Lukas Zachbauer, BSc.

Numerical Strategies for the Transient Simulation of Brake Creep Groan

Master thesis

Aspired academic degree: Diplom-Ingenieur / Master of Science

Master's programme: Mechanical Engineering

Graz University of Technology

Faculty of Mechanical Engineering and Economic Sciences

Institute of Automotive Engineering

Member of Frank Stronach Institute

Head of Institute: Univ.-Prof. Dipl.-Ing. Dr.techn. Peter Fischer

Supervisor: Dipl.-Ing. Severin Huemer-Kals, BSc.

Co-supervisor: Dipl.-Ing. Manuel Pürscher, BSc.

Graz, 18th February 2019

Acknowledgement

First and foremost, I want to thank my supervisors Dipl.-Ing. Severin Huemer-Kals and Dipl.-Ing. Manuel Pürscher for their dedicated support throughout the course of this thesis. Their professional and always friendly advice, as well as their constant availability made this work a most pleasant experience.

Furthermore, I would like to thank all members of the FTG and Univ.Prof. Dipl.-Ing. Dr.techn. Peter Fischer, especially, for giving me this opportunity and providing an enjoyable working environment.

Another huge thank you has to be said to my mother and father who made my studies possible in the first place. Even during the most difficult times I could always rely on their encouragement and unconditional support.

Lastly, a final thank you to my fellow students, friends, and of course my girlfriend for making the past years an exciting journey and a chapter in my life, that I will always gladly look back to. Thank you for being there on every step of the way!

Statutory Declaration

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ....
(Unterschrift)

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

....
(date) (signature)

Abstract

Customer satisfaction has always been a major concern of the automotive industry and many others alike. For vehicles specifically, the topic of "Noise, Vibration, and Harshness" (NVH) has become an increasingly important factor in this matter. Brake related NVH phenomena still cause severe difficulties for many OEMs (Original Equipment Manufacturers), such as squeal, moan, or creep groan. Customers often react to the occurrence of any unwanted noise with complaints and warranty claims.

The prevention of brake NVH is however a complex matter with many influencing factors. Creep groan, especially, can be pointed out as an issue with barely any simulative investigation tools available at present. Currently, minor design iterations, derived from prototype tests, provide the best option for the treatment of creep groan. A simulation based approach has not been successful yet, but could lead to considerable cost reductions if used during the design process already. A promising, but barely explored option is given by direct, transient simulation and will be topic of this work.

The thesis begins with an explanation of the phenomenon creep groan and its mathematical abstraction, followed by an overview of numerical solution strategies (explicit and implicit) for the resulting equation system. Subsequently, the issue of modelling contact interfaces is explored in more detail, as well as the implementation in the nonlinear solver Abaqus.

In the following, the two finite element models used for this thesis are elaborated, including several selected solver settings. The first model is rather simplistic and consists only of brake pads and disk, while the entire axle-corner is considered in the second model. Based on this, a parameter study for the numerical settings and contact models is derived for both models, aiming to find an optimal setting for best performance.

Subsequently, the results of the parameter study are presented, the effects of the respective settings investigated and compared to one another. From these results, an optimal option for both explicit and implicit simulations is determined.

The stick-slip process responsible for creep groan was successfully produced with explicit and implicit solution methods. Comparable results were obtained and both provide a promising simulation tool. However, especially for more complex models the explicit approach showed to lead to significantly faster computing times. On the other hand, the implicit results were found to better represent realistic physical behaviour derived from test rig experiments.

Lastly, recommendations for the transient simulation of creep groan are given based on the findings presented in this thesis.

Kurzfassung

Kundenzufriedenheit und damit das Problem „NVH“ (Noise, Vibration and Harshness) gewinnt immer mehr an Bedeutung für die Automobilindustrie. Vor allem Bremsgeräusche wie Quietschen, Muhen oder Knarzen bereiten nach wie vor große Schwierigkeiten für viele OEMs (Original Equipment Manufacturers). Kundenbeschwerden und Reklamationen sind oft die Folge. Die Vermeidung von Brems-NVH ist jedoch ein komplexes Problem mit einer Vielzahl von Einflussfaktoren. Besonders für Bremsenknarzen gibt es gegenwärtig keine etablierten Simulationsmethoden, diese könnten jedoch zu erheblichen Kosteneinsparungen führen, wenn sie bereits im Entwicklungsprozess angewendet werden. Die transiente, direkte Simulation stellt dabei eine vielversprechende Option dar und ist das Thema dieser Arbeit.

Zu Beginn wird eine Erklärung des Phänomens Bremsenknarzen und dessen mathematischer Modellbildung gegeben, gefolgt von einer Übersicht numerischer Lösungsstrategien (explizit und implizit) für das entstehende Gleichungssystem. Anschließend wird auf die Modellierung von Kontakten und die Implementierung im nichtlinearen Solver Abaqus genauer eingegangen. Es folgt eine Erklärung der beiden Finite Elemente Modelle die für diese Arbeit verwendet wurden, inklusive einiger ausgewählter Solvereinstellungen. Das erste Modell ist eher minimalistisch und besteht aus Bremsbelägen und Bremsscheibe, während das zweite Modell die gesamte Achsaufhängung sowie das Rad berücksichtigt. Basierend auf diesen Modellen wird eine Parameterstudie der numerischen Einstellungen und Kontaktmodelle abgeleitet, mit dem Ziel eine optimale Konfiguration zu finden.

Daran anschließend werden die Ergebnisse dieser Parameterstudie präsentiert, die Auswirkungen der jeweiligen Einstellungen untersucht und miteinander verglichen. Anhand dieser Erkenntnisse wird jeweils eine beste Option für explizite und implizite Simulationen bestimmt.

Der Stick-Slip Effekt, der Bremsenknarzen auslöst, konnte mit expliziten und impliziten Methoden erfolgreich reproduziert werden und vergleichbare Ergebnisse wurden dabei erhalten. Beide Ansätze stellen daher ein vielversprechendes Werkzeug für Knarzsimulationen dar. Es stellte sich jedoch heraus, dass besonders für komplexere Modelle die explizite Variante erheblich schnellere Rechenzeiten erzielt. Dem gegenüber wurde festgestellt, dass die impliziten Ergebnisse besser mit dem von Prüfstandsuntersuchungen gewonnenen Verhalten zusammenpassen.

Abschließend werden Empfehlungen für die transiente Simulation von Bremsenknarzen gegeben, basierend auf den Erkenntnissen, die im Zuge dieser Arbeit gewonnen wurden.

Contents

Abbreviations

ACM	Axle-corner Model
BWE	Backwards Euler time integrator
CEA	Complex Eigenvalue Analysis
CLOAD	Concentrated load
CSTATUS	Contact status
DLOAD	Distributed load
DOF	Degree of freedom
DPM	Disk Pad Model
FE	Finite Element
FEA	Finite Element Analysis
HHT-α	Hilber-Hughes-Taylor method
HHT-MD	Hilber-Hughes-Taylor time integrator for moderate dissipation
HHT-TF	Hilber-Hughes-Taylor time integrator for transient fidelity
ISO	International Organization for Standardization
MD	Moderate dissipation
NoE	Number of elements
NVH	Noise, Vibration and Harshness
OEM	Original Equipment Manufacturer
RMS	Root mean square
SDI	Severely discontinuous iteration
SIM	Surface interaction model
TF	Transient fidelity

Augm.	Augmented
Behav.	Behaviour
Equ.	Equation
Integr.	Integrator
Lag.	Lagrange
lat.	lateral
lin.	linear
longitud.	longitudinal
max.	maximum
min.	minimum
nonlin.	nonlinear
vert.	vertical

Symbols

Matrices and Tensors

$\boldsymbol{M}$	mass matrix
$\boldsymbol{D}$	damping matrix
$\boldsymbol{K}$	stiffness matrix
$\boldsymbol{E}$, $\delta\boldsymbol{E}$	Green-Lagrange stress tensor and its variation
$\boldsymbol{S}$	Kirchhoff stress tensor
$\boldsymbol{\sigma}$	stress tensor
$\boldsymbol{P}$	projection matrix

Parameters and Variables

α	parameter of HHT-α method
β , γ	parameters of Newmark-β method
β_s	scale factor
γ_crit	maximum allowed elastic slip
$\dot{\gamma}$	slip rate
$\dot{\gamma}_\mathrm{eq}$	equivalent slip rate
Γ	boundary
Γ_σ	boundary with prescribed surface loads
Γ_c	boundary with contact interaction
Γ_u	boundary with prescribed displacements
ϵ_N , ϵ_T	penalty parameter for normal and tangential direction
θ	temperature
λ_N , $\delta\lambda_\mathrm{N}$	Lagrange multiplier for normal direction and its variation
μ	coefficient of friction
μ_k	kinetic coefficient of friction
μ_s	static coefficient of friction
ν	Poisson's ratio
ρ	density
ρ_0	initial density
φ	rotation
φ_i	element angles
$\varphi(X)$	push-forward operator of a quantity, e.g. X

$\omega_{\max}$	maximum frequency in the system
a , A	area
AR	aspect ratio
B	body
c	clearance
c_{d}	dilatational wave speed
C_c	contact contribution
d	damping coefficient
d_{c}	decay coefficient
E	Young's modulus
$\overline{f}^{\alpha}$	field variable
f_{SS}	stick-slip frequency
F	force
F_{d}	damping force
F_{F}	friction force
F_{k}	stiffness force
F_{N}	normal force
g	gap
g_{N} , δg_{N}	gap in normal direction and its variation
g_{t}	tangential gap
G	shear modulus
$G(\varphi, \boldsymbol{\eta})$, $g(\varphi, \boldsymbol{\eta})$	weak form of balance of momentum in initial and current configuration
h	overclosure
h_i	element lengths
I	inertia
k	stiffness coefficient
K	slope
l	length
L_{elem}	element length
$L_{\min}$	smallest element dimension in the mesh
m	mass
M	moment, torque
M_{F}	friction moment
$M_{\mathrm{F,max}}$	maximum friction moment
n	discrete number
n_{inc}	number of increments
p	pressure
p_{N}	contact pressure
p_{p}	pad pressure
r	overclosure factor
s	scale factor

t	time
t_n	discrete point in time
Δt	time step, time increment
T	total time period
u_N	displacement in normal direction
v , V	volume
x	displacement
$\dot{x}$	velocity
$\ddot{x}$	acceleration

Vectors

$\boldsymbol{\eta}$	virtual displacement or test function
$\boldsymbol{\lambda}_\mathrm{T}$, $\delta\boldsymbol{\lambda}_\mathrm{T}$	Lagrange multiplier for tangential direction and its variant
$\rho_0 \cdot \overline{\boldsymbol{b}}$	body force
$\rho_0 \cdot \overline{\dot{\boldsymbol{v}}}$	inertia force
$\boldsymbol{\tau}$	Kirchhoff stress
$\boldsymbol{\Psi}$	material flow
$\boldsymbol{a}_n$	discrete acceleration at point in time t_n
$\dot{\boldsymbol{a}}_n$	discrete jerk at point in time t_n
$\boldsymbol{e}$	unit vector
$\boldsymbol{f}$	external forces
$\boldsymbol{f}_n$	discrete external force at point in time t_n
$\boldsymbol{g}_\mathrm{T}$, $\delta\boldsymbol{g}_\mathrm{T}$	relative tangential displacement and its variant
$\dot{\boldsymbol{g}}_\mathrm{T}$	relative tangential velocity
$\boldsymbol{n}$	normal vector
$\overline{\boldsymbol{t}}$	described (external) stress
$\boldsymbol{t}_\mathrm{n}$	normal stress
$\boldsymbol{t}_\mathrm{t}$	tangential shear stress
$\boldsymbol{t}_\mathrm{t,max}$	maximum tangential shear stress
$\boldsymbol{u}$	displacement
$\dot{\boldsymbol{u}}$	velocity
$\ddot{\boldsymbol{u}}$	acceleration
$\boldsymbol{u}_n$	discrete displacement at point in time t_n
$\boldsymbol{v}_n$	discrete velocity at point in time t_n
$\boldsymbol{v}^\mathrm{back}$	backwards difference quotient for velocity
$\boldsymbol{v}^\mathrm{forw}$	forwards difference quotient for velocity
$\boldsymbol{x}$	displacement
$\dot{\boldsymbol{x}}$	velocity
$\ddot{\boldsymbol{x}}$	acceleration

1. Introduction

1.1. Motivation

Ever since its invention in the late 19th century, the automobile has been subject to a continues optimization process. The necessity of avoiding any unwanted noise emissions was recognized in early stages already. The collective term "Noise, Vibration and Harshness", or NVH, was coined by the automotive industry for this issue.
Today, after more than 100 years of research, the traditional NVH sources are well understood and cars have become increasingly quiet. As a result, costumer expectations have developed as well, making quietness and comfort leading selling points.
Chassis and running gear have become one of the most relevant contributors to noise emission, which are met with great displeasure by customers. Namely, brake NVH related warranty claims have seen a significant increase, elevating the issue to a major concern of Original Equipment Manufacturers (OEMs) worldwide, [17]. Abendroth and Werbitz claim that NVH accounts for nearly 50% of the expenses at companies producing friction materials, putting it on par with efforts of performance, [3].
Brake squeal, the most prominent brake NVH phenomenon, has been subject to research for decades and is well understood today. As a result, a wealth of literature is available on this topic.
Less attention was given to creep groan, a self-excited, low-frequency brake noise, [7]. The mechanism responsible for its occurrence was identified as a stick-slip effect between brake disk and pads. With increasing popularity of automatic transmissions, creep groan has moved closer to the centre of attention. Near standstill, the idle torque from the transmission in combination with low brake pressure is the most common cause of creep groan, [2]. Consequently, the prevention of its occurrence has become a primary interest of many OEMs.
Experimental investigation is the traditional method of dealing with NVH problems; which is, however, costly and leads to long turnaround times for design iterations, [19]. Furthermore, a vast variety of influence factors on creep groan behaviour have been identified, making a precise assessment of cause and effect an extraordinarily challenging task. Computer simulations may provide a powerful tool for the analysis and prevention of creep groan and can lead to a better understanding of the phenomenon. Design modifications can be implemented easily with simulations, such as material properties, stiffness and damping, operating conditions, etc. Since test rig alterations are usually rather intricate, promising improvement measures are best derived from simulations and subsequently investigated experimentally. For reasonable results, a carefully understood numerical model is, therefore, indispensable.

For simulations of dynamic problems, two general approaches are commonly used: Complex Eigenvalue Analysis (CEA) in the frequency domain, and transient simulations in the time domain. Nunes, Shivaswamy and Könning suggest in [18] that CEA is insufficient for the highly nonlinear stick-slip effect that causes creep groan. Transient, dynamic simulations with nonlinear solvers provide a better option, [18].
This thesis will thus aim to contribute to the development of a sophisticated transient simulation approach for creep groan. The long-term goal will be an FE model capable of predicting influences on creep groan behaviour that can be validated via test rig experiments. Valuable clues for the prevention of this increasingly important brake NVH phenomenon are hoped for.

1.2. Scientific approach and structure

Firstly, the reader is provided with a general understanding of the phenomenon creep groan and the underlying stick-slip effect. A mathematical abstraction is derived, followed by an overview of numerical strategies for its solution, using direct time integration in Finite Element Analysis (FEA). Due to its significance for creep groan simulations, key aspects of contact mechanics in FEA are elaborated, with focus on the treatment of occurring overconstraint issues and discontinuities imposed by contacts. Methods for modelling contact interactions in normal and tangential direction are presented. Chapter 2 closes with the implementation of aforementioned in the nonlinear solver Abaqus.

In Chapter 3, the most significant parameters of two FE models, that were used for this work, are explained. This includes general model properties as well as solution approaches and solver settings for explicit and implicit simulations. An overview of modifications included in the subsequently performed parameter study is given, followed by a description of test rig experiments that provide the basis and means of validation for the numerical models.

The results of aforementioned parameter studies on both models are presented in Chapter 4. The main focus are the obtained computing times and stick-slip frequencies, alongside with additional findings that are included as well. A comparison of the simulated variants is given and effects of modifications are explored.

In Chapter 5, an optimal model configuration for both explicit and implicit transient simulations is sought. Differences of the results obtained with both approaches are investigated.

Lastly, a final conclusion for the transient simulation of creep groan is drawn in Chapter 6, closing with recommendations for future investigations of this topic.

2. State of the Art

In this chapter the basic mechanisms that lead to the occurrence of creep groan on disk brakes will be explained, as well as the basics of modelling and solving nonlinear dynamic problems using FEA. An overview of explicit and implicit numerical schemes will be given, followed by a more detailed elaboration of contact modelling in FEA. Lastly, the implementation in the nonlinear solver Abaqus will be explained.

2.1. Brake - NVH

There are many different types of noise phenomena that can be associated with disk brake systems and are commonly referred to under the general topic "Noise, Vibration and Harshness", or NVH. In this thesis, NVH will be limited to automotive disk brake systems, where this issue is of significant relevance. NVH is generally unpleasant for the costumer and decreases the comfort and overall satisfaction with the product, therefore manufacturers are trying to avoid brake noise as far as possible. Table 2.1 provides an overview of the most common brake noises, their respective frequency range and overall rank of importance for manufacturers, according to [8].

Since the main focus of this work is creep groan, a more detailed explanation of the phenomenon will be provided. The characteristic frequency range of creep groan may differ depending on the literature used.

Table 2.1.: Overview of typical brake noise phenomena, adapted from [8]

Rank	Name	Frequency
1	Judder	10 - 200 Hz
2	Squeal	1 - 20 kHz
3	Creep groan	50 - 200 Hz
4	Moan	200 - 500 Hz
5	Howl	500 - 1000 Hz
6	Rattle	50 - 1000 Hz

2.1.1. Creep groan

Creep groan is a brake noise phenomenon in the lower frequency range. It is generally caused by the slow opening of a disk brake, that is under torsional stress.
A simple abstraction of a disk brake, reduced to just the brake disk and the brake caliper, and the process that causes creep groan is illustrated in Figure 2.1. At first the full brake force F shall be applied without any torsional moment being present yet. The resulting brake pressure is at its maximum. The torque (moment) M is now applied to the brake disk which leads to a resulting frictional (brake) moment M_{F} in opposite direction of M in the contact interface between brake disk and pads. The maximum magnitude of the frictional moment $M_{\mathrm{F,max}}$ is correlated to the applied brake pressure and the coefficient of friction in the contact.
In this stage, the applied moment M is lower than $M_{\mathrm{F,max}}$, and M and M_{F} are in equilibrium. Static friction is holding the disk in place; the brake is closed. If now the magnitude of M is increased until $M > M_{\mathrm{F,max}}$, kinetic friction occurs. Since the kinetic friction coefficient is usually lower than the static friction coefficient, so is also the frictional moment M_{F} in this stage. Therefore, M and M_{F} are no longer in static equilibrium and the disk starts to slip. The previously higher frictional moment M_{F} present in the stick phase has caused the brake caliper and its auxiliary components to deform elastically. Thus, tension was built up in the system. In the slip phase, M_{F} is lower - a part of this tension is released which causes the caliper and pads to undergo a slight displacement in opposite direction of the disk rotation.
For specific combinations of magnitudes of the applied moment M and the brake force F, the kinetic friction can transition again into static friction after a short slip period. Now the process can start anew until once again M exceeds $M_{\mathrm{F,max}}$. An oscillating motion between brake disk and brake pads is the result. This phenomenon is known as the "stick-slip effect".

The transitions from stick to slip are very abrupt and lead thus to sudden spikes in the mechanical forces in the system. This can occur many times per second and therefore invoke vibrations in the brake pads and disk. These vibrations then spread to other connected components of the axle and suspension and eventually to the car body and passenger compartment, where it results in an audible noise, called "creep groan" [8]. The same effect can be achieved, if instead of increasing the applied moment M, the brake force F is decreased. It is the combination of the two parameters that matters.

Circumstances of this sort typically occur when cars with automatic transmission are accelerated from standstill, or upon slow brake release on inclined roads. With a typical frequency range of 50-200 Hz creep groan is considered a self-excited, low-frequency NVH problem.

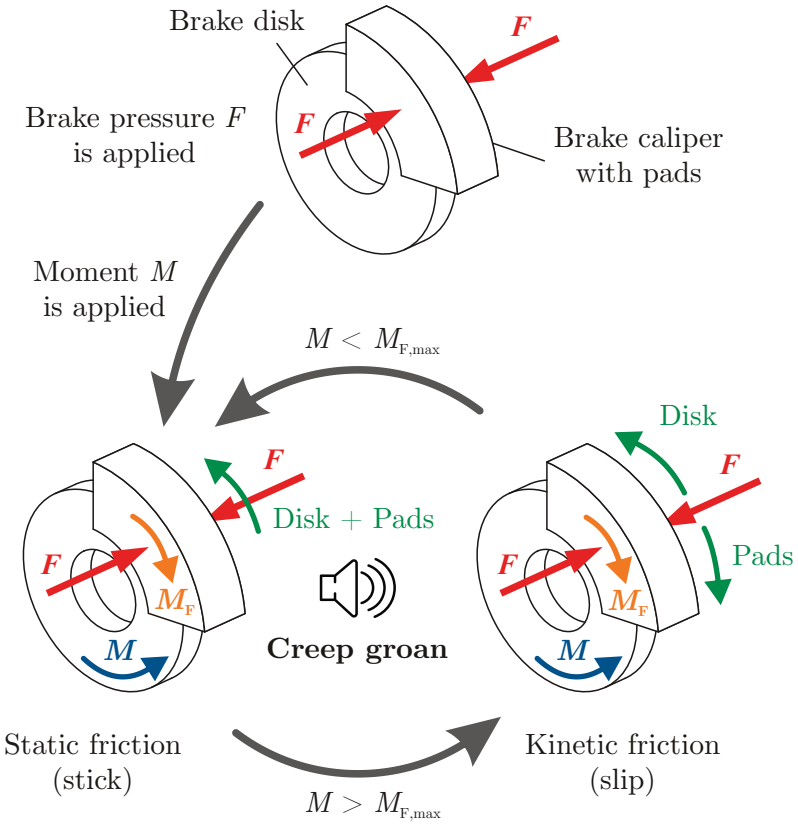

Figure 2.1.: Stick-slip mechanism that causes creep groan

2.1.2. Frequency spectrum

In [20], experimental investigations of creep groan have been conducted on a vehicle corner test rig. A typical time data plot of creep groan is shown in Figure 2.2. The top subfigure shows the signal of the acceleration sensor on the brake carrier over time, the bottom subfigure the Fourier transformed signal. A more detailed explanation of the setup for the experiment and the configurations used to achieve creep groan can be found in [20].

The upper sub-figure shows that periodically repeating events occur with a frequency of approximately 90 Hz, which is a clear indicator of the stick-slip effect. It is also easy to see that accelerations become both positive and negative throughout the measurement, the negative ones show a significantly larger amplitude. At those negative peaks, the change from sticking to slipping occurs. The lower sub-figure reveals a distinct peak at 87 Hz, which matches the estimated frequency information gained from the upper plot and represents the stick-slip frequency with this test configuration. Furthermore, additional peaks are shown at multiples of the basic frequency and can be interpreted as super-harmonics. In comparison, a second graph is plotted in the lower sub-figure

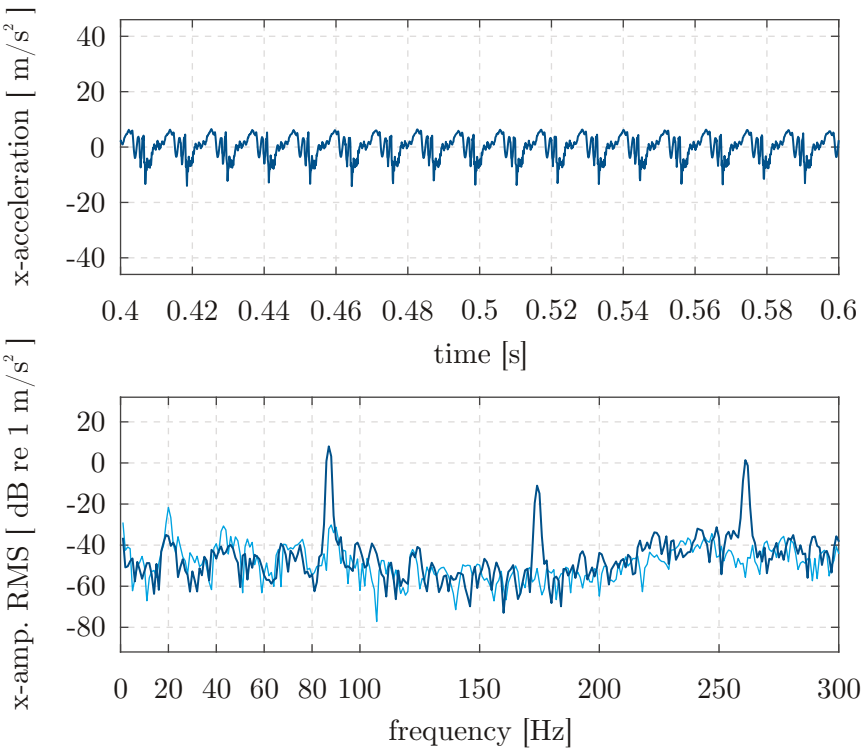

Figure 2.2.: Brake carrier x-vibration / 87 Hz creep groan / 8 bar / 0.12 km/h, adapted from [20]

which represents the acceleration signal of a test configuration, at which no creep groan occurred. No distinct peaks are present in this signal, it has an overall lower amplitude, and therefore less noise is being emitted [20].

