of thin rf mems devices using the theory of plates," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 61, no. 2, pp. 71–90, 2012.

[114] Ž. Tuković, P. De Jaeger, P. Cardiff, and A. Ivanković, "A Finite volume solver for geometrically exact Simo-Reissner beams," in *4th International Conference on Multi-scale Computational Methods for Solids and Fluids, Faculty of Civil Engineering, University of Sarajevo*, pp. 410–413, 2019.

[115] B. Seevani, Ž. Tuković, P. Cardiff, A. Ivanković, and V. Pakrashi, "A cell-centred finite volume formulation of geometrically-exact Simo-Reissner beams with arbitrary initial curvatures," *ArXiv*, vol. abs/2109.02951, 2021.

[116] J. Hallquist, G. Goudreau, and D. Benson, "Sliding interfaces with contact-impact in large-scale Lagrangian computations," *Computer Methods in Applied Mechanics and Engineering*, vol. 51, no. 1, pp. 107 – 137, 1985.

[117] I. De Dominicis, G. Cvijetić, M. Willetts, and H. Jasak, *Enhanced Turbomachinery Capabilities for Foam-Extend: Development and Validation*, pp. 145–155. Cham: Springer International Publishing, 2019.

[118] *OpenFOAM Turbo Tools: From General Purpose CFD to Turbomachinery Simulations*, vol. ASME-JSME-KSME 2011 Joint Fluids Engineering Conference: Volume 1, Symposia – Parts A, B, C, and D of *Fluids Engineering Division Summer Meeting*, 07 2011.

[119] J. M. Van Verth and L. M. Bishop, *Essential mathematics for games and interactive applications.* CRC Press, 3 ed., 2015.

[120] G. van den Bergen, *Collision Detection in Interactive 3D Environments.* The Morgan Kaufmann Series in Interactive 3D Technology, San Francisco: Morgan Kaufmann, 2003.

[121] C. Ericson, "Real-Time Collision Detection," The Morgan Kaufmann Series in Interactive 3D Technology, pp. 75–123, San Francisco: Morgan Kaufmann, 2005.

[122] D. J. Benson and J. O. Hallquist, "A single surface contact algorithm for the post-buckling analysis of shell structures," *Applied Mechanics and Engineering*, vol. 78, pp. 141–163, 1990.

[123] M. Oldenburg and L. Nilsson, "The position code algorithm for contact searching," *International Journal for Numerical Methods in Engineering*, vol. 37, no. 3, pp. 359–386, 1994.

[124] J. Klosowski, M. Held, J. Mitchell, H. Sowizral, and K. Zikan, "Efficient collision detection using bounding volume hierarchies of k-dops," *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, no. 1, pp. 21–36, 1998.

[125] S. Gottschalk, M. C. Lin, and D. Manocha, "OBBTree: A hierarchical structure for rapid interference detection," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 171–180, 1996.

[126] V. Mataix Ferrándiz, *Innovative mathematical and numerical models for studying the deformation of shells during industrial forming processes with the Finite Element Method.* PhD thesis, Universitat Politècnica de Catalunya, 2020.

[127] S. Gottschalk, *Collision Queries Using Oriented Bounding Boxes.* PhD thesis, The University of North Carolina at Chapel Hill, 2000.

[128] M. I. Shamos, *Computational geometry.* 1978.

[129] J. Linahan, "A Geometric Interpretation of the Boolean Gilbert-Johnson-Keerthi Algorithm," *arXiv preprint arXiv:1505.07873*, 2015.

[130] K. Weiler and P. Atherton, "Hidden Surface Removal Using Polygon Area Sorting," *SIGGRAPH Comput. Graph.*, vol. 11, p. 214–222, jul 1977.

[131] B. R. Vatti, "A Generic Solution to Polygon Clipping," *Commun. ACM*, vol. 35, p. 56–63, jul 1992.

[132] G. Greiner and K. Hormann, "Efficient Clipping of Arbitrary Polygons," *ACM Trans. Graph.*, vol. 17, p. 71–83, apr 1998.

[133] S. Timoshenko and J. Goodier, *Theory of Elasticity.* McGraw-Hill, 3 ed., 1970.

[134] M. Benzi, G. H. Golub, and J. Liesen, "Numerical solution of saddle point problems," *Acta Numerica*, vol. 14, p. 1–137, 2005.

[135] D. M. Neto, M. C. Oliveira, and L. F. Menezes, "Surface Smoothing Procedures in Computational Contact Mechanics," *Archives of Computational Methods in Engineering*, vol. 24, pp. 37–87, 2017.

[136] N. El-Abbasi, S. A. Meguid, and A. Czekanski, "On the modelling of smooth contact surfaces using cubic splines," *International Journal for Numerical Methods in Engineering*, vol. 50, pp. 953–967, 2001.

[137] M. Stadler, G. A. Holzapfel, and J. Korelc, "Cn continuous modelling of smooth contact surfaces using NURBS and application to 2D problems," *International Journal for Numerical Methods in Engineering*, vol. 57, pp. 2177–2203, 2003.