2.2. Finite Element Analysis

For this chapter, a basic understanding of FEA principles will be presumed. However, an overview about a few specific topics, that are necessary in order to understand the following chapters of this work, will be presented and certain crucial parts will be elaborated in more detail.

As already mentioned in Chapter 1, a transient analysis of creep groan using an implicit time integration scheme was conducted and compared to previous results from equivalent analyses, using an explicit time integration scheme, conducted in [20]. Furthermore, different methods of modelling contacts in FEA were explored, since they have shown to have a significant influence on the results as well as the overall performance of the analysis.

Therefore, an understanding of the following topics will be necessary:

- Mathematical model
- Explicit time integration schemes
- Implicit time integration schemes
- Contact mechanics in FEA

Lastly in this chapter, the implementation and a few unique properties of the nonlinear solver Abaqus, that was used for this work, will be explained.

2.2.1. Mathematical model

The basic mechanics relevant for the mathematical modelling of a stick-slip contact will be explained according to [7]. The model has to include several basic features to be able to represent the main causes of creep groan on disk brakes, such as the brake pads and disk, the contact between these two, as well as a way of applying an external force or moment. A triple-mass oscillator is a simple abstraction that is capable of representing the aforementioned features and is depicted in Figure 2.3. It represents a simplified model of the contact between disk and pad and was transformed from a radial arrangement, as would be the case on a disk brake, to a translational replacement model for the sake of simplicity.

The equation of motion for the triple-mass oscillator in Figure 2.3 is

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{D}\dot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{x} = \boldsymbol{f} \tag{2.1}$$

with the mass matrix $\boldsymbol{M}$, the damping matrix $\boldsymbol{D}$, the stiffness matrix $\boldsymbol{K}$, the vector of external forces $\boldsymbol{f}$ and the vectors of displacement, velocity and acceleration $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$ and $\ddot{\boldsymbol{x}}$.

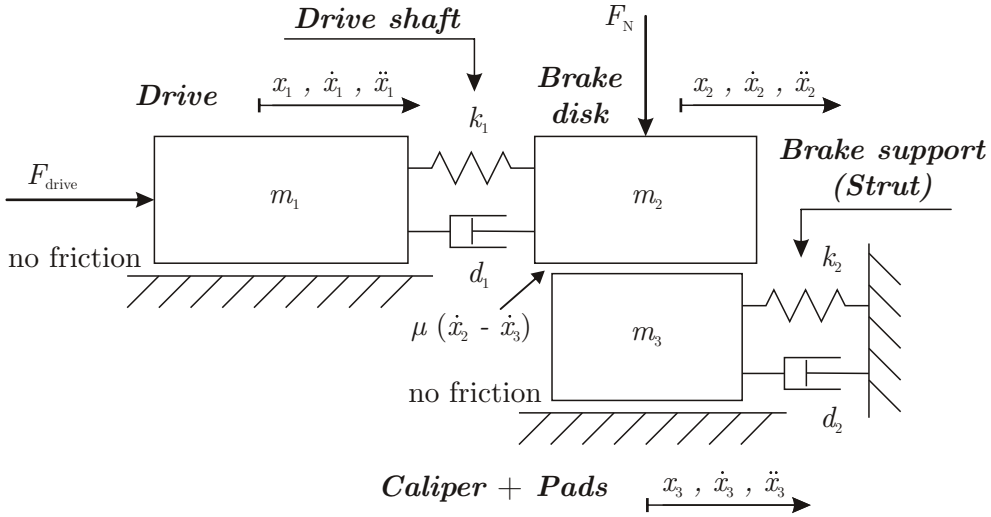

Figure 2.3.: Triple-mass oscillator for the system drive, disk, brake for a lifted vehicle with detached wheel, adapted from [7]

Since the stick-slip behaviour is the main interest, a distinction between these two states has to be made. The two different free body diagrams for slipping and sticking are shown in Figure 2.4 and 2.5 respectively. The occurring forces as well as the resulting equation of motion will be derived for each case.

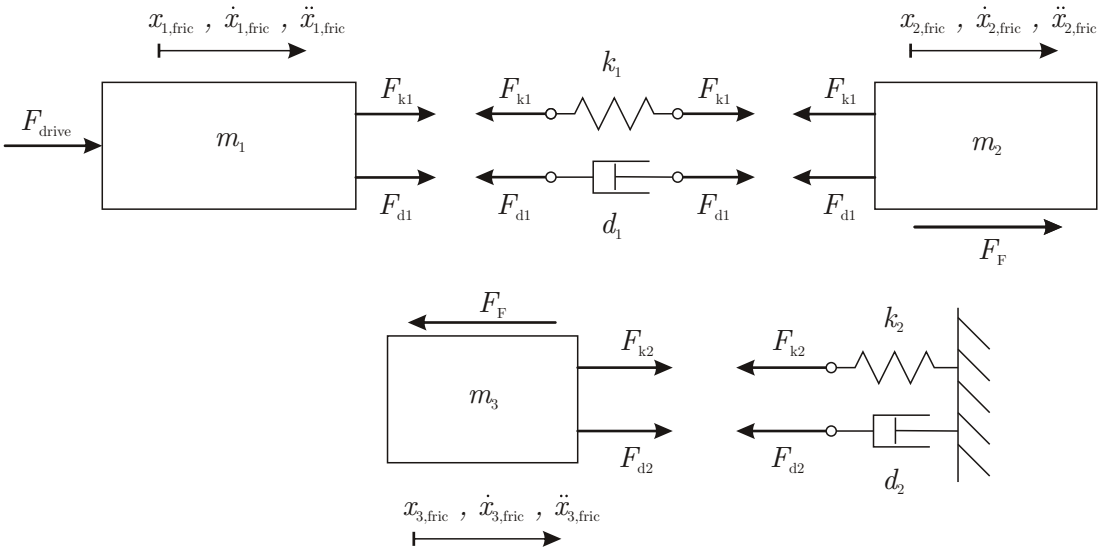

Figure 2.4.: Free body diagram of the model during slipping phase, adapted from [7]

With

$$F_{\mathrm{k1}} = k_1 \left(x_{2,\mathrm{slip}} - x_{1,\mathrm{slip}}\right), \tag{2.2}$$
$$F_{\mathrm{d1}} = d_1 \left(\dot{x}_{2,\mathrm{slip}} - \dot{x}_{1,\mathrm{slip}}\right), \tag{2.3}$$
$$F_{\mathrm{k2}} = -k_2\, x_{3,\mathrm{slip}}\,, \tag{2.4}$$
$$F_{\mathrm{d2}} = -d_2\, \dot{x}_{3,\mathrm{slip}}\,, \tag{2.5}$$
$$F_{\mathrm{F}} = -\mu F_{\mathrm{N}} \quad ; \quad \mu(\dot{x}_{2,\mathrm{slip}} - \dot{x}_{3,\mathrm{slip}})\,, \tag{2.6}$$

the equation of motion for the case of slipping becomes

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \ddot{x}_{1,\mathrm{slip}} \\ \ddot{x}_{2,\mathrm{slip}} \\ \ddot{x}_{3,\mathrm{slip}} \end{bmatrix} + \begin{bmatrix} d_1 & -d_1 & 0 \\ -d_1 & d_1 & 0 \\ 0 & 0 & d_2 \end{bmatrix} \begin{bmatrix} \dot{x}_{1,\mathrm{slip}} \\ \dot{x}_{2,\mathrm{slip}} \\ \dot{x}_{3,\mathrm{slip}} \end{bmatrix} +$$

$$+ \begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 & 0 \\ 0 & 0 & k_2 \end{bmatrix} \begin{bmatrix} x_{1,\mathrm{slip}} \\ x_{2,\mathrm{slip}} \\ x_{3,\mathrm{slip}} \end{bmatrix} = \begin{bmatrix} F_{\mathrm{drive}} \\ -F_{\mathrm{F}} \\ F_{\mathrm{F}} \end{bmatrix}\,. \tag{2.7}$$

For the case of sticking, the bodies 2 and 3 can be considered as one single body and the free body diagram becomes

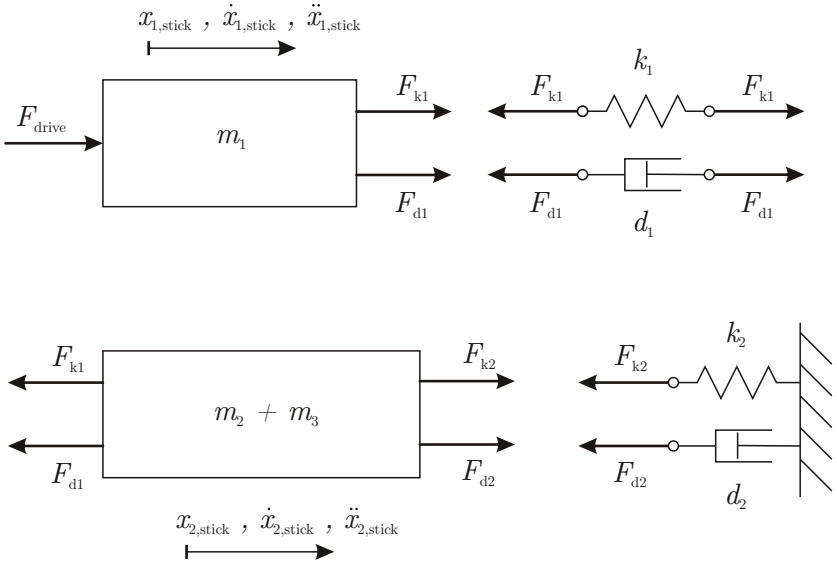

Figure 2.5.: Free body diagram of the model during sticking phase, adapted from [7]

With

$$F_{\mathrm{k1}} = k_1 \left(x_{2,\mathrm{stick}} - x_{1,\mathrm{stick}} \right), \tag{2.8}$$

$$F_{\mathrm{d1}} = d_1 \left(\dot{x}_{2,\mathrm{stick}} - \dot{x}_{1,\mathrm{stick}} \right), \tag{2.9}$$

$$F_{\mathrm{k2}} = -k_2 \, x_{2,\mathrm{stick}}, \tag{2.10}$$

$$F_{\mathrm{d2}} = -d_2 \, \dot{x}_{2,\mathrm{stick}}, \tag{2.11}$$

the equation of motion for the case of sticking becomes

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 + m_3 \end{bmatrix} \begin{bmatrix} \ddot{x}_{1,\mathrm{stick}} \\ \ddot{x}_{2,\mathrm{stick}} \end{bmatrix} + \begin{bmatrix} d_1 & -d_1 \\ -d_1 & d_1 + d_2 \end{bmatrix} \begin{bmatrix} \dot{x}_{1,\mathrm{stick}} \\ \dot{x}_{2,\mathrm{stick}} \end{bmatrix} + \\ + \begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 + k_2 \end{bmatrix} \begin{bmatrix} x_{1,\mathrm{stick}} \\ x_{2,\mathrm{stick}} \end{bmatrix} = \begin{bmatrix} F_{\mathrm{drive}} \\ 0 \end{bmatrix}. \tag{2.12}$$

As one can see, the situations for each case, Figure 2.4 (slip) or 2.5 (stick), are vastly different and, as a result, so are the respective equations of motion Equ. (2.7) and (2.12). This leads to serious difficulties when trying to simulate problems where constant changes between these two states occur, and subsequently the parameters and equations describing the systems change with it every time. Frequent stick-slip changes impose strong nonlinear effects and discontinuities which are difficult for numerical solvers to handle.
As a result, the simulation of stick-slip behaviour, such as creep groan, is very challenging and the model has to be built carefully in order to achieve convergence. Depending on the model size, the simple triple-mass oscillator in Figure 2.3 with three degrees of freedom (DOF) can easily evolve into a system with many thousand DOFs and a tremendous amount of equations that need to be solved which can result in extremely long computing times.

2.2.2. Numerical solutions

Generally, there are two different methods that can be used to predict and simulate brake noise numerically. One is a linear stability analysis, such as CEA in the frequency domain, e.g. as it was conducted in [14]. The second option is a dynamic transient analysis in the time domain which was used for this thesis.
Both have their unique advantages and disadvantages and have been analysed in different publications, e.g. [5], [16]. Combinations of both methods, that utilize both the frequency domain and the time domain, have been developed and are commonly referred to as “co-simulation”, e.g. [10]. The model is hereby split into two sections: an implicit one solved in the frequency domain, and an explicit section solved in the time domain, [10].

An additional third approach, the normal modes method, is suggested in [16], where the "normal modes with modal locking/kinematic constraint theory" are investigated. [16] also provides a comparison to the aforementioned. The transient method is advocated as the best comparison with experimental data, and the normal modes method as the most efficient means of deriving design changes. Generally, a combined application is recommended for optimal results.
CEA is the traditional approach to predict brake squeal noise, and allows for all unstable frequencies to be found in one run for one set of parameters. This option is therefore very fast and efficient. However, not all of these frequencies can be found in experiments and many nonlinear effects cannot be consider sufficiently with this linearized method. CEA is thus unsuitable for the investigation of creep groan, as it was also found in [14]. This method is however commonly used for squeal simulations.
Transient analysis is capable of considering most nonlinear effects and allows for a very detailed and accurate modelling of the problem. As a result, true unstable frequencies, that can also be validated in experiments, can be found with this method if the model is built correctly, making it a promising strategy for creep groan simulations. The disadvantages are much higher computational costs than CEA, [5].
Numerically, a dynamic transient analysis is solved iteratively, using a direct time integration scheme, e.g. provided by the nonlinear solver Abaqus, which will be explained in the next chapter.

2.2.3. Direct time integration

The basic methods of direct time integration will be explained, relying mostly on [25]. The spatial discretization using the finite element method transforms a set of partial differential equations into a set of ordinary differential equations, that is solely time-dependent and can now be solved using direct time integration. The method will be shown using the basic equation of motion

$$\boldsymbol{M}\ddot{\boldsymbol{u}}(t) + \boldsymbol{K}\boldsymbol{u}(t) = \boldsymbol{f}(t) \tag{2.13}$$

with the mass matrix $\boldsymbol{M}$, the stiffness matrix $\boldsymbol{K}$, the vectors of acceleration $\ddot{\boldsymbol{u}}$, displacement $\boldsymbol{u}$ and external forces $\boldsymbol{f}$, and the time t. Damping is left out here for the sake of simplicity. Equ. (2.13) is solved for a discrete point in time t_n and the solution for the next point in time $t_{n+1} = t_n + \Delta t$ after the time increment Δt has passed, is calculated. This is done based on an assumption of the chronological sequences of the parameters during this increment. The approximations of acceleration, velocity, displacement and forces at t_n will be written as $\boldsymbol{a}_n$ for $\ddot{\boldsymbol{u}}(t_n)$, $\boldsymbol{v}_n$ for $\dot{\boldsymbol{u}}(t_n)$, $\boldsymbol{u}_n$ for $\boldsymbol{u}(t_n)$, and $\boldsymbol{f}_n$ for $\boldsymbol{f}(t_n)$. Based on the nature of these assumptions, a distinction in two different classes of integration schemes can be made:

- Explicit time integration
  The equation of motion is solved for the current time t_n as

$$\boldsymbol{M}\boldsymbol{a}_n + \boldsymbol{K}\boldsymbol{u}_n = \boldsymbol{f}_n \tag{2.14}$$

  and the parameters at the following time t_{n+1} are found using extrapolation.

- Implicit time integration
  The equation of motion is solved for the next point in time in the future, where the parameters are currently still unknown, as

$$\boldsymbol{M}\boldsymbol{a}_{n+1} + \boldsymbol{K}\boldsymbol{u}_{n+1} = \boldsymbol{f}_{n+1} \; . \tag{2.15}$$

  Generally, this cannot be done directly in one step, but instead by utilizing an iteration scheme, usually the Newton-Raphson method, until a satisfying solution is found.

2.2.3.1. Explicit method

Explicit time integration is typically performed by using the central difference scheme. It is based on the assumption, that both displacement and velocity change linearly during one time increment. Based on this, the forward and backward difference quotient for velocity ($\boldsymbol{v}_n^{\text{forw}}$ and $\boldsymbol{v}_n^{\text{back}}$) are defined as the slope of the displacement between the times t_{n-1}, t_n and t_{n+1}, see Figure 2.6, [25].

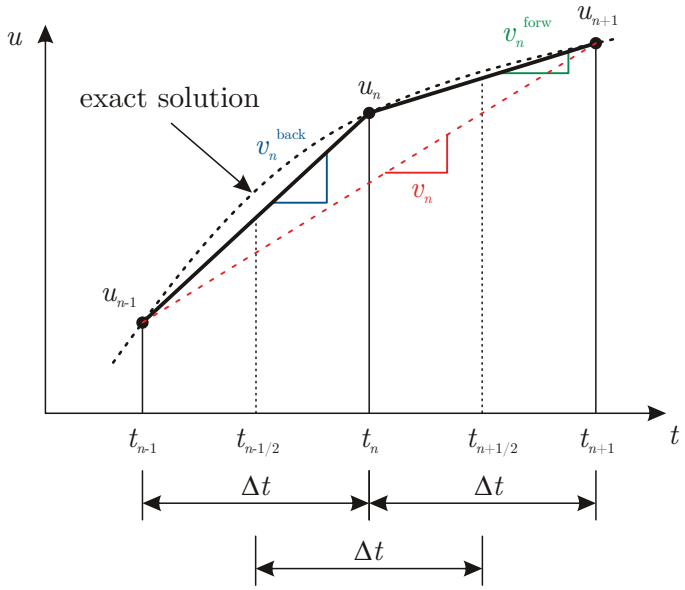

Figure 2.6.: Definition of the difference quotient: forward, backward and central difference quotient with constant time step, adapted from [25]

With

$$\boldsymbol{v}_n^{\text{forw}} = \boldsymbol{v}_{n+1/2} = \frac{\boldsymbol{u}_{n+1} - \boldsymbol{u}_n}{\Delta t} \,, \tag{2.16}$$

$$\boldsymbol{v}_n^{\text{back}} = \boldsymbol{v}_{n-1/2} = \frac{\boldsymbol{u}_n - \boldsymbol{u}_{n-1}}{\Delta t} \,, \tag{2.17}$$

the central difference quotient for the velocity $\boldsymbol{v}_n$ is then gained as the mean value of the forward and backward difference as

$$\boldsymbol{v}_n = \frac{1}{2}(\boldsymbol{v}_n^{\text{forw}} + \boldsymbol{v}_n^{\text{back}}) = \frac{\boldsymbol{u}_{n+1} - \boldsymbol{u}_{n-1}}{2\Delta t} \,. \tag{2.18}$$

The acceleration is calculated analogously by assuming the velocity changes linearly. The acceleration is interpreted as its slope and we obtain

$$\boldsymbol{a}_n = \frac{\boldsymbol{v}_n^{\text{forw}} - \boldsymbol{v}_n^{\text{back}}}{\Delta t} = \frac{\boldsymbol{v}_{n+1/2} - \boldsymbol{v}_{n-1/2}}{\Delta t} \,. \tag{2.19}$$

Assuming a linear course for both velocity and acceleration is a physically impossible approximation, thus the extrapolation of the next point in time is not perfectly accurate. Therefore the explicit integration scheme is only conditionally stable, the size of the time increment has to be chosen very small and many time steps are necessary for a simulation.

The maximum time increment allowed for stability of the scheme is defined by the Courant-Friedrichs-Lewy condition as the magnitude of the highest frequency of the system as [1]

$$\Delta t \leq \frac{2}{\omega_{\max}} \,. \tag{2.20}$$

It can also be approximated as the smallest transit time of a dilatational wave across any of the elements in the finite element mesh

$$\Delta t \approx \frac{L_{\min}}{c_{\text{d}}} \tag{2.21}$$

with the smallest element dimension in the mesh $L_{\min}$ and the dilatational wave speed c_{d}, which are calculated by Abaqus. c_{d} is determined based on material properties, such as density and stiffness, while $L_{\min}$ is defined via the coarseness of the mesh. When building a model for explicit simulations, one has to keep this relationship in mind. Fine meshes with short element lengths in combination with stiff, or high density materials will lead to a very small time increment and thus to long computing times.
The explicit method is very efficient for problems where the mass matrix can be approximated by a diagonal, lumped mass matrix.

2.2.3.2. Implicit method

Implicit time integration often uses the Newmark-β method or a variation of it. In this scheme, the acceleration is assumed to be constant between two points in time and equal to the mean value of the values at those times, see Figure 2.7.

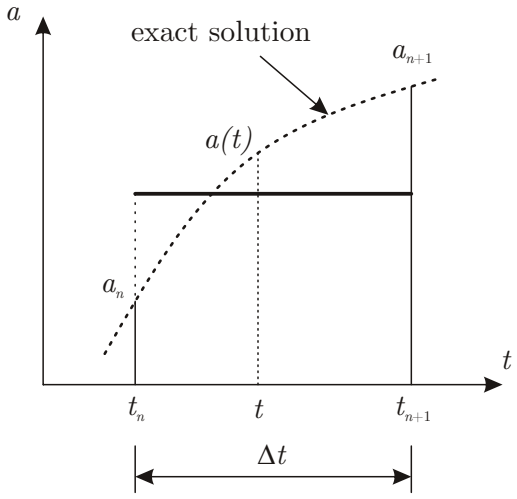

Figure 2.7.: Time integration based on the assumption of constant acceleration, adapted from [25]

We obtain

$$\boldsymbol{a}(t) = \frac{1}{2}\left(\boldsymbol{a}_{n+1} + \boldsymbol{a}_n\right) . \tag{2.22}$$

Using this, the velocity is given as

$$\boldsymbol{v}(t) = \boldsymbol{v}_n + \int_{t_n}^{t} \boldsymbol{a}\mathrm{d}\hat{t} = \boldsymbol{v}_n + \frac{1}{2}(\boldsymbol{a}_{n+1} + \boldsymbol{a}_n)(t - t_n) \tag{2.23}$$

and the displacement as

$$\boldsymbol{u}(t) = \boldsymbol{u}_n + \int_{t_n}^{t} (\boldsymbol{v}_n + \frac{1}{2}(\boldsymbol{a}_{n+1} + \boldsymbol{a}_n)(\hat{t} - t_n))\mathrm{d}\hat{t} \tag{2.24}$$

$$= \boldsymbol{u}_n + \left[\boldsymbol{v}_n(t - t_n) + \frac{1}{4}(\boldsymbol{a}_{n+1} + \boldsymbol{a}_n)(\hat{t} - t_n)^2\right]_{t_n}^{t} . \tag{2.25}$$

For t_{n+1}, this can now be approximated using the Taylor series

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \boldsymbol{v}_n\Delta t + \boldsymbol{a}_n\frac{\Delta t^2}{2} + \dot{\boldsymbol{a}}_n\frac{\Delta t^3}{6} + \dots , \tag{2.26}$$

$$\boldsymbol{v}_{n+1} = \boldsymbol{v}_n + \boldsymbol{a}_n\Delta t + \dot{\boldsymbol{a}}_n\frac{\Delta t^2}{2} + \dots . \tag{2.27}$$

Only terms until the third time derivative, which is the derivative of the acceleration, or jerk, are being considered. Higher order terms are left out. To compensate for this error, two new parameters β and γ are introduced and the approximation becomes

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \boldsymbol{v}_n \Delta t + \boldsymbol{a}_n \frac{\Delta t^2}{2} + \beta \dot{\boldsymbol{a}}_n \Delta t^3 + \dots \,, \tag{2.28}$$

$$\boldsymbol{v}_{n+1} = \boldsymbol{v}_n + \boldsymbol{a}_n \Delta t + \gamma \dot{\boldsymbol{a}}_n \Delta t^2 + \dots \,. \tag{2.29}$$

Furthermore it is assumed, that the acceleration during one time interval changes linearly and thus the jerk is constant:

$$\dot{\boldsymbol{a}}_n = \frac{\boldsymbol{a}_{n+1} - \boldsymbol{a}_n}{\Delta t} \,. \tag{2.30}$$

If this is now applied to Equ. (2.28) and (2.29), the difference quotients of the general Newmark-β-scheme are gained as

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \boldsymbol{v}_n \Delta t + \left[\left(\frac{1}{2} - \beta \right) \boldsymbol{a}_n + \beta \boldsymbol{a}_{n+1} \right] \Delta t^2 \,, \tag{2.31}$$

$$\boldsymbol{v}_{n+1} = \boldsymbol{v}_n + [(1-\gamma)\boldsymbol{a}_n + \gamma \boldsymbol{a}_{n+1}] \Delta t \,. \tag{2.32}$$

Equ. (2.31) can be rewritten to gain the acceleration at the next point in time

$$\boldsymbol{a}_{n+1} = \frac{1}{\beta \Delta t^2} (\boldsymbol{u}_{n+1} - \boldsymbol{u}_n) - \frac{1}{\beta \Delta t} \boldsymbol{v}_n - \left(\frac{1}{2\beta} - 1 \right) \boldsymbol{a}_n \,. \tag{2.33}$$

The resulting system of equations is then obtained by applying Equ. (2.33) to (2.15) and yields

$$\frac{1}{\beta \Delta t^2} \boldsymbol{M} \boldsymbol{u}_{n+1} + \boldsymbol{K} \boldsymbol{u}_{n+1} = f_{n+1} + \boldsymbol{M} \left[\frac{1}{\beta \Delta t^2} \boldsymbol{u}_n + \frac{1}{\beta \Delta t} \boldsymbol{v}_n + \left(\frac{1}{2\beta} - 1 \right) \boldsymbol{a}_n \right] \,. \tag{2.34}$$

The term $\boldsymbol{K}\boldsymbol{u}_{n+1}$ is a nonlinear function of the displacements and hence the system of equations is nonlinear. The equation of motion has to be solved iteratively at every step which results in high computational costs. This is typically done by using the Newton-Raphson method. Other options include the initial stiffness method, modified Newton-Rhapson method, quasi-Newton method and others, see [25].
The benefit of this method is that it is unconditionally stable and a large time increment can be chosen if the parameters β and γ are set appropriately. The limit for stability of the Newmark-β-method is given as [25]

$$\gamma \geq \frac{1}{2} \quad ; \quad \beta \geq \frac{1}{4} \left(\frac{1}{2} + \gamma \right)^2 \,. \tag{2.35}$$

An important property of implicit numerical time integration schemes is, that even if no mechanical damping is present in the system, the amplitude of the solution will still decrease. This is due to an effect called numerical damping and will be explained in more detail in Chapter 2.4.

2.3. Contact mechanics in FEA

For most technical problems the system consists of multiple parts that interact with each other. Contacts can occur between different parts of the model or between one part with itself in form of a self-contact. The definition of these contact interactions is often a key aspect of modelling and inherently imposes a strong nonlinearity to the model. In FEA, contact interactions have a significant impact on the convergence of the solution and the overall computing time of the analysis and need to be implemented carefully. The following contact interactions are possible, [25]:

- Joined:
  Individual parts can be joined together via processes like welding, gluing, riveting and such. A separation or relative movement is no longer possible. Contacts of this sort are called "tied" or "bonded" contacts.
- Sliding:
  Relative movement between the parts is possible, but a separation is not. They can be modelled with or without friction and are called "sliding-only" contacts.
- Dynamic:
  The parts are initially separated but can get into contact due to an impact. Both separation and relative movement are possible.

For the simulation of creep groan the contact interaction between the brake disk and the pads is the most critical issue and needs to be understood carefully. An overview of the most relevant aspects of this problem will be given in this chapter, relying mostly on [1], [22], [25] and [26]. A thorough analysis of the FE modelling of friction contacts with focus on stability can be found e.g. in [11].

2.3.1. Requirements for contacts

A contact between two physical bodies shall be assumed. The mathematical model describing the contact interaction has to be capable of considering the following conditions, [25]:

- No penetration
- No adhesion, i.e. only positive pressure forces can occur in normal direction of the contact interface
- Friction, i.e. frictional forces in tangential direction can be transmitted

An important characteristic is the gap in normal direction g_N between the two surfaces of the contact interface. A basic contact interaction is depicted in Figure 2.8. As one can see, three different states are possible.

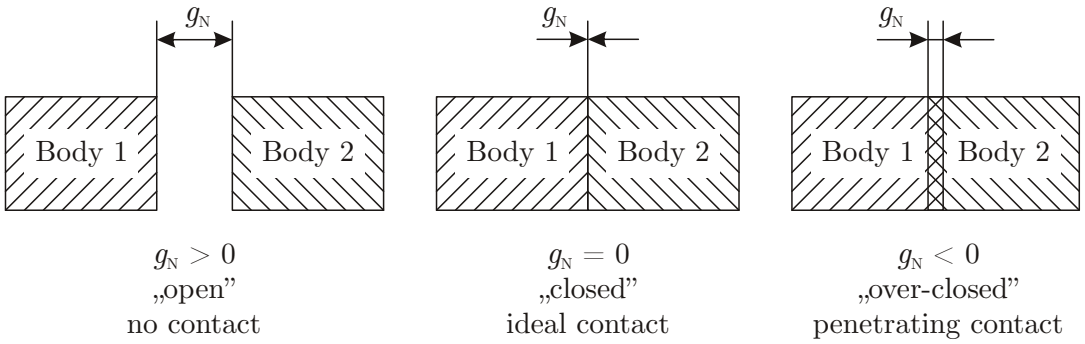

Figure 2.8.: Possible states of a contact interaction

If the gap is positive, the contact is open and no interaction occurs. When the gap reduces to zero, contact is established and the defined contact model comes into effect. The case "over-closed", with a negative gap, is physically impossible, since one body would penetrate the surface of the other one. Numerically, this situation can occur depending on the model used for defining the contact interaction. The modelling of contact behaviour in normal direction will be explained in Chapter 2.3.4.

The behaviour in tangential direction is defined via a friction model, e.g. the Coulomb model of friction. As already mentioned in 2.2.1, a distinction between two different states has to be made: static friction (stick) and kinetic friction (slip). The modelling of friction will be dealt with in Chapter 2.3.3.

2.3.2. Contact constraints and weak form

The deformation of a body can be described using the balance of momentum, see [26]. A similar approach is used e.g. in [11]. For solutions with the finite element method the weak formulation has to be derived. This can be done by multiplying the local equilibrium equation by a virtual displacement or test function $\boldsymbol{\eta}$. The derivation can be found in detail in [26]. We obtain for the weak form of the balance of momentum for a body B in the initial configuration (see Figure 2.9)

$$G(\varphi\,,\,\boldsymbol{\eta}) = \int_B \boldsymbol{S} \cdot \delta\boldsymbol{E}\, dV - \int_B \rho_0(\overline{\boldsymbol{b}} - \dot{\boldsymbol{v}}) \cdot \boldsymbol{\eta}\, dV - \int_{\partial B_\sigma} \overline{\boldsymbol{t}} \cdot \boldsymbol{\eta}\, dA = 0 \tag{2.36}$$

with the second Piola-Kirchhoff stress tensor $\boldsymbol{S}$, the variation $\delta\boldsymbol{E}$ of the Green-Lagrange stress tensor $\boldsymbol{E}$, the body force $\rho_0\overline{\boldsymbol{b}}$, the inertia force $\rho_0\dot{\boldsymbol{v}}$, and the stress vector $\overline{\boldsymbol{t}}$. The first integral in Equ. (2.36) represents the virtual internal work, the other two account for the virtual work of the external forces. The notation of a variable e.g. $\boldsymbol{t}$ as $\overline{\boldsymbol{t}}$ indicates that it is a described (external) quantity. This equation is valid for all problem classes, including plasticity, friction or non-conservative loading, [26].

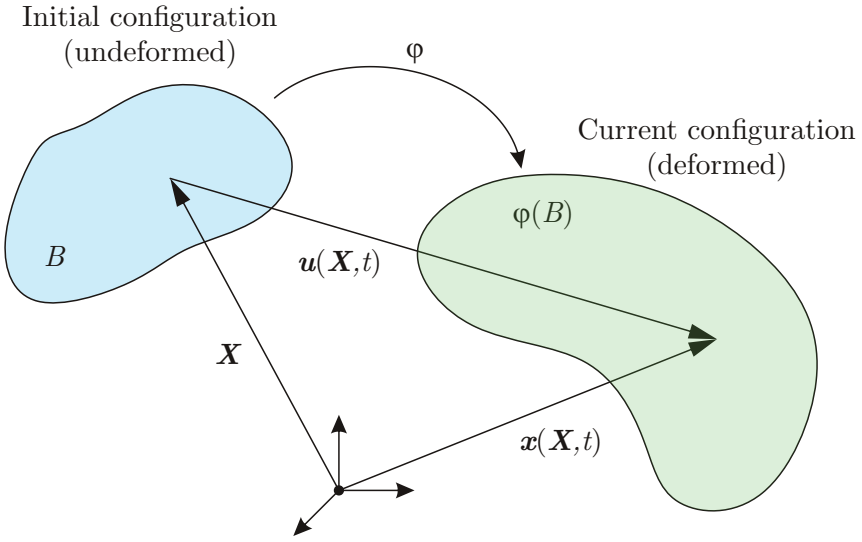

Figure 2.9.: Initial and current configuration of a body B with push-forward φ

The body B will now be transformed from the initial configuration Equ. (2.36) to the current configuration via the push-forward operator $\varphi(B)$, see Figure 2.9. We obtain

$$g(\varphi\,,\,\boldsymbol{\eta}) = \int\limits_{\varphi(B)} \boldsymbol{\sigma} \cdot \nabla^S \boldsymbol{\eta}\, dv - \int\limits_{\varphi(B)} \rho\,(\overline{\boldsymbol{b}} - \dot{\boldsymbol{v}}) \cdot \boldsymbol{\eta}\, dv - \int\limits_{\varphi(\partial B_\sigma)} \overline{\boldsymbol{t}} \cdot \boldsymbol{\eta}\, da = 0 \qquad (2.37)$$

with

$$\nabla^S \boldsymbol{\eta} = \frac{1}{2}(\operatorname{grad} \boldsymbol{\eta} + \operatorname{grad}^{\mathrm{T}} \boldsymbol{\eta})\ . \qquad (2.38)$$

Since all integrals, stresses and gradients in Equ. (2.37) have to be calculated for the current configuration, the system becomes nonlinear even for linear elasticity, [26].
If now two bodies B^γ ($\gamma = 1, 2$) come into contact, a boundary value problem occurs. In order to resolve this, additional terms related to contact have to be added. The equations that describe the behaviour of the individual bodies, e.g. Equ. (2.37), remain unchanged. The boundary value problem shall be explained using a contact of an elastic body B with a rigid surface, depicted in Figure 2.10. This contact is called unilateral, since the motion of the body is constrained from one side by a rigid surface.

The boundary Γ of the body B can be split into three parts: Γ_σ with prescribed surface loads, Γ_u with prescribed displacements and Γ_c where the bodies come into contact. A contact pressure in normal direction p_{N} is applied to both bodies in the contact interface Γ_c due to the surface traction $\overline{\boldsymbol{t}}$ that is applied on Γ_σ.

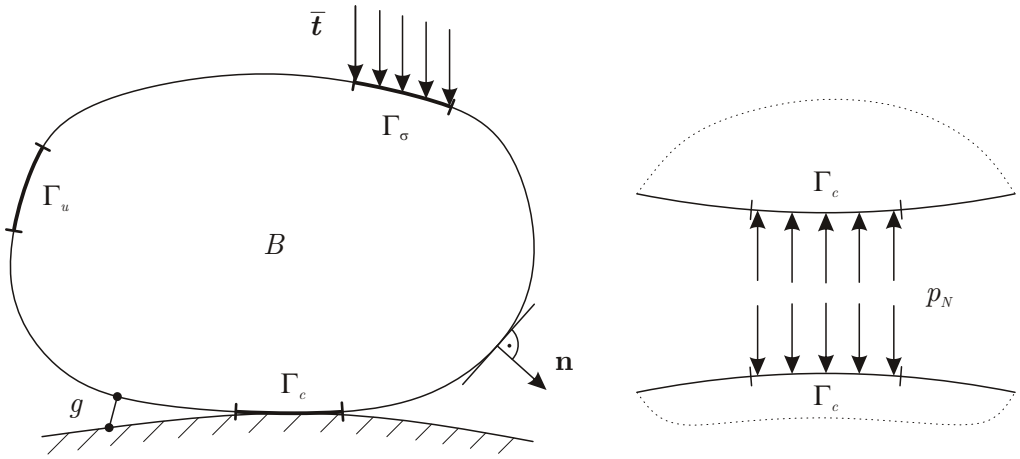

Figure 2.10.: Unilateral contact of an elastic solid, adapted from [26]

The boundaries for the model above can be defined as, [26]

- Displacement boundary conditions (Dirichlet conditions):
  $\boldsymbol{u} = \boldsymbol{0}$ on Γ_u
- Traction boundary conditions (Neumann conditions):
  $\boldsymbol{\sigma}\,\boldsymbol{n} = \bar{\boldsymbol{t}}$ on Γ_σ
  with the stress tensor $\boldsymbol{\sigma}$ and the outward normal of the surface $\boldsymbol{n}$ of the solid.
- Contact conditions:

$$\begin{aligned} u_\mathrm{N} - g &\leq 0 \\ p_\mathrm{N} &\leq 0 \;\text{ on }\; \Gamma_c \\ (u_\mathrm{N} - g)p_\mathrm{N} &= 0 \end{aligned} \tag{2.39}$$

  with the normal component of the displacement field $u_\mathrm{N} = \boldsymbol{u} \cdot \boldsymbol{n}$, the gap g and the contact pressure p_N.

With the condition that no stress due to adhesion can occur in the contact interface, the boundary conditions for normal contact can be written as

$$g_\mathrm{N} \geq 0 \quad ; \quad p_\mathrm{N} \leq 0 \quad ; \quad g_\mathrm{N}\, p_\mathrm{N} = 0 \quad \text{on} \quad \Gamma_c\,. \tag{2.40}$$

Equ. (2.40) can be understood according to Figure 2.11 and are known as the Hertz-Signorini-Mareau conditions for frictionless contact, [26]. If the contact interface is known, as it shall be assumed here, the weak form with inclusion of the contact constraints is obtained as

$$\sum_{\gamma=1}^{2} \left\{ \int_{B^\gamma} \boldsymbol{\tau}^\gamma \cdot \operatorname{grad} \boldsymbol{\eta}^\gamma \, dV - \int_{B^\gamma} \overline{\boldsymbol{f}}^\gamma \cdot \boldsymbol{\eta}^\gamma \, dV - \int_{\Gamma_\sigma^\gamma} \overline{\boldsymbol{t}}^\gamma \cdot \boldsymbol{\eta}^\gamma \, dA \right\} + C_c = 0 \tag{2.41}$$

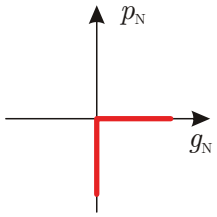

Figure 2.11.: Reaction force versus normal gap, adapted from [26]

with the contact contributions C_c, the Kirchhoff stress $\boldsymbol{\tau}^\gamma$ and the body forces $\overline{\boldsymbol{f}}^\gamma$. The variable γ identifies which body is described ($\gamma = 1, 2$).
Equ. (2.41) represents the total energy of two bodies in contact. For the formulation of C_c a variety of contact constraint enforcement methods are available and will be discussed in Chapter 2.3.4.4.
These methods are generally designed for contact in normal direction. For tangential contact and the occurrence of stick-slip, constitutive relations are needed and will be dealt with in Chapter 2.3.3 , [26].

2.3.3. Tangential contact - friction

Especially for the topic of creep groan the contact behaviour in tangential direction and stick-slip transitions are of utmost significance and will be dealt with in this chapter, mainly relying on [25] and [26].
When a tangential load is applied to a contact interface, a shear stress or friction force is imposed. Depending on the properties of the contact, a distinction in three types of friction can be made:

- Dry friction:
  The contact partners are in direct contact with one another.
- Lubricated friction:
  A lubricant is present in the contact interface that separates the partners.
- Mixed friction:
  A hybrid between the two types above. The partners are partly in contact and partly separated by a layer of lubricant.

The friction force depends on a variety of factor and is thus difficult to describe accurately. It depends on the normal stress between the partners, temperature, material properties, surface roughness, relative velocity and more. For the mathematical description several models have been developed such as e.g. Amonton's law or Coulomb's law.
Friction in FEA is commonly implemented based on the Coulomb law of friction, [25]. It consists of two parts depending on the magnitude of the tangential shear stress $\boldsymbol{t}_\mathrm{t}$ in relation to the maximum possible shear stress $\boldsymbol{t}_{\mathrm{t,max}}$. If $\boldsymbol{t}_\mathrm{t}$ is lower than $\boldsymbol{t}_{\mathrm{t,max}}$ no relative motion occurs and the partners stick (static friction). If $\boldsymbol{t}_{\mathrm{t,max}}$ is exceeded,

relative motion is initiated and the partners slip (kinetic friction), see Figure 2.12.

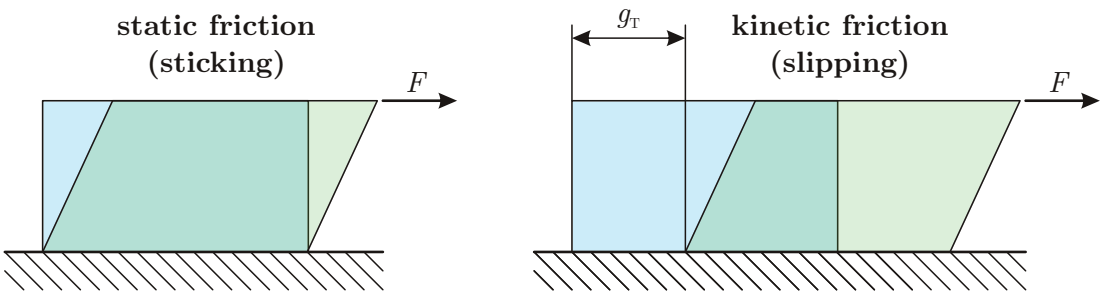

Figure 2.12.: Sticking or slipping in a contact interface, adapted from [26]

For the case of static friction, $\boldsymbol{t}_{t,max}$ is proportional to the normal stress $\boldsymbol{t}_n$ via the static friction coefficient μ_s. The stick condition can thus be formulated as

$$|\boldsymbol{t}_t| \leq \mu_s |\boldsymbol{t}_n| \qquad \text{with} \qquad \boldsymbol{g}_T = \boldsymbol{0} \quad \text{and} \quad \dot{\boldsymbol{g}}_T = \boldsymbol{0} \tag{2.42}$$

with the relative tangential movement $\boldsymbol{g}_T$ and the relative tangential velocity $\dot{\boldsymbol{g}}_T$. It is determined from static equilibrium and prevents any relative motion, [25].

For the case of kinetic friction, the kinetic friction coefficient μ_k has to be used instead. As already mention is μ_k dependent on many factors and can be understood as a $\mu_k = \mu_k(\dot{\boldsymbol{g}}_T, p_N, \theta, ...)$. p_N denotes the contact pressure in normal direction and θ the temperature. It shall be assumed that μ_k is lower than μ_s as is generally the case. $\boldsymbol{t}_t$ for slipping is always pointed in opposite direction to the relative motion $\dot{\boldsymbol{g}}_T$ and is gained as

$$\boldsymbol{t}_t = -\mu_k |\boldsymbol{t}_n| \frac{\dot{\boldsymbol{g}}_T}{|\dot{\boldsymbol{g}}_T|} . \tag{2.43}$$

The transition from static to kinetic friction is very abrupt and a discontinuous shear stress $\boldsymbol{t}_t$ would be the result. In FEA programs this transition is often smoothened with an exponential decay approximation, see Figure 2.13. We obtain μ as a function of the relative velocity as

$$\mu(\dot{\boldsymbol{g}}_T) = \mu_k + (\mu_s - \mu_k)\, e^{-d_c|\dot{\boldsymbol{g}}_T|} . \tag{2.44}$$

d_c represents a parameter that determines how fast the static coefficient approaches the dynamic one, [26].

Upon changes of a node from slip to stick the relative velocity becomes zero. This imposes a severe discontinuity which is very difficult for numerical solvers. According to [25], this can lead to the occurrence of contact chattering for explicit simulations or to convergence issues for implicit ones.

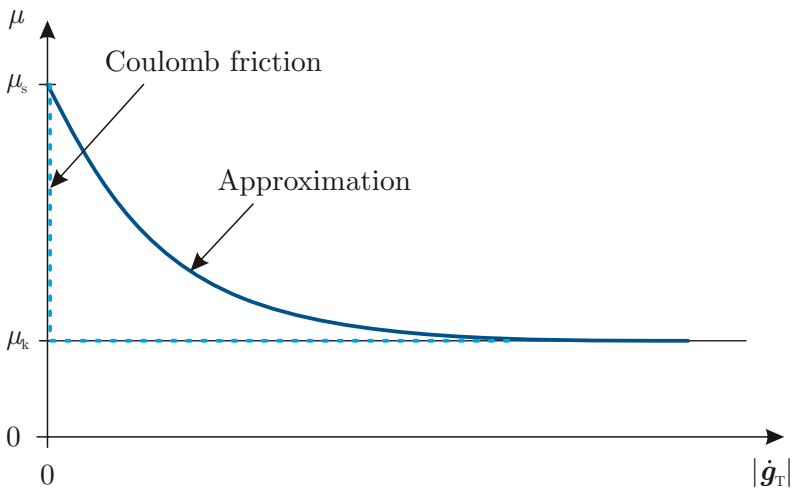

Figure 2.13.: Coulomb law of friction and exponential decay approximation, adapted from [1] and [25]

For the simulation of creep groan, the friction model is likely the biggest influencing factor for the obtained model behaviour. Many modifications to the parameters were necessary in order to achieve stick-slip effects with the models used for this thesis. Parameter studies regarding the effects of various static and kinetic friction coefficients, as well as different slopes for the exponential relationship between μ and $\dot{\boldsymbol{g}}_\mathrm{T}$ were conducted in [23]. The investigation was continued in the follow-up work [24] on a more complex model. Influences of several additional factors were included as well. High values of μ_s and μ_k were found to be beneficial for materials with identical negative μ-$|\dot{\boldsymbol{g}}_\mathrm{T}|$ slope. If materials have the same μ_s, a lower negative slope leads to better creep groan behaviour, [23], [24].

2.3.4. Contact modelling

When modelling a contact interaction in FE, there are many different options to choose from. Usually the user has to decide on a

- Contact discretization,
- Tracking approach,
- Master-slave role assignment,
- Contact constraint enforcement method, and
- Pressure-overclosure relationship.

All of these choices have a significant impact on how the contact surfaces interact with one another and need to be defined carefully for the problem at hand.

2.3.4.1. Contact discretization

The method of contact discretization defines how the contact status is determined. Practically, only two options are commonly used: the node-to-surface discretization and the surface-to-surface discretization. Other variants are either very niche or outdated and will be left out (e.g. edge-to-surface, edge-to-edge, vertex-to-vertex, etc.).

Node-to-surface:

The node-to-surface discretization is the more conservative approach. Every node on the slave surface has to be checked for contact with the master surface. The contact forces are calculated based on the position where contact occurs. Depending on the software the way these positions are calculated can differ. Usually the normal projections of the nodes on the slave surface onto the master surface are sought, see Figure 2.14. Every slave node interacts with its closest point of projection on the master surface on the opposite side of the contact interface. Therefore, one slave node and a group of nearby master nodes are needed to define a single contact condition. The contact for every node on the slave surface has to be defined with this method. Per definition it is impossible for slave nodes to penetrate the master surface, however, the master nodes can theoretically penetrate into the slave surface, [1], [25].

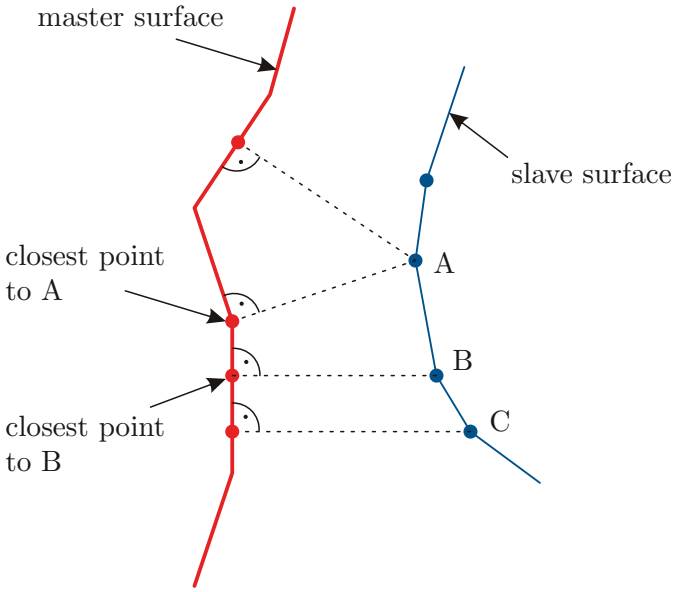

Figure 2.14.: Contact interaction of a node-to-surface discretization, adapted from [1]

The detection of possible contact interactions can be a very challenging task. For example, for two surfaces with 1000 nodes each, one million contact pairs would be possible. That is why certain contact detection algorithms are applied to minimize the computational effort, such as bucket-sort or the pinball-algorithm. More information about contact detection can be found in [22].