[138] V. Padmanabhan and T. A. Laursen, "A Framework for Development of Surface Smoothing Procedures in Large Deformation Frictional Contact Analysis," *Finite Elem. Anal. Des.*, vol. 37, p. 173–198, mar 2001.

[139] M. A. Puso and T. A. Laursen, "A 3d contact smoothing method using gregory patches," *International Journal for Numerical Methods in Engineering*, vol. 54, no. 8, pp. 1161–1194, 2002.

[140] L. Krstulović-Opara, P. Wriggers, and J. Korelc, "A C1-continuous formulation for 3D finite deformation frictional contact," *Computational Mechanics*, vol. 29, pp. 27–42, 2002.

[141] M. Boschiroli, C. Fünfzig, L. Romani, and G. Albrecht, "Technical Section: A Comparison of Local Parametric C0 Bézier Interpolants for Triangular Meshes," *Computers & Graphics*, vol. 35, p. 20–34, feb 2011.

[142] T. Sekine and T. Obikawa, "Normal-Unit-Vector-Based Tool Path Generation Using a Modified Local Interpolation for Ball-End Milling," *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, vol. 4, no. 7, pp. 1246–1260, 2010.

[143] D. OuYang and H.-Y. Feng, "On the Normal Vector Estimation for Point Cloud Data from Smooth Surfaces," vol. 37, pp. 1071–1079, sep 2005.

[144] D. Neto, M. Oliveira, L. Menezes, and J. Alves, "Applying Nagata patches to smooth discretized surfaces used in 3D frictional contact problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 271, pp. 296–320, 2014.

[145] H. Gouraud, "Continuous Shading of Curved Surfaces," *IEEE Transactions on Computers*, vol. C-20, no. 6, pp. 623–629, 1971.

[146] G. Thürrner and C. A. Wüthrich, "Computing Vertex Normals from Polygonal Facets," *Journal of Graphics Tools*, vol. 3, no. 1, pp. 43–46, 1998.

[147] N. Max, "Weights for Computing Vertex Normals from Facet Normals," *Journal of Graphics Tools*, vol. 4, no. 2, pp. 1–6, 1999.

[148] P.-A. Ubach, C. Estruch, and J. Garcia-Espinosa, "On the interpolation of normal vectors for triangle meshes," *International Journal for Numerical Methods in Engineering*, vol. 96, no. 4, pp. 247–268, 2013.

[149] S. Hartmann, *Kontaktanalyse dünnwandiger Strukturen bei großen Deformationen.* PhD thesis, University of Stuttgart, Institute for Structural Analysis and Dynamics, 2007.

[150] A. Popp, M. W. Gee, and W. A. Wall, "A finite deformation mortar contact formulation using a primal–dual active set strategy," *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1354–1391, 2009.

[151] M. Gitterle, *A dual mortar formulation for finite deformation frictional contact problems including wear and thermal coupling.* PhD thesis, Technische Universität München, 2012.

[152] Code_Aster, "General public licensed structural mechanics finite element software, [v6.01.122] SSNA122 – Benchmark NAFEMS of validation of contact 2: punch (rounded edges)." https://www.code-aster.org/V2/doc/v14/en/man_v/v6/v6.01.122.pdf, 2021. Accessed: October 2021.

[153] T. Sussman and K.-J. Bathe, "A finite element formulation for nonlinear incompressible elastic and inelastic analysis," *Computers and Structures*, vol. 26, no. 1, pp. 357–409, 1987.

[154] K. A. Fischer and P. Wriggers, "Mortar based frictional contact formulation for higher order interpolations using the moving friction cone," *Computer*

Methods in Applied Mechanics and Engineering, vol. 195, no. 37-40, pp. 5020–5036, 2006.

[155] S. Hartmann, J. Oliver, R. Weyler, J. Cante, and J. Hernández, "A contact domain method for large deformation frictional contact problems. part 2: Numerical aspects," *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 33-36, pp. 2607–2631, 2009.

[156] Code_Aster, "General public licensed structural mechanics finite element software, [v6.03.153] SSNP153 — deformable-deformable 2D rubbing contact in large deformations (shallow ironing)." https://www.code-aster.org/V2/doc/default/en/man_v/v6/v6.03.153.pdf, 2020. Accessed: October 2021.

[157] M. Tur, E. Giner, F. J. Fuenmayor, and P. Wriggers, "2d contact smooth formulation based on the mortar method," *Computer Methods in Applied Mechanics and Engineering*, vol. 247, pp. 1–14, 2012.

[158] P. Areias, T. Rabczuk, F. J. Melo, and J. C. Sá, "Coulomb frictional contact by explicit projection in the cone for finite displacement quasi-static problems," *Computational Mechanics*, vol. 55, no. 1, pp. 57–72, 2015.