Surface-to-surface:

The surface-to-surface discretization does not use an individual slave node for the contact definition. Instead a wider region centered around the slave node is used. Thus a contact constraint uses one primary slave node, but also adjacent secondary nodes. Slight penetrations can occur at individual nodes on either surface of the contact interface, large penetrations of master nodes into the slave surface are however prevented. This more complex contact formulation leads to higher computational costs, [1].

A special type of surface-to-surface contacts is the so called "mortar contact". It can be used to smoothen contacts with non-matching meshes. The general idea is to define the contact interaction via an additional intermediate contact surface between master and slave surface, see Figure 2.15. According to [25], the mortar contact is especially suitable for implicit simulations, since it can help to diminish the influence of discontinuities and improve convergence. Compared to the other options, this contact method leads to longer computing times.
Abaqus does however not support the implementation of mortar contacts. Thus no investigations with this contact model were conducted. More information can be found in e.g. [25] and [26].

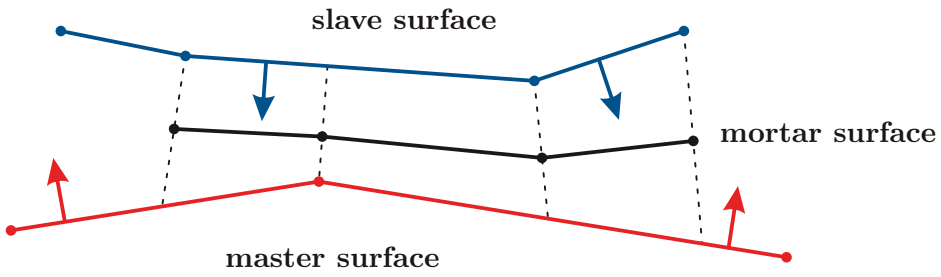

Figure 2.15.: Contact interaction with mortar contact, adapted from [25]

Comparison:

Surface-to-surface discretization generally leads to smoother and more accurate results for contact pressure and contact force than node-to-surface discretization. This is due to the fact that with node-to-surface discretization, a penetration of slave nodes into the master surface is impossible which can lead to large spikes at individual nodes. A refinement of the mesh can help diminish this effect but leads to higher computational costs.
Master-slave roles must be assigned carefully with node-to-surface contacts, but are less critical with surface-to-surface contacts. As already mentioned above, surface-to-surface involves more nodes per contact constraint which leads to higher computing costs. This can especially be the case if the master surface is more refined than the slave surface.
In conclusion, it depends on the contact situation which discretization is preferable. Surface-to-surface works very well for contacts of flat surfaces that are approximately

opposite of each other. Node-to-surface is typically better suited for contacts involving edges or corners and for surfaces whose normal directions are not opposite, [1], [25].

2.3.4.2. Tracking approach

The tracking approach is the method used to determine the relative positions of the contacting surfaces and their relative motion throughout the simulation. Abaqus for example, provides two different options, the finite-sliding and the small-sliding tracking approach, [1].

Finite-sliding tracking approach:

A finite-sliding contact allows for arbitrary relative separation, sliding and rotation of the surfaces in contact. It is the most general approach and can be used for all contacts.

Small-sliding tracking approach:

Small sliding contacts are based on the assumption, that the relative motion of the surfaces is small and thus slave nodes interact always with the same local area of the master surface. This allows for the definition of fixed contact constraints for certain groups of node, which can be set active or inactive. If this approximation is reasonable for the problem at hand, small-sliding can lead to higher robustness and lower computational costs of the solution.

2.3.4.3. Master-slave role assignment

Every contact interaction consists of two surfaces, a master surface and a slave surface. The master surface defines the surface geometry and is a continuous surface. The slave surface only has information about the position of its individual nodes and is thus discontinuous, [22]. There are a few general rules when it comes to assigning master and slave roles. According to [22], the master surface should be:

- the coarser meshed surface,
- the larger or overhanging surface, if overhanging occurs,
- the flat or slightly bent surface,
- the surface that has higher order elements,
- the surface with the stiffer underlying material.

The case of differently refined meshes is particularly important and is depicted in Figure 2.16. If the surface with the finer mesh is chosen as master, it only interacts with two node on the slave surface. The master nodes in between do not interact with any slave nodes and can penetrate into the slave surface without detection. If the surface with the coarser mesh is chosen as master, this situation cannot occur.

Every slave node can interact with master nodes and every potential penetration can be detected.

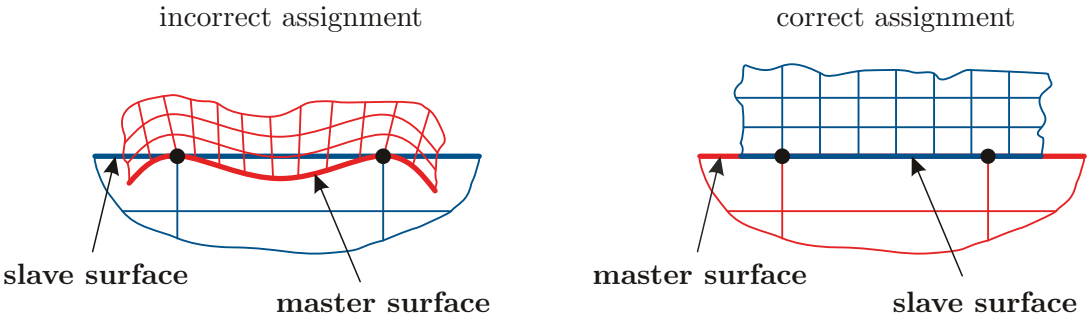

Figure 2.16.: Interaction of surfaces with differently coarse meshes and master slave assignments, adapted from [22]

Contacts can also be defined as symmetric contacts, where the contact is defined twice with respectively swapped master-slave roles and should be considered, if a contradiction between the rules above occurs. They are however susceptible to over-constraint issues, see [22].

2.3.4.4. Contact constraint enforcement methods

Once contact has been established, the contact constraints have to be enforced and solved in combination with the weak form derived in Chapter 2.3.2. This can be done utilizing one of many contact constraint enforcement methods, such as [25]:

- Direct method
- Lagrange multiplier method
- Penalty method
- Augmented Lagrange method

The method determines how physical contact constraints are resolved numerically. Most standard finite element codes capable of contact simulation use either the penalty method or the Lagrange multiplier method. Augmented Lagrange is a combination of the penalty method and the Lagrange multiplier method. Other methods are also available and can be found e.g. in [26]. Besides the direct method, the other approaches enforce the contact constraint in form of a secondary condition in the energy principle and will be explained in more detail.
In Chapter 2.3.5 the direct method, Lagrange multiplier method and the penalty method will be demonstrated on a simple contact of two trusses.

2.3.4.4.1. Direct method

The direct method strictly enforces the contact constraints, which leads to a coupling of different DOFs in Equ. (2.41) and thus reduces the number of unknowns, [26]. However, this method is susceptible to over-constraint issues and therefore common FE solvers may offer additional options to prevent that. Abaqus e.g. provides two options that can be used in combination with this method, [1]:

- hard pressure-overclosure relationship:
  Strict (hard) enforcement of constraints. Available for all methods.
- softened pressure-overclosure relationship:
  For modelling softened contact behaviour according to a function defined or chosen by the user. Only available for the direct method.

A more detailed explanation of pressure-overclosure relationships is provided in Chapter 2.4.2.2.

2.3.4.4.2. Lagrange multiplier method

With the Lagrange multiplier method the contact condition is enforced by strictly enforcing zero penetration $g_\mathrm{N} = 0$, using a Lagrange multiplier λ as a secondary condition in the energy principle. This method is therefore capable of modelling real "hard" contact behaviour where next to no penetration occurs. The multiplier λ is an additional unknown in the equation system which leads to higher computational costs for finding the solution compared to the penalty method. It can be understood similar to a stress in normal direction in the contact interface. The structure of the equation system and its amount of equations are changed by this method and requires specific solvers with high hardware demands. This method by itself is therefore not very suitable for most cases and is mainly used for static problem or in combination with the penalty method. The combination is generally known as augmented Lagrange method, see Chapter 2.3.4.4.4, [25].

According to [26], the contact constraints with Lagrange multipliers are added to Equ. (2.41) as the constraint formulation

$$C_\mathrm{c}^{LM} = \int_{\Gamma_\mathrm{c}} (\lambda_\mathrm{N}\, \delta g_\mathrm{N} + \boldsymbol{\lambda}_\mathrm{T} \cdot \delta \mathbf{g}_\mathrm{T}) dA + \int_{\Gamma_\mathrm{c}} (\delta\lambda_\mathrm{N} g_\mathrm{N} + \delta\boldsymbol{\lambda}_\mathrm{T} \cdot \mathbf{g}_\mathrm{T}) dA \tag{2.45}$$

with the Lagrange multipliers λ_N and $\boldsymbol{\lambda}_\mathrm{T}$ and the normal and tangential gap functions g_N and $\mathbf{g}_\mathrm{T}$.
The first integral represents the virtual work of the Lagrange multipliers along the variation of the gap functions in normal and tangential directions, whereas the second integral describes the enforcement of the constraints. The Lagrange multiplier can be understood similar to the contact pressure p_N and the terms $\boldsymbol{\lambda}_\mathrm{T} \cdot \delta\mathbf{g}_\mathrm{T}$ and $\delta\boldsymbol{\lambda}_\mathrm{T} \cdot \mathbf{g}_\mathrm{T}$

are associated with the tangential stick.
Kinetic friction (slip) has to be considered differently. In case of slipping a tangential stress vector $\boldsymbol{t}_\mathrm{T}$ is gained from the applied friction law (e.g. Coulomb friction) and we obtain

$$C_\mathrm{c}^{slip} = \int_{\Gamma_\mathrm{c}} (\lambda_\mathrm{N}\, \delta g_\mathrm{N} + \boldsymbol{t}_\mathrm{T} \cdot \delta \mathbf{g}_\mathrm{T}) dA + \int_{\Gamma_\mathrm{c}} \delta\lambda_\mathrm{N} g_\mathrm{N} dA \,. \tag{2.46}$$

2.3.4.4.3. Penalty method

The penalty method approximates hard pressure-overclosure behaviour by applying a contact force (penalty), that is proportional to the penetration g_N of the contact surfaces. It can be understood similar to the elastic behaviour of a spring as it is depicted in Figure 2.17. The relationship between overclosure and contact pressure can be either linear or nonlinear and an explanation of how these functions are defined in Abaqus will be provided in Chapter 2.4.2.2.3, [1].

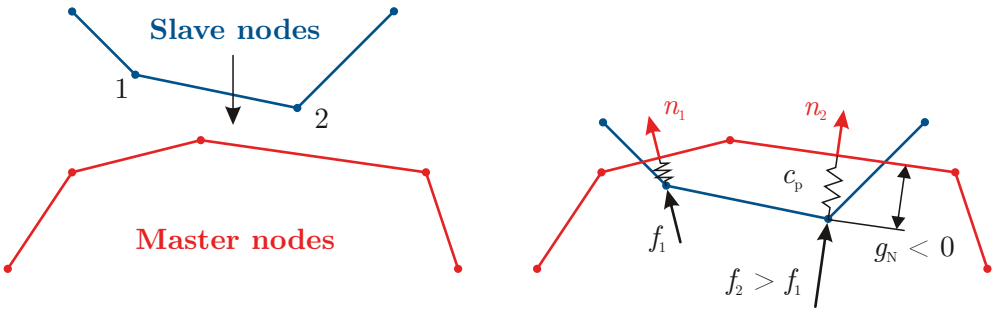

Figure 2.17.: Penalty method for normal contact for node-to-surface discretization, adapted from [25]

A penalty term has to be added to Equ. (2.41) in the form of

$$C_\mathrm{c}^{P} = \int_{\Gamma_\mathrm{c}} (\epsilon_\mathrm{N} g_\mathrm{N} \delta g_\mathrm{N} + \epsilon_\mathrm{T}\, \boldsymbol{g}_\mathrm{T} \cdot \delta \boldsymbol{g}_\mathrm{T}) dA \,, \quad \epsilon_\mathrm{N}\,, \epsilon_\mathrm{T} > 0 \tag{2.47}$$

with the penalty parameters ϵ_N and ϵ_T for normal and tangential direction. The term is only added, when the contact condition Equ. (2.39) is violated, [26].
Equ. (2.47) is valid for normal contact and static friction (stick). For the case of kinetic friction (slip) the term becomes

$$C_\mathrm{c}^{slip} = \int_{\Gamma_\mathrm{c}} (\epsilon_\mathrm{N} g_\mathrm{N} \delta g_\mathrm{N} + \boldsymbol{t}_\mathrm{T} \cdot \delta \boldsymbol{g}_\mathrm{T}) dA \,, \quad \epsilon_\mathrm{N} > 0 \,. \tag{2.48}$$

Since the contact stiffness can be determined from known parameters, no additional unknowns are added to the equation system and its structure is therefore not changed. In order for a penalty force to occur, a minimum penetration is necessary, hence a real "hard" contact behaviour cannot be modelled with this method.

Another disadvantage that is unique to this method is the possibility of contact chattering to occur. If the contact condition is no longer violated, the penalty force disappears, the reaction forces on the other surface of the interface are however still active. This causes the nodes to be pushed back again into the master surface, where a penalty force is applied once again. This can lead to oscillations of the node and of all its associated numerical values and as a result disrupt the solution with a high frequency noise.
For implicit methods this can lead to convergence problems, hence the contact stiffness has to be chosen carefully. A high stiffness achieves small penetrations but high chattering and poor convergence, a low stiffness leads to high penetrations but low chattering and good convergence. Since the contact stiffness is a numerical parameter rather than a physical one, the user may have to try a few different variants in order to find a reasonable value [25].

2.3.4.4.4. Augmented Lagrange method

The augmented Lagrange method is a combination of the penalty method and the Lagrange multiplier method and utilizes the advantages of both methods. Both a Lagrange multiplier and a penalty parameter are introduced to the energy principle.
At first a converging solution is sought with the penalty method. If the penetration of the slave nodes exceeds a certain threshold, the contact force is "augmented" via a Lagrange multiplier and another series of iterations is performed until convergence is achieved. This process is repeated until all penetrations are within an acceptable tolerance.
The resulting equation system is the same as with the penalty method only with an additional force in the right-hand side term. Significantly smaller penetrations can be achieved with this method than with the penalty method and contact chattering does not occur which leads to better convergence. Also, unlike with the Lagrange multiplier method, no additional unknowns are added since the Lagrange multiplier is obtained iteratively from the penalty method. The augmented Lagrange method can lead to higher computing costs since additional iterations are usually necessary, but can help to stabilize contact problems if convergence is an issue.
Due to its intrinsic iterative nature, this method can only be used for implicit solution methods.

2.3.4.5. Pressure-overclosure relationship

In addition to the contact constraint enforcement methods, pressure-overclosure relationships can be defined in Abaqus for contact interfaces and can be either "hard" or "softened". A comparison of the two options is given in Figure 2.18. Overclosure can be understood as the amount of penetration that occurs in the contact interface. Similarly, the term clearance is identical to gap g in previous chapters.
Softened contact relationships can be modelled in different ways (linear, exponential, etc.). An overview of the options provided by Abaqus will be given in Chapter 2.4.2.2.1.

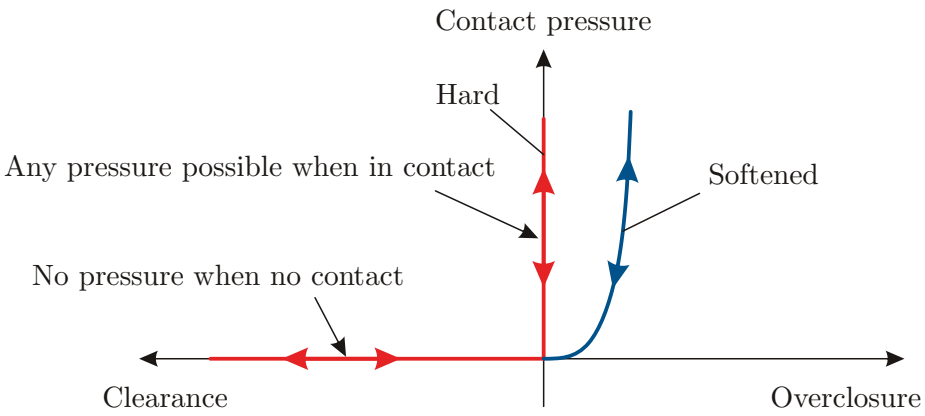

Figure 2.18.: Hard vs. softened pressure-overclosure relationship, adapted from [1]

For the penalty method, the Lagrange multiplier method and the augmented Lagrange method the pressure-overclosure relationship is automatically always "hard". For the direct method, "hard" or "softened" can be chosen. If the softened contact is modelled too stiff, Lagrange multipliers may have to be used in this case. For hard pressure-overclosure with the direct method, Lagrange multipliers always have to be used, [1].

2.3.5. Example: Trusses

The way contact constraints are enforced with the penalty method, Lagrange multiplier method and direct method shall be demonstrated using a basic example of two trusses with an initial gap g that come into contact, see Figure 2.19, according to [26]. Only behaviour in normal direction will be considered here, the tangential behaviour (friction) will be left out.

The axial stiffness of the left truss with length $3l$ is EA/l and shall be discretized by three truss elements with linear shape functions. The stiffness of the right truss with length l is $2EA/l$. A point load F_3 is applied to node 3 on the left truss. With increasing force F_3, the gap g between the trusses decreases until it closes at a certain value of F_3 and the trusses are in contact.

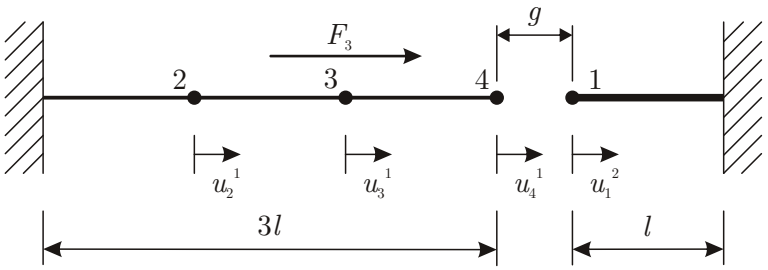

Figure 2.19.: Truss structure with initial gap, adapted from [26]

The system can be mathematically described as

$$\boldsymbol{K}_0\,\boldsymbol{u} = \boldsymbol{f} \;\rightarrow\; \frac{EA}{l}\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \begin{Bmatrix} u_2^1 \\ u_3^1 \\ u_4^1 \\ u_1^2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ F_3 \\ 0 \\ 0 \end{Bmatrix} . \tag{2.49}$$

The solution yields

$$\boldsymbol{u}^{\mathrm{T}} = \frac{F_3 l}{EA}\,\{\,1\;,\;2\;,\;2\;,\;0\,\}^{\mathrm{T}} . \tag{2.50}$$

Once the gap is closed, the constraint problem has to be solved using one of the methods derived in Chapter 2.3.4.4. Before this can be done, the gap condition and its variation have to be formulated. For normal direction we obtain

$$g_{\mathrm{N}} = g - (u_4^1 - u_1^2) \tag{2.51}$$

and the variation

$$\delta g_{\mathrm{N}} = -\delta u_4^1 + \delta u_1^2 . \tag{2.52}$$

For the Lagrange multiplier and the penalty method a system of linear equations can be defined and the solution can be obtained directly. For the augmented Lagrange method the equation system is nonlinear and has to be solved iteratively, [26].

2.3.5.1. Solution with penalty method

With the penalty method, the constraint is added by $\epsilon_{\mathrm{N}}\, g_{\mathrm{N}}\, \delta g_{\mathrm{N}}$ to Equ. (2.49), which leads with Equ. (2.51) and (2.52) to the equation system $\boldsymbol{K}_{\mathrm{P}}\,\boldsymbol{u}_{\mathrm{P}} = \boldsymbol{f}_{\mathrm{P}}$ and can be written as

$$\begin{bmatrix} 2\frac{EA}{l} & -\frac{EA}{l} & 0 & 0 \\ -\frac{EA}{l} & 2\frac{EA}{l} & -\frac{EA}{l} & 0 \\ 0 & -\frac{EA}{l} & \frac{EA}{l} + \epsilon_{\mathrm{N}} & -\epsilon_{\mathrm{N}} \\ 0 & 0 & -\epsilon_{\mathrm{N}} & 2\frac{EA}{l} + \epsilon_{\mathrm{N}} \end{bmatrix} \begin{Bmatrix} u_2^1 \\ u_3^1 \\ u_4^1 \\ u_1^2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ F_3 \\ \epsilon_{\mathrm{N}} g \\ -\epsilon_{\mathrm{N}} g \end{Bmatrix} . \tag{2.53}$$

As already mentioned, this equation system can be solved directly. For the variable u_4^1 for example, we obtain, [26]

$$u_4^1 = \frac{2(3EAl\epsilon_{\mathrm{N}}g + 2EAFl + \epsilon_N F_3 l^2)}{2(EA)^2 + 7\epsilon_{\mathrm{N}}EAl} \,. \tag{2.54}$$

2.3.5.2. Solution with Lagrange multiplier method

For the Lagrange multiplier method, Equ. (2.45) is used and the terms $\lambda_{\mathrm{N}}\delta g_{\mathrm{N}} + \delta\lambda_{\mathrm{N}} g_{\mathrm{N}}$ are added. In combination with Equ. (2.51) and (2.52), the equation system $\boldsymbol{K}_{\mathrm{LM}}\boldsymbol{u}_{\mathrm{LM}} = \boldsymbol{f}_{\mathrm{LM}}$ is obtained and can be written as

$$\begin{bmatrix} 2\frac{EA}{l} & -\frac{EA}{l} & 0 & 0 & 0 \\ -\frac{EA}{l} & 2\frac{EA}{l} & -\frac{EA}{l} & 0 & 0 \\ 0 & -\frac{EA}{l} & \frac{EA}{l} & 0 & -1 \\ 0 & 0 & 0 & 2\frac{EA}{l} & 1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix} \begin{Bmatrix} u_2^1 \\ u_3^1 \\ u_4^1 \\ u_1^2 \\ \lambda_{\mathrm{N}} \end{Bmatrix} = \begin{Bmatrix} 0 \\ F_3 \\ 0 \\ 0 \\ -g \end{Bmatrix} \,. \tag{2.55}$$

Since this problem is geometrically linear, the initial gap g can be used on the right-hand side of Equ. (2.55) instead of the the gap g_{N} as given in Equ. (2.51). A closed form solution of Equ. (2.55) can be obtained as

$$\boldsymbol{u}_{\mathrm{LM}} = \frac{2}{7EA} \begin{Bmatrix} (2EAg + 3F_3 l)/2 \\ 2EAg + 3F_3 l \\ 3EAg + F_3 l \\ (-EAg + 2F_3 l)/2 \\ EA(-EAg + 2F_3 l)/l \end{Bmatrix} \,. \tag{2.56}$$

The Lagrange multiplier is represented by the fifth term in Equ. (2.56) and can be understood as the reaction force in the contact interface, [26].

2.3.5.3. Solution with Direct method

With this method, the constraint Equ. (2.51) is enforced directly. A projection matrix $\boldsymbol{P}$ has to be constructed in order to reduce the amount of displacement variables by one:

$$\boldsymbol{u} = \begin{Bmatrix} u_2^1 \\ u_3^1 \\ u_4^1 \\ u_1^2 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u_2^1 \\ u_3^1 \\ u_4^1 \end{Bmatrix} - \begin{Bmatrix} 0 \\ 0 \\ 0 \\ g \end{Bmatrix} = \boldsymbol{P}^{\mathrm{T}}\,\overline{\boldsymbol{u}} - \boldsymbol{g} \,. \tag{2.57}$$

Equ. (2.49) can now be pre- and post-multiplied by $\boldsymbol{P}$, and we obtain

$$\boldsymbol{K}_{\mathrm{D}}\overline{\boldsymbol{u}} = \boldsymbol{P}\boldsymbol{K}_0\boldsymbol{P}^{\mathrm{T}}\overline{\boldsymbol{u}} = \boldsymbol{P}\boldsymbol{f} + \boldsymbol{P}\boldsymbol{K}_0\boldsymbol{g} \;, \tag{2.58}$$

or in explicit form,

$$\frac{EA}{l}\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{bmatrix}\begin{Bmatrix} u_2^1 \\ u_3^1 \\ u_4^1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ F_3 \\ \frac{2EAg}{l} \end{Bmatrix} \;. \tag{2.59}$$

The stiffness matrix is hereby still symmetrical. The solution is obtained as

$$\overline{\boldsymbol{u}} = \frac{2}{7EA}\begin{Bmatrix} (2EAg + 3F_3l)/2 \\ 2EAg + 3F_3l \\ 3EAg + F_3l \end{Bmatrix} \;. \tag{2.60}$$

The eliminated displacement u_1^2 can be calculated with Equ. (2.51) as

$$u_1^2 = u_4^1 - g = \frac{2}{7EA}(3EAg + F_3l) - g = \frac{2}{7EA}(-\frac{1}{2}EAg + F_3l) \;. \tag{2.61}$$

When combining Equ. (2.60) and (2.61) one can notice, that the solution is identical to Equ. (2.56) gained with the Lagrange multiplier method.