[159] B. K. Zimmerman and G. A. Ateshian, "A surface-to-surface finite element algorithm for large deformation frictional contact in FEBio," *Journal of Biomechanical Engineering*, vol. 140, no. 8, 2018.

[160] R. A. Sauer and L. De Lorenzis, "An unbiased computational contact formulation for 3D friction," *International Journal for Numerical Methods in Engineering*, vol. 101, no. 4, pp. 251–280, 2015.

[161] V. L. Popov, M. Heß, and E. Willert, *Handbook of contact mechanics: exact solutions of axisymmetric contact problems.* Springer Nature, 2019.

[162] M. Ciavarella, D. A. Hills, and G. Monno, "The influence of rounded edges on indentation by a flat punch," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 212, no. 4, pp. 319–327, 1998.

[163] M. Ciavarella, "Indentation by nominally flat or conical indenters with rounded corners," *International Journal of Solids and Structures*, vol. 36, no. 27, pp. 4149–4181, 1999.

[164] J. Živić, I. Batistić, Ž. Tuković, and I. Karšaj, "Modeling of Incompressible Materials Using Finite Volume Method and its Application in Biomechanics," in *Proceedings of 9nd Congress of Croatian Society of mechanics*, pp. 343–349, 2019.

[165] T. Massé, *Study and optimization of high carbon steel flat wires.* PhD thesis, École Nationale Supérieure des Mines de Paris, 2010.

[166] P. J. Roache, "Quantification of uncertainty in computational fluid dynamics," *Annual Review of Fluid Mechanics*, vol. 29, no. 1, pp. 123–160, 1997.

[167] B. Carlsson, "The contact pressure distribution in flat rolling of wire," *Journal of Materials Processing Technology*, vol. 73, no. 1, pp. 1–6, 1998.

[168] T. Cao, P. Bouchard, and P. Montmitonnet, "Ductile damage prediction in different cold forming processes," in *22ème Congrès Français de Mécanique*, (Lyon, France), Aug. 2015.

[169] C. Erdönmez, "Computational Design of the Compacted Wire Strand Model and Its Behavior Under Axial Elongation," *International Journal of Precision Engineering and Manufacturing*, 2019.

[170] D. Nowell, D. Hills, and A. Sackfield, "Contact of dissimilar elastic cylinders under normal and tangential loading," *Journal of the Mechanics and Physics of Solids*, vol. 36, no. 1, pp. 59–75, 1988.

[171] E. M'Ewen, "XLI. Stresses in elastic cylinders in contact along a generatrix (including the effect of tangential friction)," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 40, no. 303, pp. 454–459, 1949.

[172] Q. J. Wang and D. Zhu, *Hertz Theory: Contact of Cylindrical Surfaces, Encyclopedia of Tribology.* Boston, MA: Springer US, 2013.

[173] M. T. Huber, "Zur Theorie der berührung fester elastischer Körper," *Annalen der Physik*, vol. 319, no. 6, pp. 153–163, 1904.

[174] Q. J. Wang and D. Zhu, *Hertz Theory: Contact of Ellipsoidal Surfaces, Encyclopedia of Tribology*, pp. 1647–1654. Boston, MA: Springer US, 2013.

Bibliography

[175] J. Jäger, "New analytical and numerical results for two-dimensional contact profiles," *International Journal of Solids and Structures*, vol. 39, pp. 959–972, 2002.

Abbreviations

1D One Dimensional

2D Two Dimensional

3D Three Dimensional

AABB Axis-Aligned Bounding Box

BS Bounding Sphere

CAD Computer-Aided Design

CPU Central Processing Unit

DN Dirichlet and Neumann

FA Finite Area

FAM Finite Area Method

FEM Finite Element Method

GGI Generalised Grid Interface

ICCG Incomplete Cholesky Pre-Conditioned Conjugate Gradient

NAFEMS National Agency for Finite Element Methods and Standards

NN Neumann and Neumann

NTN Node to Node

NTS Node to Segment

OBB Oriented Bounding Box

OpenFOAM Open Source Field Operation And Manipulation

PC Personal Computer

SAT Separating Axis Theorem

SIMPLE Semi-Implicit Method for Pressure Linked Equations

STS Segment to Segment

TL Total Lagrangian

TL Updated Lagrangian

Curriculum Vitae

Ivan Batistić was born in Dubrovnik, Croatia, on 25 October 1993. In 2013, he finished high school Blato, the Computer Technician in Mechanical Engineering program. The same year he enrolled in the Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb. During his undergraduate and graduate studies in mechanical engineering, he specialised in Engineering Modelling and Computer Simulations. He finished the undergraduate and graduate programs with CFD thesis topics related to turbomachinery and fluid flow. During his studies, he received multiple awards. In 2018, he finished the graduate program with the highest distinction and immediately started his doctoral studies. His central research focus is computational contact mechanics and solid mechanics. Meanwhile, he is collaborating on various projects, mostly in the area of turbomachinery, heat transfer and mesh generation.

Declaration

Parts of the work presented in this thesis have been published in scientific journals.