Once $\boldsymbol{u}$ is known, λ_{N} can be calculated by decomposing the equation system, Equ. (2.49), in

$$\boldsymbol{K}_0\boldsymbol{u} + \boldsymbol{A}\lambda_{\mathrm{N}} = \boldsymbol{f} \quad \text{with} \quad \boldsymbol{A}^{\mathrm{T}} = \{0\ ,\ 0\ ,\ -1\ ,\ 1\} \quad \text{and} \quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{u} = -g \;. \tag{2.62}$$

This equation can now be pre-multiplied with $\boldsymbol{A}^{\mathrm{T}}$ to gain

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\lambda_{\mathrm{N}} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{f} - \boldsymbol{A}^{\mathrm{T}}\boldsymbol{K}_0\boldsymbol{u} \;. \tag{2.63}$$

It is necessary to use $\boldsymbol{A}^{\mathrm{T}}$ since $\boldsymbol{A}^{-1}$ does not exist. For this example we have $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = 2$ and $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{f} = 0$ which leads to

$$\lambda_{\mathrm{N}} = -\frac{1}{2}\{0\ ,\ 0\ ,\ -1\ ,\ 1\}\frac{EA}{l}\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}\begin{Bmatrix} u_2^1 \\ u_3^1 \\ u_4^1 \\ u_1^2 \end{Bmatrix} = \frac{4}{7}F_3 - \frac{2}{7}\frac{EAg}{l} \;. \tag{2.64}$$

As one can see we obtained the same solution with the direct method as with the Lagrange multiplier method, [26].

2.4. Implementation in Abaqus

The implementation in Abaqus of the topics presented in Chapter 2.2 and 2.3 along with a few additional options available will be provided in this chapter, based on [1].

2.4.1. Direct time integration

For direct time integration of dynamic problems an integration method has to be chosen, either explicit or implicit (see Chapter 2.2.3). Abaqus provides explicit time integration in Abaqus/Explicit, while implicit time integration is provided in Abaqus/Standard.

2.4.1.1. Abaqus/Explicit

Abaqus/Explicit always uses the central difference operator (see Chapter 2.2.3.1) for time integration and only allows for very limited modification by the user.

The time incrementation can be modified via the time increment parameter. The following mutually exclusive options are available:

- DIRECT USER CONTROL:
  A fixed time increment is used and has to be specified by the user. The total time period T and the increment size Δt have to be defined.
- ELEMENT BY ELEMENT:
  A variable, automatic time incrementation is used. Only the total time period T has to be defined. Optionally, an upper limit for the time increment $\Delta t_{\max}$ can be added. If omitted, there is no upper limit.
- FIXED TIME INCREMENTATION:
  A fixed time increment is used that is determined by Abaqus/Explicit automatically at the beginning of the simulation. Again, only the total time period T has to be defined.

Several additional options can be used, but are mostly irrelevant for the topic of this thesis, such as:

- ADIABATIC:
  Can be set to "YES" or "NO" and enables an adiabatic stress analysis (only relevant for metal plasticity).
- IMPROVED *DT* METHOD:
  Can be set to "YES" or "NO" and defines, whether an "improved" or a conservative method should be used for the estimation of the stable time increment. Default setting is "YES".
- SCALE FACTOR:
  Only available in conjunction with ELEMENT BY ELEMENT or FIXED TIME

INCREMENTATION. A scale factor can be defined by which the calculated time increment should be multiplied (default: SCALE = 1.0).

2.4.1.2. Abaqus/Standard

Abaqus/Standard offers several different operators for implicit time integration to choose from, [1]:

- BWE: Backward Euler time integrator
  Since this operator was not used for this thesis, a further explanation of the scheme will not be given.
- HHT-TF: Hilber-Hughes-Taylor time integrator for transient fidelity
  The HHT method is based on the Newmark-β method explained in Chapter 2.2.3.2. The suffix TF represents the application type and sets the parameters for the HHT method. More information can be found below.
- HHT-MD: Hilber-Hughes-Taylor time integrator for moderate dissipation
  MD sets different parameters for the HHT method (see below).

Before the algorithm and its parameters are elaborated, an explanation of the different application types available in Abaqus/Standard and their intended usage will be given. Depending on the requirements of the problem at hand, the user can choose one of the following applications types, [1]:

- QUASI-STATIC:
  For the determination of a final static response. The BWE time integrator is chosen by default for this application type. Since this type is not suitable for the subject of this thesis, no further explanation will be provided.
- TRANSIENT FIDELITY: (default for problems without contact in the model)
  For problems, where a minimum amount of dissipation is desired. The solver is set to control time increments "conservatively" (see below) by default to assure an accurate solution of the vibrational response of the structure. This can lead to convergence problems for simulations involving contacts and/or nonlinearities.
- MODERATE DISSIPATION: (default for problems with contact in the model)
  For a broader range of dynamic problems, where a moderate amount of energy is dissipated (e.g. due to plasticity, viscous damping, impact, friction, etc.). For these kind of applications, high-frequency response is usually not of interest, hence numerical dissipation is desired. Furthermore, time incrementation is more "aggressive" than for transient fidelity if the default setting is left unchanged. This leads to better convergence behaviour of the solution.

The application types above have to be chosen in combination with a time incrementation scheme. This scheme governs when and how changes to the time increment are made, based on the convergence behaviour of the Newton-Raphson iterations. There

are two default settings available in Abaqus/Standard (AGGRESSIVE or CONSERVATIVE), and a customization is possible if needed.

- CONSERVATIVE increment control: (default for HHT-TF)
  In case of divergence or slow convergence rate, the increment size is reduced (same as with aggressive control). Additionally, size reductions are performed upon detection of changes of the contact status (open, closed, stick, slip). If rapid convergence occurred in previous increments, the increment size is slowly increased (conservatively).
- AGGRESSIVE increment control: (default for HHT-MD)
  If an increment appears to diverge or if the convergence rate is too slow, the increment size is reduced. If rapid convergence occurred in previous increments, the increment size is increased fairly aggressively.

Abaqus allows the user to change every setting and criterion related to convergence behaviour and solution control if the predefined settings are deemed insufficient. However, this is generally not recommended and can lead to highly nonphysical results, [1]. A profound understanding of the parameters and effects associated with their modification is mandatory.

Hilber-Hughes-Taylor (HHT-α) method:

The HHT-α method is an extension of the Newmark-β method and was developed in [13] where the exact derivation can be found. In the follow-up work [12], the properties of this scheme are compared to similar algorithms and analysed with respect to overshooting, dissipation and dispersion. α-dissipation schemes were found to be superior regarding aforementioned criteria, [12]. The requirements for this method were:

- Unconditionally stable when applied to linear problems
- Numerical dissipation which can be controlled by a parameter other than the time step
- Zero numerical dissipation should be possible
- Numerical dissipation should not affect the lower modes too strongly

These requirements are predicated on the assumption, that for dynamic applications only the low mode response is of interest. Higher modes should be damped out by numerical dissipation. The Newmark-β method allows for the amount of dissipation to be controlled by a parameter γ, see (2.35), other than the time step. Increasing γ would increase the numerical dissipation for a fixed time step. The dissipative behaviour of this method was not satisfying, since the lower modes were affected too strongly, [13]. A new parameter α is added to the method which introduces a new form of dissipation, called α-dissipation.

This will be shown using the simple equation of motion without damping

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{f} \tag{2.65}$$

analogous to Equ. (2.13). With the newly introduced parameter α, the discrete equation becomes

$$\boldsymbol{M}\boldsymbol{a}_{n+1} + (1+\alpha)\boldsymbol{K}\boldsymbol{u}_{n+1} - \alpha\boldsymbol{K}\boldsymbol{u}_n = \boldsymbol{f}_{n+1} \tag{2.66}$$

with

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \Delta t\boldsymbol{v}_n + \Delta t^2[(\frac{1}{2} - \beta)\boldsymbol{a}_n + \beta\boldsymbol{a}_{n+1}] \tag{2.67}$$

and

$$\boldsymbol{v}_{n+1} = \boldsymbol{v}_n + \Delta t[(1-\gamma)\boldsymbol{a}_n + \gamma\boldsymbol{a}_{n+1}] \; . \tag{2.68}$$

$\boldsymbol{u}_n$, $\boldsymbol{v}_n$ and $\boldsymbol{a}_n$ are approximations of $\boldsymbol{u}(t_n)$, $\dot{\boldsymbol{u}}(t_n)$ and $\ddot{\boldsymbol{u}}(t_n)$ with $t_n = n\Delta t$ and $\boldsymbol{f}_n = \boldsymbol{f}(t_n)$. α, β and γ are parameters that govern the stability and numerical dissipation of the algorithm. As one can see in Equ. (2.66), α basically defines which portion if the displacement shall be used from the current time step t_n, and which one from the next time step t_{n+1}.

If the parameter α is set to $\alpha = 0$, the algorithm is identical to the Newmark-β method and Equ. (2.66) becomes identical to Equ. (2.15). In this case no numerical dissipation is present if γ is set to $\gamma = \frac{1}{2}$ and β is obtained according to Equ. (2.35). With $\gamma > \frac{1}{2}$, numerical dissipation is introduced to the algorithm.

As it is stated in [13], positive values of α lead to rather ineffective dissipation, since its qualitative behaviour is the same as that of linear viscous damping. Therefore, negative values of α in combination with particular values of β and γ were found to lead to the desired behaviour of the algorithm. β and γ are set as

$$\beta = \frac{(1-\alpha)^2}{4} \qquad \text{and} \qquad \gamma = \frac{1}{2} - \alpha \; . \tag{2.69}$$

It is derived in [13], that this algorithm is stable for $\Delta t/T \to \infty$ whenever $-\frac{1}{2} \leq \alpha \leq 0$. It is further suggested, that decreasing α below $-\frac{1}{3}$ does not lead to a further increase of numerical damping, hence the range $-\frac{1}{3} \leq \alpha \leq 0$ is of particular interest.
The HHT-α integrator is commonly used for solving nonlinear problems (e.g. Abaqus, Ansys). Alternative options include:

- Newmark-β operator (e.g. Nastran, Ansys)
- BWE operator (e.g. Abaqus, Comsol)
- Generalized α method (e.g. Comsol)
- Houbolt operator (e.g. Nastran)

As already mentioned, the HHT-α algorithm is used for implicit direct time integration in Abaqus/Standard. The application type, TRANSIENT FIDELITY (TF), or MODERATE DISSIPATION (MD), sets the parameters α, β and γ to predefined values that can be found in Table 2.2.

Table 2.2.: Default parameters for the HHT-α integrator, adapted from [1]

Parameter	Application	
	TRANSIENT FIDELITY	MODERATE DISSIPATION
α	-0.05	-0.41421
β	0.275625	0.5
γ	0.55	0.91421

As one can see in Table 2.2, the values of β and γ for either application type are obtained by applying Equ. (2.69). The parameter α can be modified manually within the allowed range of $-\frac{1}{2} \leq \alpha \leq 0$. $\alpha = -\frac{1}{3}$ provides the maximum amount of numerical damping, whereas $\alpha = 0$ provides no numerical damping, [1].

The HHT-α method provided the basis for the development of more advanced integration algorithms, such as the "Generalized α method" in [9], or the "two-step Lambda method" in [15]. They are however not implemented in Abaqus.

For problems involving contacts, Abaqus provides an additional setting that can be chosen when it comes to defining the integration scheme. If this option is used, an additional time increment with a very small size is subsequently performed upon changes in contact status in order to enforce compatibility of velocities and accelerations across the contact interface. This setting is called IMPACT and the available options are:

- NO: (default for MD)
  Additional increments are not performed. The option is disabled.
- AVERAGE TIME: (default for TF)
  The above described procedure is performed with an automatic cut back of the increment size.
- CURRENT TIME:
  The procedure is performed without an automatic cut back.

A brief summary of the relevant options available for time integration schemes in Abaqus/Standard can be found in Figure 2.20. Options that were not relevant for the matter of this thesis are omitted.

	MODERATE DISSIPATION	TRANSIENT FIDELITY
IMPACT	**Default:** **NO** Optional: AVERAGE TIME CURRENT TIME	**Default:** **AVERAGE TIME** Optional: NO CURRENT TIME
INCREMENTATION	**Default:** **AGGRESSIVE** Optional: CONSERVATIVE	**Default:** **CONSERVATIVE** Optional: AGGRESSIVE
INTEGRATOR	**Default:** **HHT-MD** Optional: HHT-TF	**Default:** **HHT-TF** Optional: HHT-MD

Figure 2.20.: Default and optional parameter for MODERATE DISSIPATION and TRANSIENT FIDELITY applications of the HHT-α algorithm, adapted from [1]

2.4.2. Contact definition

The implementation of contact modelling aspects in Abaqus laid out in Chapter 2.3 will be explained, mainly relying on [1]. The behaviour of a contact interaction can be separated in two general topics: normal direction (penetration, overclosure, clearance) and tangential direction (friction, stick, slip). Some of the methods may differ in Abaqus/Standard and Abaqus/Explicit or are only available for one of the two.

2.4.2.1. GENERAL CONTACT vs. CONTACT PAIR

Abaqus provides two general approaches when it comes to defining contacts: GENERAL CONTACT and CONTACT PAIR. They both have in common, that a contact definition must always contain at least one node-based surface, and at least one surface that is a non-analytical rigid surface, [1].

A further distinction between GENERAL CONTACT and CONTACT PAIR definitions in Abaqus/Standard and Abaqus/Explicit has to be made since there are fundamental differences.

- Abaqus/Standard:
  GENERAL CONTACT and CONTACT PAIR are similar in Abaqus/Standard and share many underlying algorithms. The main difference lies in the user interface and the default numerical settings, see Table 2.3, [1].

Table 2.3.: Default settings for GENERAL CONTACT and CONTACT PAIR in Abaqus/Standard, adapted from [1]

GENERAL CONTACT	CONTACT PAIR
FINITE-SLIDING, SURFACE-TO-SURFACE contact discretization	FINITE-SLIDING, NODE-TO-SURFACE contact discretization
Automatically accounts for thicknesses and offsets associated with shell-like surfaces	Does not account for shell thicknesses and offsets
Penalty method for constraint enforcement	Augmented Lagrange method for constraint enforcement
Strain-free adjustments remove initial overclosures	Initial overclosures are treated as interference fits and are resolved in the first increment
Automatic pure master-slave assignment	Manual assignment of master-slave roles

  The CONTACT PAIR definition allows for a few additional aspects to be modelled, e.g. such as:

  – Contact involving node-based surfaces
  – SMALL-SLIDING contact
  – Tied contact

  It is possible to define both GENERAL CONTACT and CONTACT PAIR definitions for different contacts of the same analysis if needed.

- Abaqus/Explicit:
  In Abaqus/Explicit, GENERAL CONTACT and CONTACT PAIR differ vastly in design of the numerical algorithms and in their general implementation. GENERAL CONTACT offers more freedom of the types of surfaces that can be used and provides a more automated contact definition. The CONTACT PAIR algorithm restricts the surfaces that can be used and is more complicated to implement. It does however allow for the use of additional features, such as kinematically enforced contact, small sliding contact or exponential contact pressure-overclosure relationships.

2.4.2.2. Behaviour in normal direction

The methods of contact constraint enforcement, as they are described in Chapter 2.3.4.4, can be implemented in Abaqus. The pure Lagrange multiplier method can not be used by itself but is utilized in combination with the direct method as the augmented Lagrange method in Abaqus/Standard.
Abaqus/Explicit offers a unique constraint enforcement method that is based on a predictor/corrector algorithm, the kinematic contact algorithm. The penalty method can be used with either Abaqus/Standard or Abaqus/Explicit.

2.4.2.2.1. Direct method and pressure-overclosure relationships

As mentioned in Chapter 2.3.4.4.1, the direct method is used in combination with either hard or softened pressure-overclosure behaviour (see Chapter 2.3.4.5). For a direct, hard contact definition no further specifications are possible or needed. In case of softened contact behaviour, the user can choose one of four options to define the correlation of pressure and overclosure, [1]:

- Linear:
  For linear pressure-overclosure behaviour only the slope can be defined. The linear function always starts at the origin, see Figure 2.21.

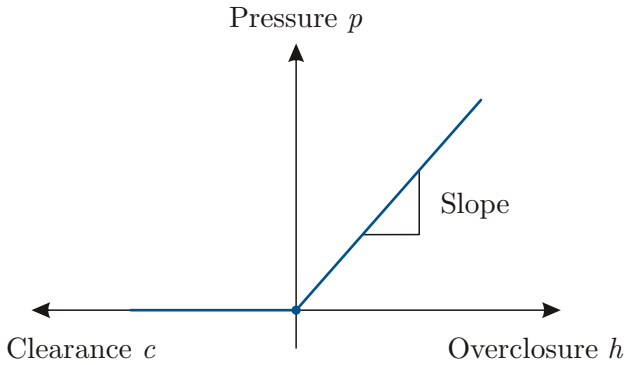

Figure 2.21.: Softened, linear pressure-overclosure relationship, adapted from [1]

- Tabular:
  A piecewise linear relationship can be defined via a table of coordinates of the points (p_i, h_i). It is possible to define contact pressure being transmitted before overclosure occurs. The piecewise linear functions must always have a positive slope.

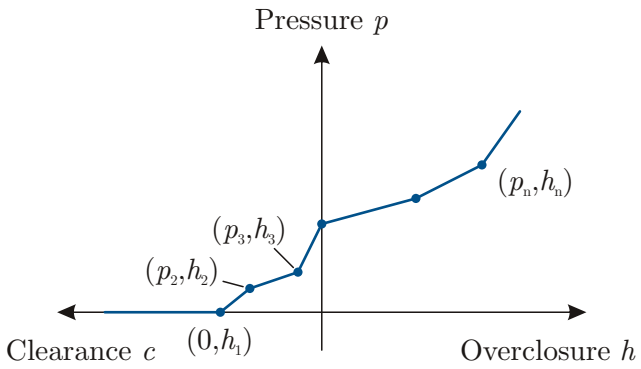

Figure 2.22.: Softened, tabular pressure-overclosure relationship, adapted from [1]

- Exponential:
  Only available for CONTACT PAIR in Abaqus/Explicit, for both GENERAL CONTACT and CONTACT PAIR in Abaqus/Standard. The pressure-overclosure relationship follows an exponential function. The parameters c_0 and p_0 can be defined. In Abaqus/Explicit a maximum slope $K_{\max}$ can be specified additionally.

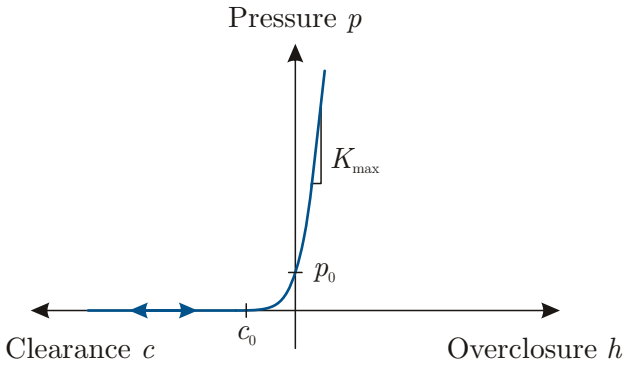

Figure 2.23.: Softened, exponential pressure-overclosure relationship, adapted from [1]

- Scale factor:
  Only available for GENERAL CONTACT in Abaqus/Explicit, for both GENERAL CONTACT and CONTACT PAIR in Abaqus/Standard. Alternatively, the relationship can be constructed by geometrically scaling the default contact stiffness. Whenever a multiple of a user-defined penetration measure is exceeded,

the default stiffness is scaled by a factor s. The penetration measure d can be defined directly, or as a fraction r of the minimum element stiffness L_{elem} in the contact interface. The initial stiffness is gained by multiplying the default contact stiffness by a factor s_0.

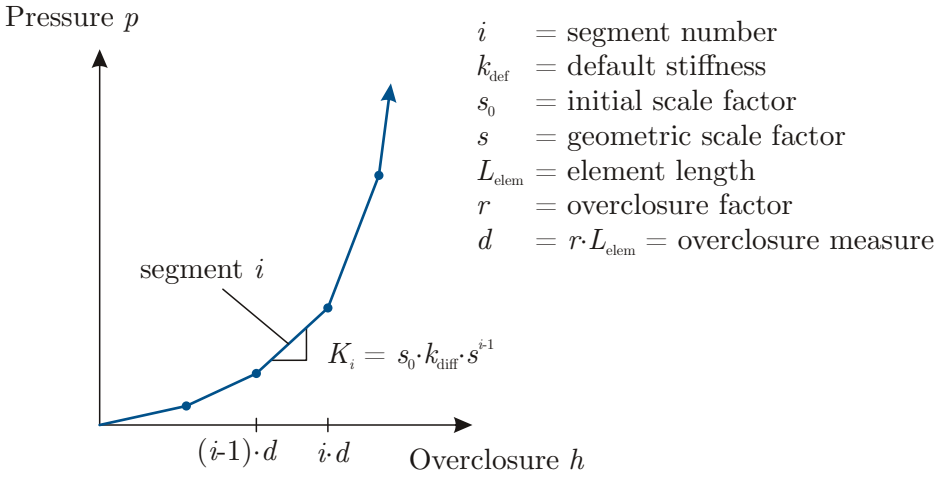

Figure 2.24.: Softened pressure-overclosure relationship with scale factor, adapted from [1]

2.4.2.2.2. Kinematic contact algorithm

The kinematic contact algorithm is exclusive to Abaqus/Explicit. Similar to the direct method in the previous chapter, it is used in combination with the same pressure-overclosure relationships (hard or softened).

At the start of each increment, Abaqus/Explicit first predicts the kinematic state of the model without consideration of contact conditions. The (predicted) penetration of the slave nodes into the master surface is then determined. Based on this, the force required to oppose penetration is calculated and subsequently applied to the master surface.
If the master surface is formed by element faces, the combined force of all slave nodes as well as their mass is distributed to the nodes of the master surface. This is then used to determine an acceleration correction for the master nodes, and based on the results, an acceleration correction for the slave nodes is also calculated. With these corrections, Abaqus/Explicit can now obtain the corrected configuration in which the contact constraints are enforced.
If the master surface is a rigid surface, the combined force is applied as generalized force on the rigid body. The remaining procedure is the same, [1].

2.4.2.2.3. Penalty method

The penalty method (see Chapter 2.3.4.4.3) is used both in Abaqus/Standard and Abaqus/Explicit, however their implementation differs. In Abaqus/Explicit the default penalty stiffness is determined automatically based on the representative stiffness of the underlying elements. The user can only specify a factor by which the default stiffness should be scaled. In Abaqus/Standard the stiffness can be specified exactly by the user with either a linear (default) or a nonlinear function, [1].

- Penalty linear
  The overclosure/clearance c_0 at which the contact pressure is starting to be applied can be specified. The linear slope K_{lin} can be defined directly, or in form of a scale factor for the default linear penalty stiffness.

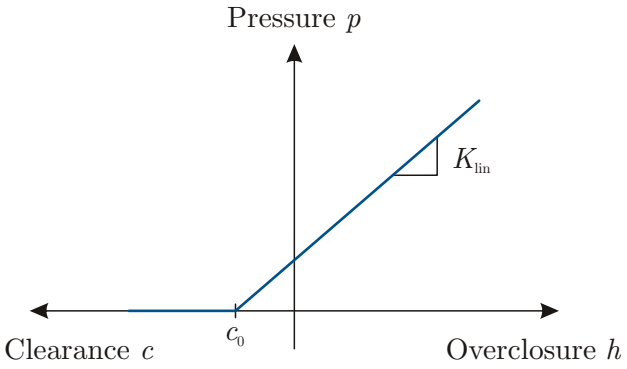

Figure 2.25.: Linear penalty contact, adapted from [1]

- Penalty nonlinear
  The nonlinear function for the penalty method consists of four distinct regions:
  - Inactive contact regime:
    The contact pressure and the penalty stiffness are zero for clearances $c > c_0$ (default: $c_0 = 0$).
  - Constant initial penalty stiffness regime:
    The contact pressure increases linearly with a constant stiffness (slope) of K_{i} in the range c_0 to e.
  - Stiffening regime:
    The contact pressure increases quadratically in the range e to d. The penalty stiffness increases linearly from K_{i} to K_{f}.
  - Constant final penalty stiffness regime:
    The contact pressure increases linearly with the constant stiffness (slope) K_{f} for overclosures (penetrations) greater than d.

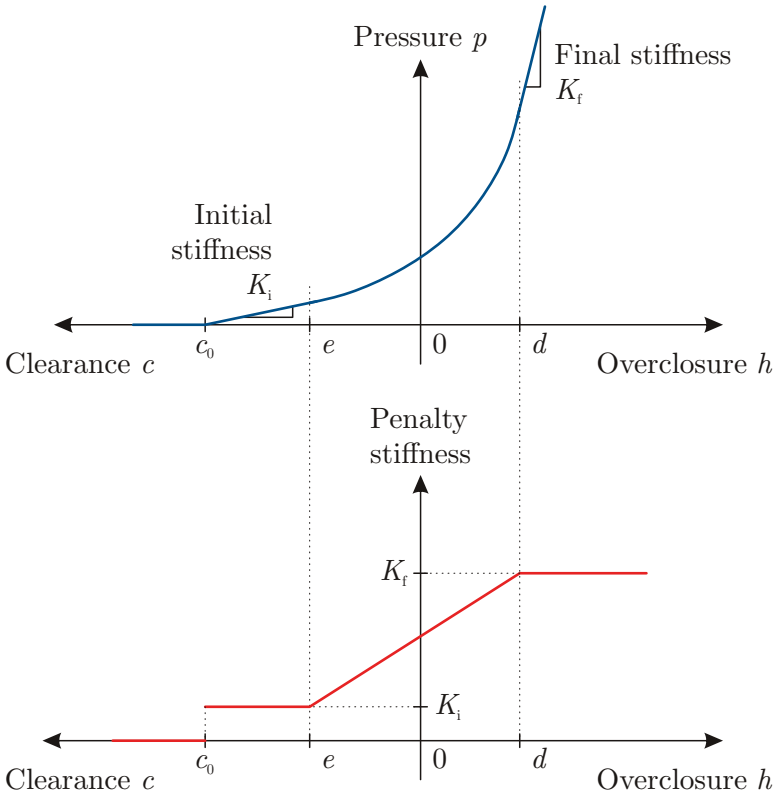

Figure 2.26.: Nonlinear penalty contact, adapted from [1]

2.4.2.2.4. Augmented Lagrange method

As mentioned in Chapter 2.3.4.4.4, the augmented Lagrange method uses the linear penalty method until the penetrations exceed a certain threshold. Then Lagrange multipliers are applied to reduce the penetration. Hence the implementation is identical to the linear penalty method. The user defines the clearance c_0, the linear penalty stiffness K_{lin}, and optionally a scale factor for the default linear penalty stiffness or the linear penalty stiffness K_{lin}, if defined, [1].

2.4.2.3. Behaviour in tangential direction

Additionally to normal forces, shear forces in tangential direction can be transmitted in contact interfaces. The tangential force is a result of friction and can be determined via a friction model using the normal contact force. Abaqus provides a variety of options for modelling friction that are all based on the classical isotropic Coulomb model of friction (see Figure 2.13). It allows for a definition of the friction coefficient in terms of slip rate, contact pressure, surface temperature and other field variables in combination with an optional exponential approximation of the transition from static to kinetic friction coefficients, [1]. The basic Coulomb friction model can de defined in two ways:

- Default model:
  The coefficient of friction is defined as a function of the equivalent slip rate $\dot{\gamma}_{\mathrm{eq}}$, the contact pressure p, the average temperature $\bar{\theta}$ and a predefined average field variable $\bar{f}^{\alpha}$, that is not used by default.

  $$\mu = \mu(\dot{\gamma}_{\mathrm{eq}}, p, \bar{\theta}, \bar{f}^{\alpha}) \tag{2.70}$$

  The average values are gained from the values of the surfaces A and B that are in contact as

  $$\bar{\theta} = \frac{1}{2}(\theta_A + \theta_B) \qquad \text{and} \qquad \bar{f}^{\alpha} = \frac{1}{2}(f_A^{\alpha} + f_B^{\alpha}) \; . \tag{2.71}$$

  In this model the static friction coefficient is given as the value at $\dot{\gamma}_{\mathrm{eq}} = 0$. The maximum value of $\dot{\gamma}_{\mathrm{eq}}$ yields the kinetic friction coefficient.

- Specifying the coefficients:
  With this option the static and kinetic friction coefficient can be defined directly. They can be defined as a function of contact pressure, temperature and field variables (see above), or with the use of an exponential decay law as it was used for this work. The friction coefficient μ is gained as

  $$\mu = \mu_{\mathrm{k}} + (\mu_{\mathrm{s}} - \mu_{\mathrm{k}})\, e^{-d_{\mathrm{c}}\dot{\gamma}_{\mathrm{eq}}} \tag{2.72}$$

  with the kinetic friction coefficient μ_{k}, the static friction coefficient μ_{s}, a user-defined decay coefficient d_{c}, and the slip rate $\dot{\gamma}_{eq}$. For this case, dependencies on contact pressure, temperature and field variables cannot be implemented and the friction has to be isotropic. μ_{s}, μ_{k} and d_{c} can be either defined directly or alternatively in form of test data.

Using test data for exponential model:
At least the points (1) and (2) in Figure 2.27 have to be defined. The inclusion of point (3) is optional.

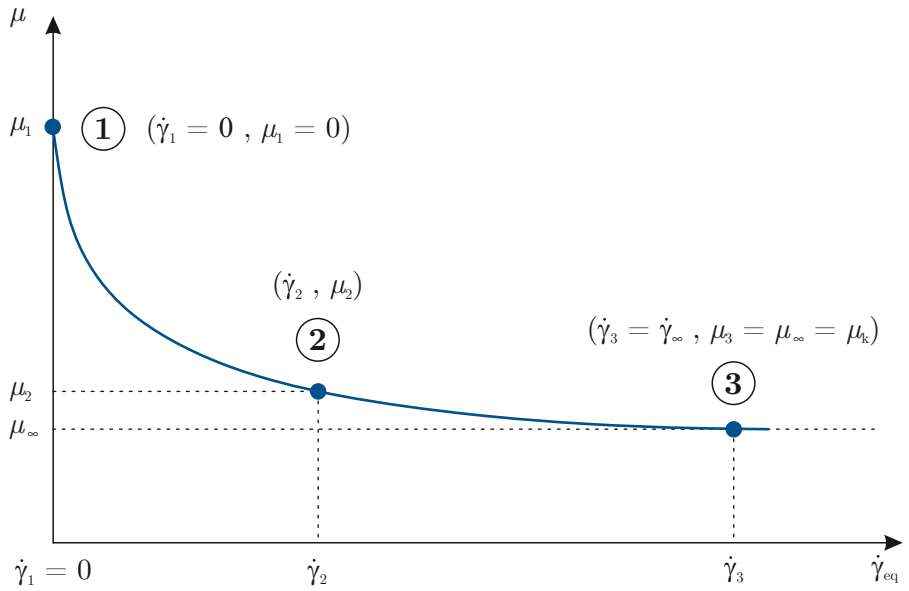

Figure 2.27.: Exponential decay friction model with given data points, adapted from [1]

(1) represents the static friction coefficient at $\dot{\gamma}_{eq} = 0$, and (2) the measured friction coefficient at a reference slip rate $\dot{\gamma}_2$. To fully define the function, point (3) can be added.
If (3) is omitted, Abaqus will assume asymptotical behaviour and automatically determine (3) as $(\dot{\gamma}_\infty, \mu_\infty)$ following

$$\frac{(\mu_2 - \mu_\infty)}{(\mu_1 - \mu_\infty)} = 0.05\ . \tag{2.73}$$

Additional parameters can be added to the friction model to further specify its desired behaviour, such as an equivalent shear stress limit. Anisotropic friction behaviour can be modelled, as well as models with quasi $\mu = \infty$ by using a ROUGH model where slipping is prevented. ROUGH friction means in Abaqus, that a tangential relative motion between the surfaces of the contact interface is impossible while a constraint in normal direction is active, [1].

For the enforcement of tangential contact constraints, Abaqus/Standard uses a penalty method, and Abaqus/Explicit either a penalty or a kinematic constraint method as they were described in Chapter 2.3.4.4 and 2.4.2.2.

The penalty method for frictional constraints is augmented with an additional parameter in form of a maximum allowed elastic slip γ_{crit} to gain a variation of the penalty method, the stiffness method.
The idea is to permit some relative motion of the surfaces (elastic slip) when they should be sticking. Large values of γ_{crit} will lead to good convergence and a rapid solution at the cost of accuracy. For small values of γ_{crit} the solution is very accurate, but problems with convergence can occur as a result, [1].
By default, Abaqus chooses a value for γ_{crit} that should lead to a good compromise between solution efficiency and accuracy. The value is determined based on the characteristic contact surface length calculated by Abaqus. Additionally, the user can manually define a value for γ_{crit}. Large values lead to rapid convergence at the cost of accuracy, and vice versa for small values. Choosing a very small γ_{crit} may result in convergence problems, [1].

3. Methodology

In this chapter the core aspects of the finite element models used for the analysis of creep groan will be presented. Subsequently, the approach for finding a reasonable model and solver configuration for direct time integration in form of a parameter study will be explained.

3.1. Modelling

The starting point for this thesis were two already existing and working FE models implemented in Abaqus/Explicit, as well as results of creep groan experiments that were presented in [20].
Since the aim of this work was to compare the results and efficiency of explicit and implicit direct time integration, the models and solvers had to be modified and fitted for simulations in Abaqus/Standard. The first version is a very simplistic model only consisting of a brake disk and two brake pads, while the second one is much more complex and considers the entire axle-corner. Both were modelled with the pre-processor ANSA and will be explained in detail later in this chapter. The aim was to find a reasonable configuration using the simpler model, and then to apply the knowledge gained to the more complex one.

3.1.1. Disk Pad Model - DPM

This model only consists of a brake disk and two brake pads and was thus named "DPM" (Disk Pad Model). The goal was to develop a very simple model that is capable of modelling the stick-slip effect responsible for creep groan. It is depicted in Figure 3.1. In order to achieve this goal, sacrifices regarding the accurate representation of realistic physical parameters (e.g. brake pressure) had to be made. It is important to note, that the DPM is a strictly academic model without any connections to experiments. The stick-slip frequencies found are not representative of realistic behaviour.
The disk consists of 280 RIGID-elements combined to one rigid body with an inner and outer diameter of 200 mm and 330 mm. The brake pads left and right are modelled as two layers of first order elements: an outer layer with 248 elements (242 hexas + 6 pentas) and an inner layer with 374 elements (372 hexas + 2 pentas). A single brake pad is shown in Figure 3.2.

This adds up to a total of 1244 solid elements (1228 hexas + 16 pentas) for both pads plus the disk, which is one single rigid body and a point mass. The total mass of the model is 9.58 kg. As Figure 3.1 shows, the coordinate system in the FE model does not match the ISO-vehicle coordinates. Note that the model represents the left front brake system of a car and thus the pads are located on the front side.

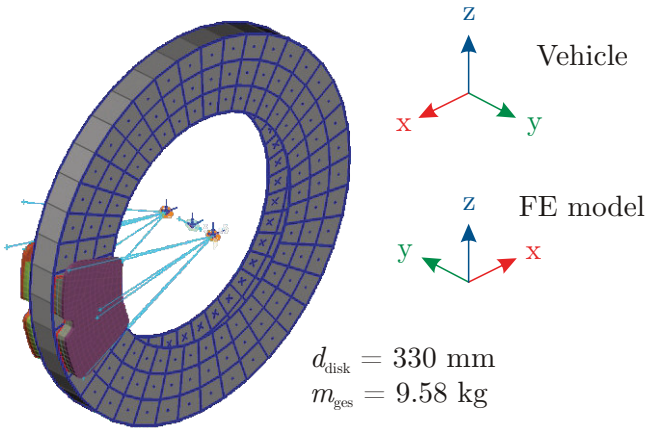

Figure 3.1.: Disk Pad Model DPM

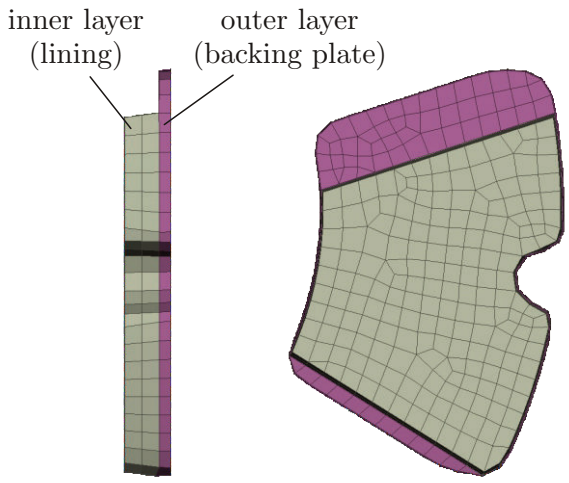

Figure 3.2.: Mesh of one brake pad of the DPM

Mesh quality:
Information about the quality of the elements and the overall mesh can be easily accessed with the "deck report" function in ANSA.
For the definition of common mesh quality criteria, ANSA offers several options to choose from. For this work, the definition of the aspect ratio according to "Patran" was used and is depicted for Penta and Hexa elements in Figure 3.3.

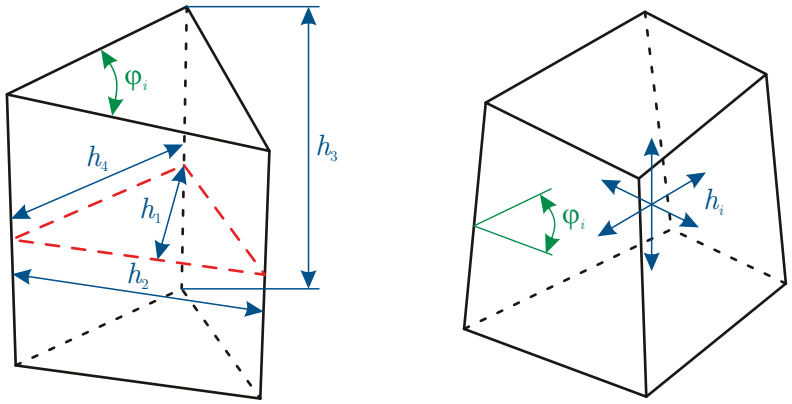

Figure 3.3.: Length and angle of penta and hexa elements, adapted from [6]

The following quality criteria will be used:

$$h_{\min} = \min(h_i) \quad ; \quad h_{\max} = \max(h_i) \tag{3.1}$$

$$\text{Aspect ratio for Penta} : \quad \mathrm{AR_P} = \frac{h_4}{h_3} \cdot \frac{\sqrt{3}h_2}{2h_1} \tag{3.2}$$

$$\text{Aspect ratio for Hexa} : \quad \mathrm{AR_H} = \frac{h_{\max}}{h_{\min}} \tag{3.3}$$

$$\varphi_{\min} = \min(\varphi_i) \quad ; \quad \varphi_{\max} = \max(\varphi_i) \; . \tag{3.4}$$

Additionally, the "Jacobian deviation" quality criterion was used. It is applicable to first and second order elements and its definition is provided in [6]. For each integration point, the determinant of the Jacobian matrix is calculated. Integration points are based on the element type, e.g. a first order Hexa element has 8 integration points.

The final value for the Jacobian deviation is then gained as the ratio between the smallest value over the largest. A value equal to 1 represents the ideal shape of the element. For concave elements, the Jacobian deviation becomes negative.

The Hexa and Penta elements of the mesh shall be checked with regards to their aspect rations, minimum/maximum lengths (see Table 3.1) and angles (see Table 3.2), and their Jacobian deviations (see Table 3.3).

Table 3.1.: Quality criteria associated with length for the DPM

Element type	Penta	Hexa	Total
Number of elements (NoE)	16	1228	1244
$h_{\min}$ [mm]	2.7	2.12	2.12
$h_{\max}$ [mm]	6.52	7.41	7.41
Mean length [mm]	4.08	4.53	4.52
$\mathrm{AR}_{\max}$	2.79	2.31	2.79

Table 3.2.: Quality criteria associated with angle for the DPM

Element type	Penta	Hexa	Total
$\varphi_{\min}$ [°]	23.3	41.88	23.3
$\varphi_{\max}$ [°]	109.25	150.7	150.7
NoE with $\varphi_{\min} < 30$ °	2	0	2
NoE with $\varphi_{\max} > 100$ °	2	492	494
NoE with $\varphi_{\max} > 120$ °	0	148	148
NoE with $\varphi_{\max} > 140$ °	0	8	8

Table 3.3.: Jacobian deviation criteria for the DPM

Jacobian deviation	Penta	Hexa	Total
Min. Jacob.	0.903	0.707	0.707
Max. Jacob.	0.996	0.995	0.996

For the limits of the respective criteria, the suggested values in ANSA were kept and are the following for solid elements:

- Aspect ratio: 8
- Jacobian: 0.7
- Min. angle Penta: 30°
- Max. angle Penta: 120°
- Min. angle Hexa: 30°
- Max. angle Hexa: 140°

When applying these limits one can see, that the aspect ratio and Jacobian deviation are acceptable for every element in the mesh. 2 out of 16 Penta elements are below the threshold of 30°, and 8 out of 1228 Hexa elements are above 140°. This results in a total of 10 "off" elements out of 1244 elements in the mesh, or 0.8%.
The most significant criteria for the mesh quality were found to be aspect ration and Jacobian deviation and are met by all elements. The angle tolerance is only of secondary importance and thus the very few "off" elements can be disregarded. The mesh was deemed acceptable for simulations.

For assessing the quality of the mesh, another crucial aspect is the side length of the solid elements, since this is what the determination of many parameters of the solver Abaqus is based on. For example, the time increment size of explicit analyses is based on the minimum element length and stiffness in the model.

Additionally to the general properties of the mesh, the connector elements, boundaries and constraints will be explained, as well as the application of the loads and the friction model that was used.

Loads:
In order to achieve a stick-slip motion, the loads have to be applied in a similar way as it was described in Chapter 2.1.1.
The term "brake pressure" is generally understood as the hydraulic pressure in the brake system that is then applied to the brake piston. For this model, the pressure is applied to the outer surface of the brake pads directly, which is a larger surface than the piston would be. This means, that the pressure applied in the DPM does not represent the actual hydraulic brake pressure and thus will be referred to as "pad pressure" p_{P} instead.

For this model, the following sequence was implemented:

1. The pad pressure p_{P} is applied and and held constant (① to ② in Figure 3.4). The brake is closed.
2. The torque M_{drive} is applied with a linearly increasing magnitude (① to ②), and then held constant (② to ③), according to Figure 3.4. The brake is still closed.
3. Now the pad pressure is linearly decreased (② to ③) until a magnitude is reached, where the stick-slip effect starts to take place.
4. The pad pressure is then again kept constant until the end of the simulation (③ to ④).

This method represents the scenario of a car standing on an inclined road and then slowly releasing the brakes.

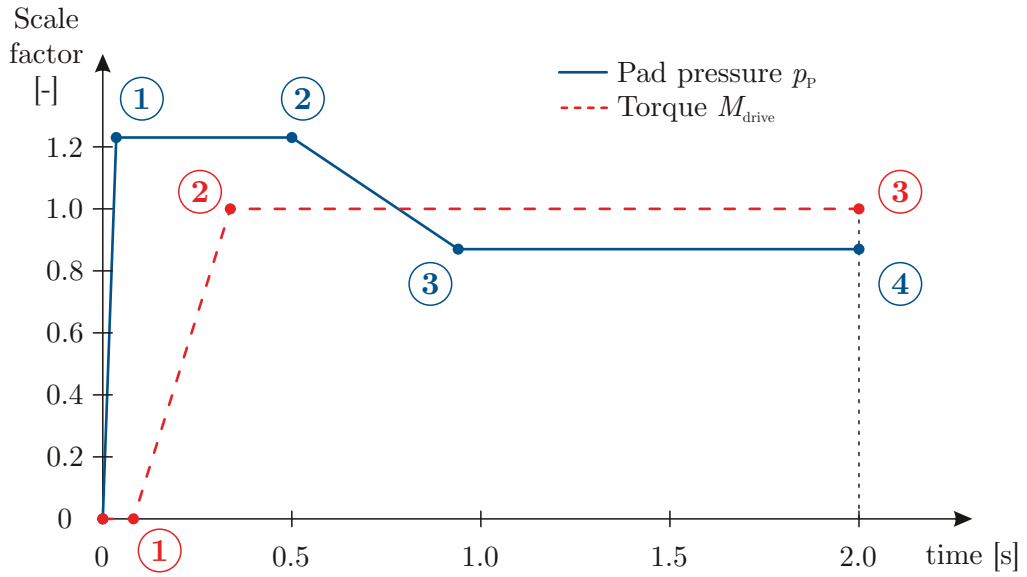

Figure 3.4.: Amplitude function (scale factor) for pad pressure and torque for the DPM

Defining loads in Abaqus:
The user can define both a magnitude of whichever load should be added, as well as a variable scale factor in form of an amplitude function, e.g. as depicted in Figure 3.4. The effective magnitude at a certain time during the analysis is then gained by multiplying the constant value by the variable scale factor defined via the amplitude function. The initial constant values used in the DPM are:

- Pad pressure: $p_P = 0.01834\ \mathrm{N/mm^2}$
- Torque: $M_{drive} = 12.99\ \mathrm{Nm}$

The values defining the amplitude functions in Figure 3.4 can be found in Table 3.4.

Table 3.4.: Values for points in Figure 3.4

Point	Pressure		Torque	
	Time [s]	Scale factor	Time [s]	Scale factor
①	0.01	1.25	0.03	0
②	0.5	1.25	0.33	1
③	0.942	0.87	2.0	1
④	2.0	0.87	-	-

The torque M_{drive} is applied in form of a concentrated load (CLOAD) to the brake disk, the pad pressure p_{P} as a distributed load (DLOAD) to the outer surfaces of the outer layer of both brake pads (see Figure 3.5).

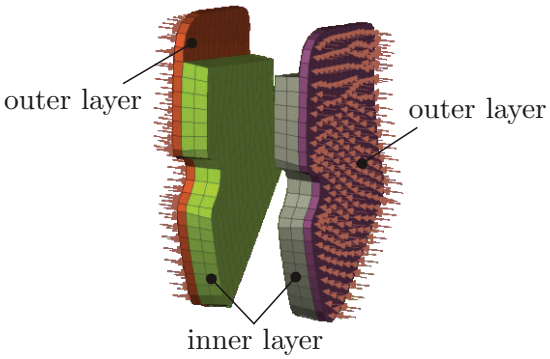

Figure 3.5.: Pad pressure as distributed load on the brake pads (brake disk not displayed)

Friction:

The friction in the contact interface between disk and pads was modelled in accordance with the exponential decay model utilizing test data described in Chapter 2.4.2.3. The following values were used for the definition of the exponential decay function, see Figure 2.27:

(1) : $\mu_1 = 0.5$
(2) : $\mu_2 = 0.45$ at $\dot{\gamma}_2 = 10$ mm/s
(3) : $\mu_3 = 0.4$

Contact surface:

Two nonlinear contact interactions were defined for the inner and outer pad with the disk respectively. Abaqus' GENERAL CONTACT approach was used with the inner and outer surface of the brake disk as master surfaces, and the inner layer surfaces of the brake pads as slave surfaces. A default surface interaction with only the above friction model was assigned to either contact interaction. No further changes were made to the default settings.

Material assignment:

For the brake pads, a linear elastic, orthotropic material behaviour was defined. The material parameters listed in Table 3.5 were assigned to the inner and outer layer (see Figure 3.2) of both pads.

Table 3.5.: Material parameters assigned to the brake pads

Material parameter	Value
Young's modulus E_1	10754.8 N/mm^2
Young's modulus E_2	10754.8 N/mm^2
Young's modulus E_3	3549.1 N/mm^2
Shear modulus G_{12}	4301.9 N/mm^2
Shear modulus G_{13}	2920.1 N/mm^2
Shear modulus G_{23}	2920.1 N/mm^2
Poisson's ratio ν_{12}	0.25
Poisson's ratio ν_{12}	0.1
Poisson's ratio ν_{12}	0.1

The indices 1, 2 and 3 in Table 3.5 represent the local material directions x, y and z, with z being the direction normal to the friction surface.

A slightly different density was defined for the inner and outer layer of the pads respectively, as:

- Density of outer layer (backing plate): $\rho_{\text{top}} = 2539.58$ kg/m^3
- Density of inner layer (lining): $\rho_{\text{bot}} = 2484.7$ kg/m^3

For the point mass assigned to the brake disk, the parameters listed in Table 3.6 were used.

Table 3.6.: Material parameters assigned to the brake disk

Material parameter	Value
Mass m_{disk}	9.30067 kg
Rotary inertia I_{11}	75583 kgm^2
Rotary inertia I_{22}	148935 kgm^2
Rotary inertia I_{33}	75602 kgm^2
Product of inertia I_{12}	1.7493 E-03 kgm^2
Product of inertia I_{13}	3.3728 E-03 kgm^2
Product of inertia I_{23}	10806.9 E-03 kgm^2

In addition to the material behaviour, a connector behaviour had to be defined for the elements connecting the brake pads and brake disk to the rotational axis respectively. Damping and elasticities were added to the connector behaviour. Values for up to all 6 degrees of freedom (DOFs) can be defined. Abaqus numbers the DOFs from 1 to 6, with the three translational DOFs 1, 2, 3 and the three rotatory DOFs 4, 5, 6, see Figure 3.6.

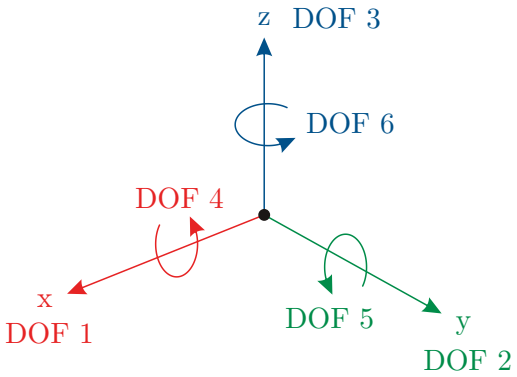

Figure 3.6.: Correlation between DOFs in Cartesian coordinates and DOFs in Abaqus notation

The following parameters were assigned:

- Damping of pad connector:
  - Rotatory linear viscous damping coefficient about z: 0.3897 Nm · s/rad
  - Other DOFs without damping
- Elasticity of pad connector:
  - Rotatory linear elastic stiffness about z: 256.0252 Nm/rad
  - Other DOFs without elasticity (= rigid)
- Damping of disk connector:
  - Rotatory linear viscous damping coefficient about z: 7.014 Nm · s/rad
  - Other DOFs without damping
- Elasticity of disk connector:
  - All DOFs without elasticity (= rigid)

Now that every relevant aspect of the model has been defined, a solution method can be implemented.

3.1.1.1. DPM in Abaqus/Explicit

The solution with explicit direct time integration in Abaqus/Explicit allows for a fairly automatized implementation of the solver. The time incrementation parameter "element by element" was used, so only the total time period T has to be defined. Other options were omitted or left at default settings.

Total time period: $T = 2.0$ s
Time increment: $\Delta t = 5.523$ E-07 s (computed by Abaqus/Explicit)

For contact simulations, the penalty method is used by default and its parameters are determined automatically by Abaqus/Explicit based the minimum element length in the model and its material properties.

The parameters and results that should be printed to the output file have to be added to the model in form of output requests. Some outputs are separated in components in Cartesian coordinates and the identifier $i = 1, 2, 3$ shall be used to identify them ($i = 1$ is equivalent to the x-direction, $i = 2$ to y, and $i = 3$ to z). For the model at hand, the following outputs were requested and evaluated:

- Node outputs (For a selection of potentially relevant nodes):
  - COORDi: Coordinates of the node in x, y, z
  - Ui: Displacement of the node in x, y, z
  - Vi: Velocity of the node in x, y, z
  - Ai: Acceleration of the node in x, y, z
- Contact outputs (for inner and outer contact nodes):
  - CSTATUS: Contact status for general contact analyses

Additionally a large number of outputs related to energies and contact parameters were generated, but are not mentioned here, since they were not used for the matter of this thesis. A comprehensive list of all output variables requested can be found in the appendix. All outputs were generated with an output frequency of 1000 Hz.

3.1.1.2. DPM in Abaqus/Standard

After the DPM was already successfully simulated in Abaqus/Explicit, the transition to Abaqus/Standard was made. The following fundamental changes were mandatory:

- Contacts have to be defined as "model data" instead of "history data" for simulations in Abaqus/Standard. These are Abaqus-specific terms and for more information see [1].
- A suggested initial time increment size Δt_0 has to be defined.

- A maximum number of time increments $n_{\text{inc max}}$ has to be defined. If omitted, default settings are used. In this case, the smaller value of either the suggested Δt_0 or 10^{-5} times the total simulation time is chosen. For the DPM the default value would be $2 \cdot 10^{-5}$ s.

Output parameters related to contact energies are unique to Abaqus/Explicit and Abaqus/Standard respectively. The previously used outputs have to be replaced for implicit simulations and can be found in the appendix.

Initially, problems with the convergence of the solution occurred as soon as the first transition from global stick to slip started to take place. The initial time increment of $\Delta t_0 = 0.001$ s is reduced drastically by the automatic time increment control during this phase. Thus, the minimum allowed increment size had to be set to a very small value. Moreover, the default setting for the maximum amount of cutbacks of the increment size allowed per increment was found to be insufficient and had to be increased.

Reliable simulations were achieved with the following settings:

- Maximum number of increments: $n_{\text{inc max}} = 5000$
- Initial increment size: $\Delta t_0 = 0.001$ s
- Minimum allowed increment size: $\Delta t_{\text{min}} = 10^{-12}$ s
- Maximum amount of cutbacks per increment: $I_{\text{A}} = 20$ (default: $I_{\text{A}} = 5$ was sufficient in most cases)
- ANALYSIS DISCONTINUOUS (predefined setting in Abaqus/Standard)

The setting ANALYSIS DISCONTINUOUS can help improve severely discontinuous behaviour by changing settings associated with convergence criteria. The following parameters are changed, [1]:

- Number of equilibrium iterations after which the convergence of the residuals in two consecutive iterations is checked: $I_0 = 8$ (default: $I_0 = 4$)
- Number of consecutive equilibrium iterations at which logarithmic rate of convergence check begins: $I_{\text{R}} = 10$ (default: $I_{\text{R}} = 8$)

After these changes were made, the model was fit for simulations in Abaqus/Standard. A variety of modifications to the model were then performed in order to find an optimal configuration for the simulation of creep groan (see Chapter 3.2). The results will be presented in Chapter 4.2.

3.1.2. Axle-corner Model - ACM

The full Axle-corner Model, or ACM, is much more complex than the Disk Pad Model DPM and represents realistic physical behaviour a lot better. It is an improved version of the model that was used in [20] and was, in its basic version, provided by the supervisor. The model represents the whole MacPherson axle setup on the left side and is depicted in Figure 3.7.

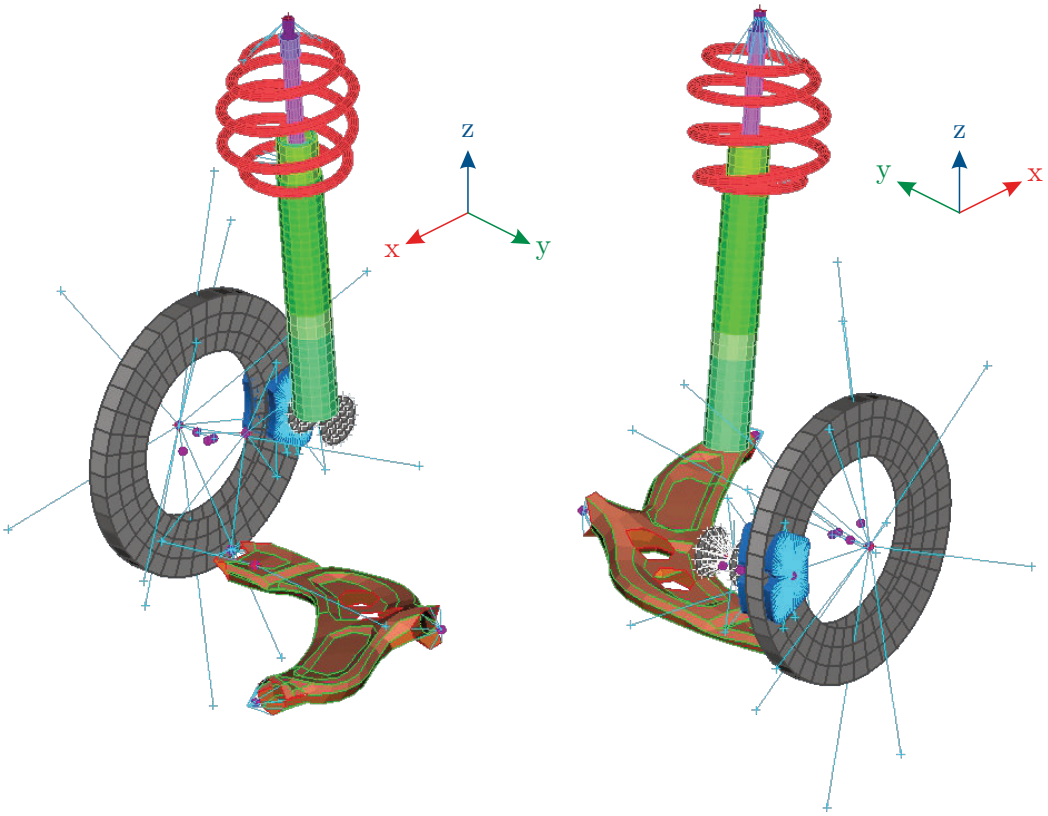

Figure 3.7.: Axle-corner Model ACM

Components of the ACM:

Compared to the DPM, which only consists of the brake disk and pads, the ACM has many more components. Table 3.1.2 provides an overview of these parts and their FE-representation used in this model. "Point inertia" is a term used by Abaqus and means, that a point mass with translational and rotatory inertia was assigned to the respective element.

Table 3.7.: Components of the Axle-corner Model ACM

Part		**Model / Element**
Disk		Rigid shells + point inertia
Rim		Kinematic coupling + point inertia
Tyre		Spring and damper + point inertia
Pads	Linings	Solid continuum elements
	Backing plates	Kinematic coupling + point inertia
Caliper		Rigid shells + kinematic coupling + point inertia
Brake piston		Rigid shells + kinematic coupling + point inertia
Anchor bracket		Kinematic coupling + point inertia
Wheel carrier		Kinematic coupling + point inertia
Lower control arm		Shell continuum elements
Strut	Coil spring	Timoshenko beams
	Damper tube	Timoshenko beams
	Damper piston	Timoshenko beams
Steering link		Rigid + point inertia
Elastomer bushings		Spring and damper elements (lower control arm rear hydro bushing with nonlinear, displacement-dependent stiffness characteristics)
others		Kinematic / elastic connector elements between components

Boundaries:

For the connection to the vehicle body or test rig, boundaries have to be defined. Here, 4 boundary conditions at the strut, the control arm, and the steering link were added, as it is depicted in Figure 3.8. All 6 degrees of freedom are locked in these points. Thus, influences of the vehicle body are neglected.

Mesh quality:

The ACM consists of a total of 1584 first order solid elements (1512 hexa elements, 60 penta elements, and 12 tetra elements) and 316 first order shell elements (257 quad elements, and 59 tria elements). The minimum element length in the model is 1.503 mm and has a total mass of 52.676 kg.

The assessment of the mesh quality was performed analogous to the DPM, using the same quality criteria. Where the DPM only consists of solid elements, both solid

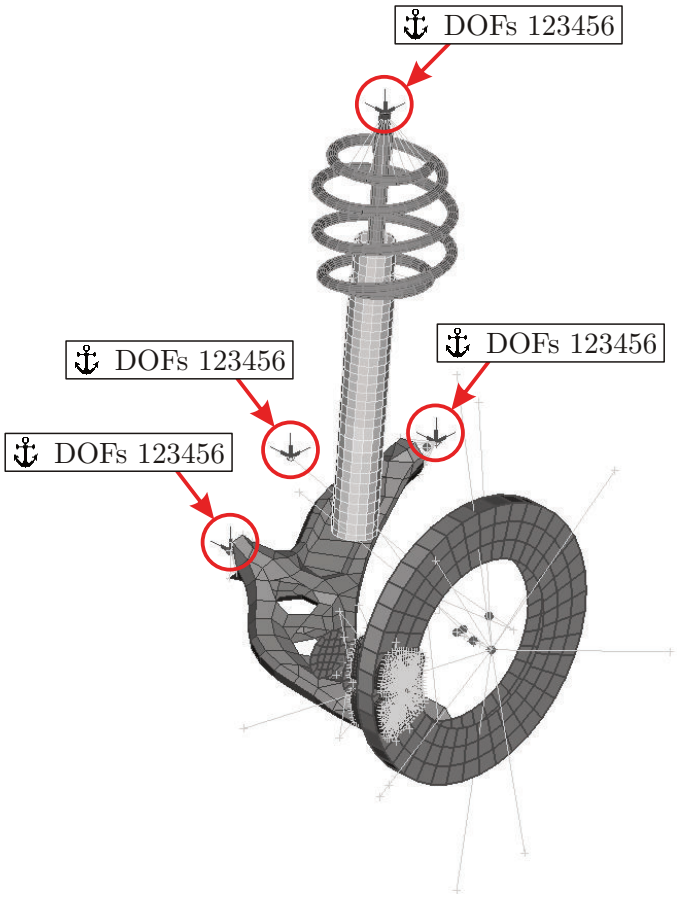

Figure 3.8.: Boundaries of the ACM

elements and shell elements are used in the ACM. Shell elements were used to roughly model the lower control arm, its stiffness and mass. It is thus only subsidiary for the investigation of creep groan and therefore quality criteria are mostly irrelevant for the shell elements.
The solid elements of the brake pads were checked with regards to their aspect ratios and Jacobian deviations. In Table 3.8 the respective minimum and maximum values are listed.

Table 3.8.: Quality criteria of solid elements in the ACM

Element type	Penta	Hexa	Tetra	Total	Limit
AR_{max}	3.87	2.97	3.03	3.87	8
Jacob. min.	0.76	0.75	1	0.75	0.7

As Table 3.8 shows, all elements are within the limits of the respective quality criteria.

Differences to the DPM:

Similar to the DPM, the brake disk is modelled as a single rigid body and a point mass. Besides the disk, almost every other aspect of the model was either modified, or newly added in general. The most significant changes were made to:

- the brake pads,
- the load application, and
- the friction model.

Brake pads / linings:

The geometry of the brake pad linings was changed to match the real shape of the pads and linings used for experimental investigations in [21], [20]. It is now modelled as a single body instead of two separate layers and a chamfer was added. It is depicted in Figure 3.9. In [23], the effects of the geometry of the chamfer on the creep groan using FEA were investigated. It was found, that the chamfer size influences tangential acceleration, which is suggested to be correlated with advantageous creep groan behaviour, [23].

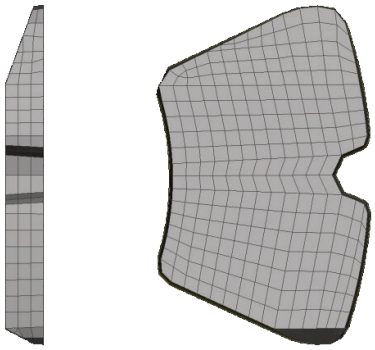

Figure 3.9.: Mesh of one brake pad lining of the ACM

As one can see, the mesh has been changed compared to the DPM due to its new shape. A single lining consists of a total of 792 first order elements (756 hexa elements, 30 penta elements, and 6 tetra elements) with a mean side length of 4.39 mm.
The assigned material is identical to the material of the inner layer of the pads in the DPM. The material parameters can be found in Table 3.5, a density of $\rho_{\text{pad}} = 2484.7\ \text{kg/m}^3$ was defined.

Load application:

As a reminder, in the DPM the torque was applied directly to the disk and the "brake pressure" was applied directly to the outer surface of the pads as a so-called pad pressure.

Since this method does not represent realistic circumstances that occur on disk brakes, the application of the loads was changed fundamentally in the ACM. A brake caliper was modelled, that allows for the brake pressure to be applied to circular rigid planes of itself and the brake piston in form of a distributed load (DLOAD). It is then transmitted via connector elements to the brake pads. The amplitude function applies the full brake pressure within the first 0.05 s of the simulation, which is then held constant.

The direct application of the torque was replaced by a velocity excitation model. This was done to model an excitation similar to the velocity-controlled test rig experiments conducted in [21], [20] through the tyre including its longitudinal, lateral and vertical stiffness and damping. Instead of a directly applied torque in the DPM, a velocity is prescribed to a node representing the contact between tyre and road in the ACM. This accurately models the circumstances at the test rig, where the tyre is driven by a drum. For more details about the test rig see [21], [20]. The velocity model is illustrated in Figure 3.10.

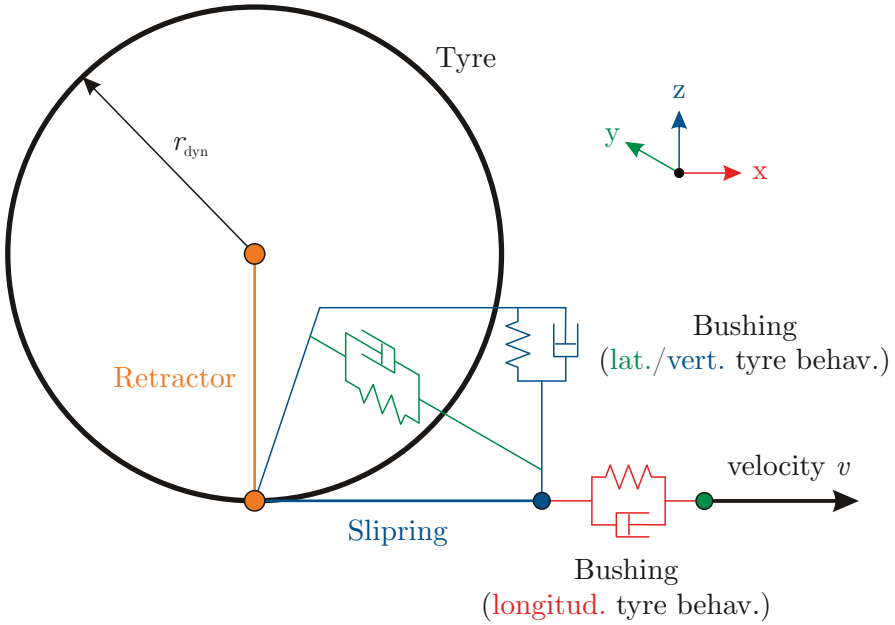

Figure 3.10.: Velocity excitation model with tyre elasticity and damping

As one can see, the point at which the velocity v is applied, is connected via a Retractor and a Slipring connector element to the axis of the wheel. These elements are necessary to apply both the torque and the longitudinal force correctly to the model. A detailed explanation about these elements can be found in [1]. Their main functions are:

- Retractor element:
  Join the positions of two nodes (DOFs 1, 2, 3) and convert material flow into rotation (new DOF 10). A Retractor functions as a join + flow-converter.

- Slipring element:
  Models material flow and stretching between two points of a belt system.

It can be understood similarly to the unwinding of a cable drum (see Figure 3.11).

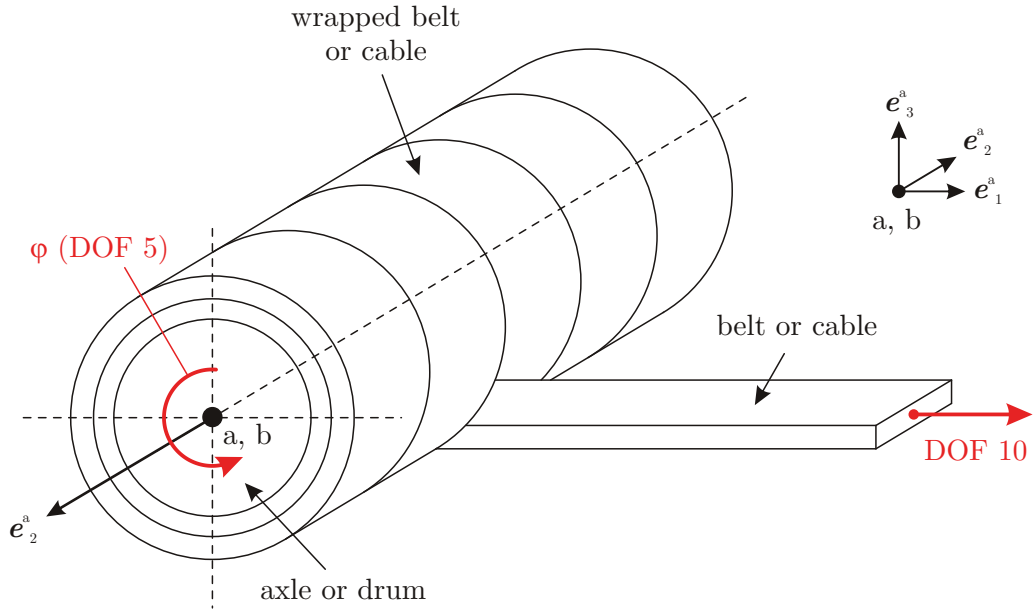

Figure 3.11.: Connection type flow-converter; analogy to cable drums; adapted from [1]

The kinematic connector constraint $ur_2 = 0$ of a flow-converter is given as

$$ur_2 = \boldsymbol{e}_2^{\mathrm{a}} \cdot (\varphi_{\mathrm{a}} - \varphi_{\mathrm{b}}) - \beta_{\mathrm{s}} \boldsymbol{\Psi}_{\mathrm{b}} = 0 \tag{3.5}$$

with the local direction $\boldsymbol{e}_2^{\mathrm{a}}$, the relative rotation $(\varphi_{\mathrm{a}} - \varphi_{\mathrm{b}})$ between node a and b, a scale factor β_{s}, and the material flow at node b $\boldsymbol{\Psi}_{\mathrm{b}}$.

Additionally, bushing elements are included in all three directions to account for the three-dimensional elastic properties of the tyre.

The amplitude function for the application of the velocity is depicted in Figure 3.12. It as applied at 0.1 s, linearly increased until 0.5 s, then held constant for the remaining 1.5 s.

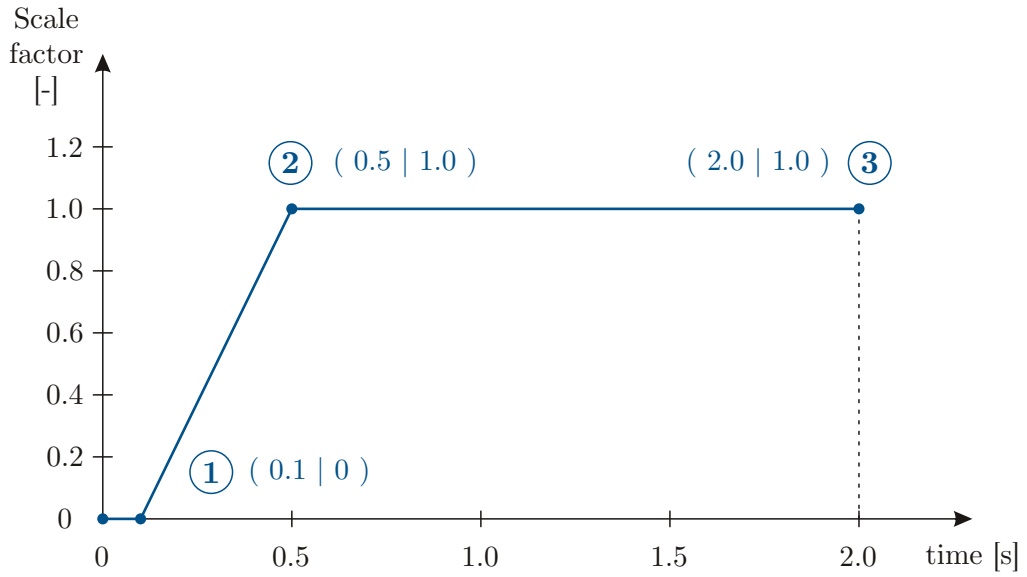

Figure 3.12.: Amplitude function for the application of the velocity

Friction model:

Based on the results of the creep groan experiments presented in Chapter 3.3, two friction models were derived for the ACM. The respective brake pressure and velocity used at the test rig were combined with various friction parameters to produce stick-slip frequencies similar to 21 Hz and 87 Hz. This was achieved with the friction models illustrated in Figure 3.13. As one can see, the model was defined as an exponential decay law (see Figure 2.27).

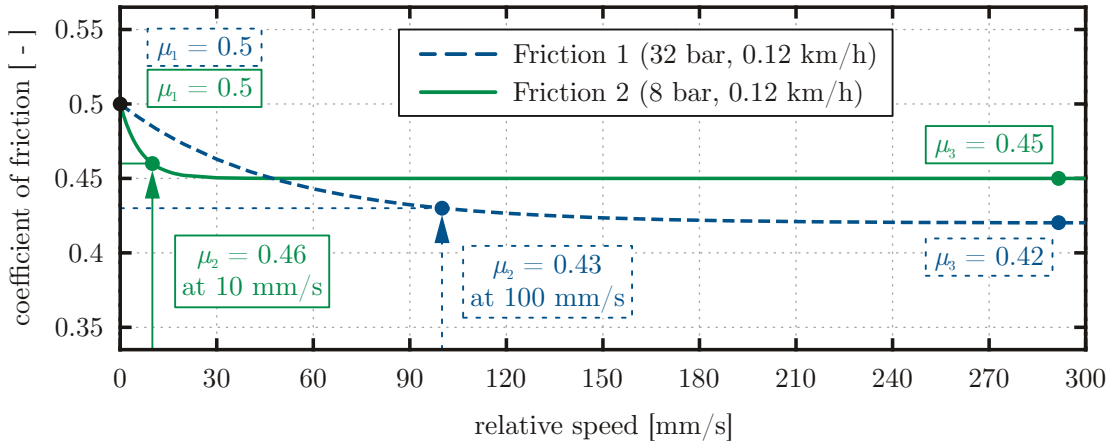

Figure 3.13.: Friction models derived from test rig experiments, adapted from [20]

The ACM was implemented with these configurations in order to have means of validation of the results.

3.1.2.1. ACM in Abaqus/Explicit

The solver was implemented similarly to the DPM in Chapter 3.1.1.1. The total time period for the analysis was set to $T = 2.0$ s with "element by element" time incrementation. A time increment of $\Delta t = 3.61221$ E-07 s was computed by Abaqus/Explicit. For the enforcement of contact constraints, the penalty method is used by default with automatically determined parameters.

Several additional output variables related to reaction forces and moments were requested and can be found in the appendix. Outputs were generated with an output frequency of 1000 Hz.

3.1.2.2. ACM in Abaqus/Standard

For the transition of the ACM to Abaqus/Standard, the same changes as for the DPM had to be made according to Chapter 3.1.1.2.

Due to the greatly increased complexity of the ACM compared to the DPM, severe convergence issues occurred. Additionally, highly non-physical behaviour was observed upon investigation of the results in the post-processor META and thus the model had to be modified. The following changes were found to significantly improve the solution:

- Connector damping for SATTELAUFLAGE added:
  - Linear, viscous damping for DOF 4: 100000 kg/s
  - Linear, viscous damping for DOF 5: 100000 kg/s
  - Linear, viscous damping for DOF 6: 100000 kg/s
- Connector elasticity for RL KUGEL modified:
  - Linear, elasticity for DOF 4 changed to RIDIG (before: 10^{13} N/m)
  - Linear, elasticity for DOF 6 changed to RIDIG (before: 10^{13} N/m)
- Connector damping for PINS modified:
  - Linear, viscous damping for DOF 1 changed to 400 kg/s (before: 200 kg/s)
- Initial gap between the brake pads and disk of 0.05 mm removed

The changes listed above were also applied to the explicit ACM to ensure an accurate comparison between the implicit and explicit solutions. The initial gap between disk and pads of 0.05 mm was kept in the explicit model.

3.2. Parameter study

Now that both explicit models were successfully implemented in Abaqus/Standard, a parameter study can be conducted. Certain parameters of the implicit models were modified, that were suspected to potentially influence solution time and accuracy. The results were then compared to one another and to the results of the explicit simulations. This was done extensively for the minimal model since the computation times were significantly faster and a large number of modifications could be tested. Following conclusions were then applied to the Axle-corner Model.

3.2.1. Disk Pad Model - DPM

Upon closer examination of the results of the explicit simulation that were provided by my supervisor, questionable behaviour was observed for certain output parameters. Several output variables were found to be superimposed by a high-frequency noise in the earlier stages of the simulation before the beginning of the stick-slip phase.
The reason for this was identified as an incorrect assignment of the point mass to the brake disk and was subsequently corrected. The noise was successfully removed. However, this change affected the model in a way, that a stable stick-slip motion could no longer be achieved. Thus, changes to the load application were made, namely the amplitude function of the brake pressure was modified. As one can see in Figure 3.14, the slope was left unchanged, but the scale factor was increased to a higher constant value in the later phase of the simulation. The new values are listed in Table 3.9.

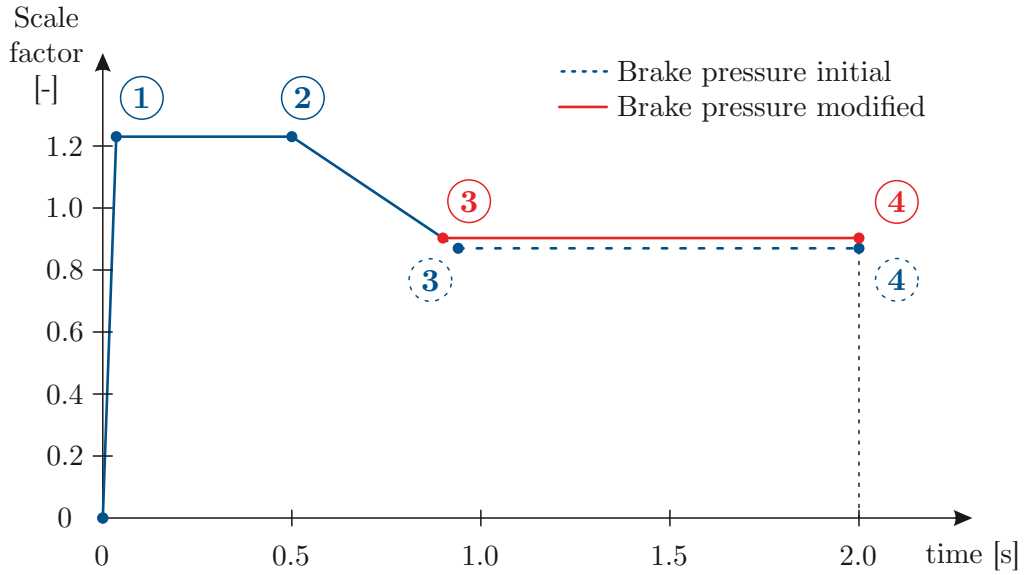

Figure 3.14.: Changes to the amplitude function of the brake pressure in the DPM

Table 3.9.: Old and new values of the amplitude function for the brake pressure

Point	Time [s]		Scale factor	
	initial	modified	initial	modified
③	0.942	0.9071	0.87	0.9
④	2.0	2.0	0.87	0.9

With these changes, the collapse of the stick-slip motion could not be removed entirely, but it could be delayed to a later point in time. This makes a total of 3 different variants for the DPM in Abaqus/Explicit:

- Variant 01: Initial model with noise (DPM_Expl_01)
- Variant 02: Correct mass assignment (DPM_Expl_02)
- Variant 03: Correct mass assignment + new amplitude function (DPM_Expl_03)

The results for the variants above will be presented in detail in Chapter 4.2.

For the implicit simulation in Abaqus/Standard, the following aspects of the model were assumed to be of significant importance for convergence, accuracy, and overall computing time of the solution:

- Contact definition approach:
  - GENERAL CONTACT
  - CONTACT PAIR
- Surface interactions:
  - Augmented Lagrange method
  - Penalty method with linear penalty function
  - Penalty method with nonlinear penalty function
  - Direct method with linear pressure-overclosure relationship
  - Direct method with hard pressure-overclosure relationship
- Solver settings:
  - MODERATE DISSIPATION (HHT-MD)
  - TRANSIENT FIDELITY (HHT-TF)

A detailed explanation of the aspects above is provided in Chapter 2.3. Additionally, the influence of the amount of numerical dissipation on the solution, governed by the parameter α in the solver settings, was included in the parameter study.

For the following values of α, simulations were performed and analysed:

- $\alpha = 0$ (no numerical dissipation)
- α = -0.333 (maximum numerical dissipation)
- α = -0.2 (moderate numerical dissipation)

Furthermore, Abaqus offers the choice of how severe discontinuities should be treated throughout an analysis. According to [1], Abaqus/Standard distinguishes between two different types of iterations: equilibrium iterations and severe discontinuity iterations (SDIs). An iteration is considered an SDI, when sudden changes in stiffness occur. This is usually caused by changes in contact status (open/closed, stick/slip). For creep groan simulations constant changes between sticking and slipping are expected to occur throughout the analysis and thus this setting was found to potentially have a significant impact on the solution. It was therefore included in the parameter study. The setting regarding the treatment of SDIs can be found in the step-definition window, an explanation is provided in [1]. The following options are available:

- Convert SDI = YES (default):
  If SDIs are detected, Abaqus/Standard will continue to iterate until they are sufficiently small using local convergence criteria based on the maximum penetration and force errors. If the severe discontinuities are sufficiently small and the equilibrium tolerances are satisfied, the iteration converges.
- Convert SDI = NO:
  If SDIs are detected, a new iteration is forced regardless of the amount of penetration or force errors that occur. Additionally, this setting changes various aspects of the time incrementation and convergence criteria in Abaqus/Standard which are listed below. According to [1], this setting may lead to convergence problems or the occurrence of contact chattering.

Changes to convergence and incrementation parameters due to the "Convert SDI" setting according to [1] are:

- T^{cont}: Contact and slip compatibility tolerance (default: $T^{cont} = 0.005$)
  - Convert SDI = YES:
    The ratio of the maximum error in the contact or slip constraints to the maximum displacement increment must be less than this tolerance.
  - Convert SDI = NO:
    Used only with softened contact pressure-overclosure relationships. The ratio of the error in the soft contact constraint clearance to the user-specified clearance at which the contact pressure is zero must lie below this tolerance for $p > p^0$, where p^0 is the pressure value at zero clearance and p the current pressure.

- T^{soft}: Soft contact compatibility tolerance for low pressure (default: $T^{soft} = 0.1$)
  - Convert SDI = YES:
    Not used for this setting.
  - Convert SDI = NO:
    Similar to T^{cont}, but it represents the tolerance when $p = 0.0$. The actual tolerance is interpolated linearly between T^{cont} and T^{soft} for $0 \leq p \leq p^0$.
- T^{cfe}: Contact force error tolerance: (default: $T^{cfe} = 1.0$)
  - Convert SDI = YES:
    The ratio of the maximum error in the contact force to the time average force must be less than this tolerance.
  - Convert SDI = NO:
    Not used for this setting.
- I_S: Maximum number of SDIs allowed in an increment. (default: $I_S = 12$)
  Only used for SDI = NO.
- I_J: Maximum number of SDIs allowed in two consecutive increments for the time increment to be increased. (default: $I_J = 6$)
  Only used for SDI = NO.
- I_J^c: Maximum number of equilibrium and severe discontinuity iterations allowed in two consecutive increments for the time increment to be increased. (default: $I_J^c = 50$)
  Only used for SDI = YES.

An analysis was performed in Abaqus/Standard for every variation named above and then rated with regards to their influence on the solution. Figure 3.15 provides an overview of all variants, that were simulated with the DPM. Based on the parameter settings for the solver, the variants were divided into groups consisting of the different surface interaction models (SIMs) that were implemented, e.g. variant 01 to 05 in group 1. The results will be presented in Chapter 4.2.
For group 1 to 4 the α parameter was left unchanged at default values. In this case the value is determined by the time integrator as $\alpha = -0.41421$ for HHT-MD, and $\alpha = -0.05$ for HHT-TF, as it was also listed in Table 2.2.

* default value

Parameters							
Group	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Contact def.	GC	GC	GC	CP	GC	GC	GC
Time integr.	HHT-MD	HHT-MD	HHT-TF	HHT-MD	HHT-MD	HHT-MD	HHT-MD
Convert SDI	yes	no	no	no	no	no	no
α – HHT	-0.41421*	-0.41421*	-0.05*	-0.41421*	-0.2	0	-0.333

SIM	**Variant number**						
Augm. Lagr.	01	06	11	16	21	26	31
Penalty nonlin.	02	07	12	17	22	27	32
Penalty linear	03	08	13	18	23	28	33
Direct linear	04	09	14	19	24	29	34
Direct hard	05	10	15	20	25	30	35

Figure 3.15.: Overview of the simulated variants of the Disk Pad Model (DPM)

3.2.2. Axle-corner Model - ACM

The parameter study on the DPM suggests, that the following combination of settings leads to the best results (for details see Chapter 4.2.4):

- Contact definition: GENERAL CONTACT
- Surface interaction: Augmented Lagrange, penalty method or direct linear
- Solver setting: MODERATE DISSIPATION (HHT-MD)
- Convert SDI: YES or NO.

The simulated variants of the ACM were based in these findings. Thus simulations for all three test rig configurations (see Chapter 3.3) were performed with the above settings.

While running the simulations, it was discovered, that the ACM showed very poor convergence, which lead to very small time increment sizes and as a consequence to unreasonably long computing times. Predictions suggested more than one month per simulation, therefore modifications to the model were deemed necessary.

After several attempts to improve the convergence it was found that the numerical parameters of the HHT time integrator set by HHT-TF (instead of HHT-MD) resulted in greatly improved convergence behaviour. HHT-TF means less numerical damping than with the algorithm HHT-MD. Details about the HHT integrator and its parameters were presented in Chapter 2.4.1.2.

The time integrator was then set for all further simulations to the following settings:

- Impact: no
- Time incrementation: aggressive
- Time integrator: HHT-TF

The results for the three configurations with the contact models augmented Lagrange, linear penalty method, nonlinear penalty method and direct method respectively will be presented in Chapter 4.3. In Figure 3.16 and 3.17 every variant of the ACM that was simulated in Abaqus/Explicit and Abaqus/Standard is listed with their respective specifications. The position "Friction ID" refers to the friction model parameters according to Figure 3.13. The variants for Abaqus/Standard were once again divided into groups, as it was done for the DPM.
The setting "Convert SDI" was assumed to have very little or no impact on the explicit scheme, since it modifies the convergence behaviour of iterations, which an explicit scheme does not perform. However, it was still included to validate this assumption.
Furthermore, the number of CPUs used to run the simulations is included as well. Its relevance is discussed in Chapter 4.5.1. 4 CPUs were used for all simulations on the DPM.

Parameters				
Convert SDI	yes	no	yes	no
Friction ID	1	1	2	2
Brake pressure [bar]	32	32	8	8
Velocity [mm/s]	33.333	33.333	33.333	33.333
CPUs	4	4	4	4

SIM	Variant number			
Penalty linear	01	02	03	04

Figure 3.16.: Overview of the simulated variants of the Axle-corner Model (ACM) in Abaqus/Explicit

Parameters						
Group	**1**	**2**	**3**	**4**	**5**	**6**
Time integrator	HHT-TF	HHT-TF	HHT-TF	HHT-TF	HHT-TF	HHT-TF
Convert SDI	yes	yes	yes	no	no	no
Friction ID	1	2	2	1	2	2
Brake pressure [bar]	32	8	8	32	8	8
Velocity [mm/s]	33.333	33.333	88.888	33.333	33.333	88.888
CPUs	16	16	16	4	4	4

SIM	Variant number					
Augm. Lagrange	01	05	09	13	17	21
Penalty nonlinear	02	06	10	14	18	22
Penalty linear	03	07	11	15	19	23
Direct linear	04	08	12	16	20	24

Figure 3.17.: Overview of the simulated variants of the Axle-corner Model (ACM) in Abaqus/Standard

3.3. Test rig experiments

The test rig experiments conducted in [20] provide the basis for comparison of the simulation results of the ACM in Chapter 4. Detailed information about the test rig setup is provided in [20] and [21].
The investigation was conducted as follows: The parameters brake pressure and velocity were varied in form of test matrix screening measurements. The major creep groan frequencies that were found were then visualized in form of a "creep groan map", depicted in Figure 3.18. If creep groan was detected for a certain operating point, a coloured dot was added to the map. The colour indicates the major frequency of the creep groan that was measured. As one can see, two dominant frequencies were found at about 22 and 87 Hz.

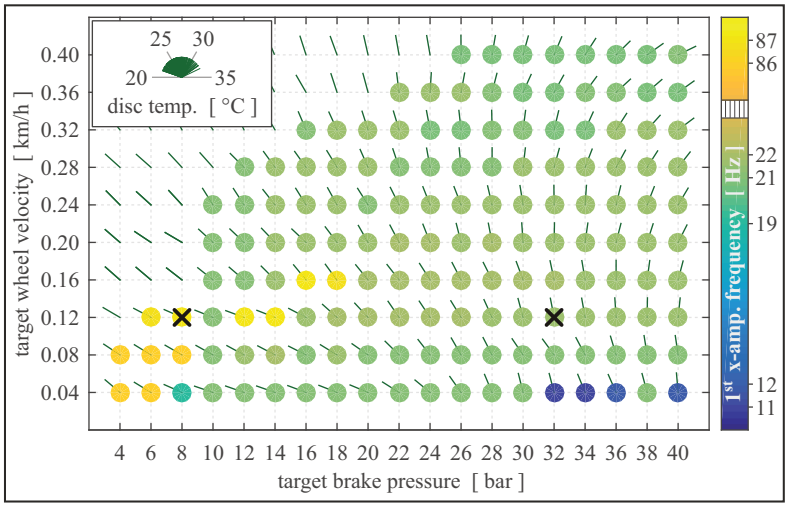

Figure 3.18.: Map of major creep groan frequencies gained from test rig experiments, adapted from [20]

The two operating points marked in Figure 3.18 were chosen as representatives of either frequency and are, [20]:

- Low-frequency creep groan:
  21.7 Hz at 32 bar and 0.12 km/h
- High-frequency creep groan:
  87.1 Hz at 8 bar and 0.12 km/h

The ACM was developed with the goal to obtain numerical results similar to the test rig. One additional operating point was added at which no creep groan was detected, with the parameters, [20]:

- No creep groan:
  At 8 bar and 0.32 km/h

The results were obtained using triaxial accelerometers at the caliper, pads and carrier, as depicted in Figure 3.19.

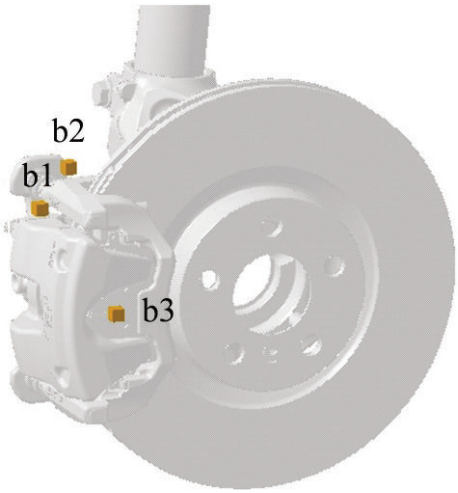

Figure 3.19.: Accelerometer positions on the test rig, adapted from [21]

A visualisation of the results can be derived, using the sensor data in combination with a brake geometry model, [21]. In Figure 3.20, one full repetition cycle is depicted for a low- and high-frequency creep groan respectively. The green path represents the trajectories of the sensors during the cycle, the yellow dots their initial positions. Note that the operating points that led to Figure 3.20 are not the same as the ones presented above, they are however similar.

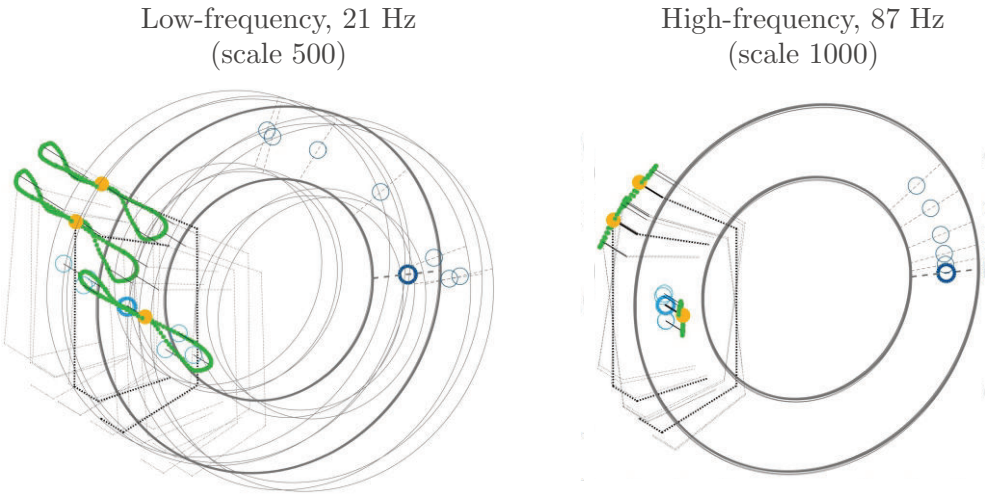

Figure 3.20.: Vibration cycles for low- and high-frequency creep groan, adapted from [21]

The tangential velocities of brake disk and pads at the theoretical friction radius, as well as their relative velocity to one another, have been derived in [21]. The respective graphs are depicted in Figure 3.21 and 3.22 for the low- and high-frequency creep groan.

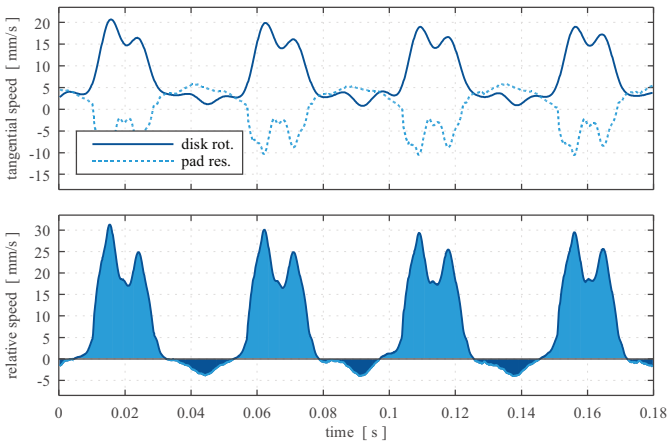

Figure 3.21.: Tangential and relative speed in vehicle z-direction for low-frequency creep groan, adapted from [21]

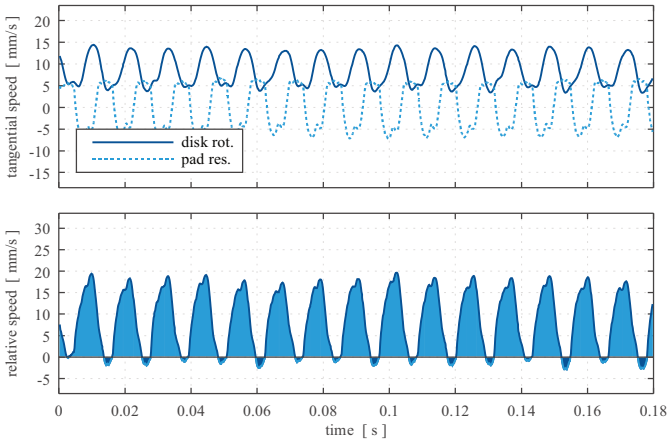

Figure 3.22.: Tangential and relative speed in vehicle z-direction for high-frequency creep groan, adapted from [21]

Similar plots were derived from the simulation results on the ACM and their validation will be based on the above figures.

4. Results

In this chapter the results of the parameter study of both models will be presented as they were explained in Chapter 3.2. This includes the Disk Pad Model DPM and the Axle-corner Model ACM in both Abaqus/Explicit and Abaqus/Standard respectively. All variants that were simulated are listed in Figure 3.15, 3.16 and 3.17.

For all further reference to the individual variants, the following denomination shall be used, e.g.:
Variant number 18 of the DPM in Abaqus/Standard will be called "DPM_Impl_18".
Variant number 02 of the ACM in Abaqus/Explicit will be called "ACM_Expl_02".

The variants will be compared with regards to their computing times and quality of the results, specifically how well they produce stick-slip. All simulations were performed on a simulation computer of the Institute of Automotive Engineering at the Technical University Graz with the following specifications:

- OS: Windows Server 2012 R2 Standard 64-Bit
- CPU: Intel(R) Xeon(R) CPU E5-2620 0 @ 2.00 GHz
- Cores: 12 (physical), 24 (logical)
- RAM: 96 GB DDR3
- GPU: NVIDIA GeForce GTX 980 4095 MB

4.1. Determination of the stick-slip frequency

For every simulation, a large amount of output data is generated as it was listed in Chapter 3.1.1. Generally, the stick-slip motion and thus the occurrence of creep groan starts at around 0.8 s into the simulation for the DPM. The low-frequency creep groan of the ACM starts at about 0.7 s, or at 0.5 s for the high-frequency one. This can be observed in the post-processor or by analysing the output data. To avoid influences of the transient at the beginning of the stick-slip phase, later stages of the simulation are of particular interest, namely the period from 1.7 to 1.9 seconds was used for the determination of creep groan in most cases.
The phenomenon can be best observed by calculating the relative velocity between the brake disk and the brake pads at the theoretical friction radius using output parameters for the velocities of nodes on the disk and pads. In Figure 4.1 a typical plot gained from

the Disk Pad Model (DPM) with this method is depicted. The algorithm performing this task was provided by the supervisors, as used in [20].

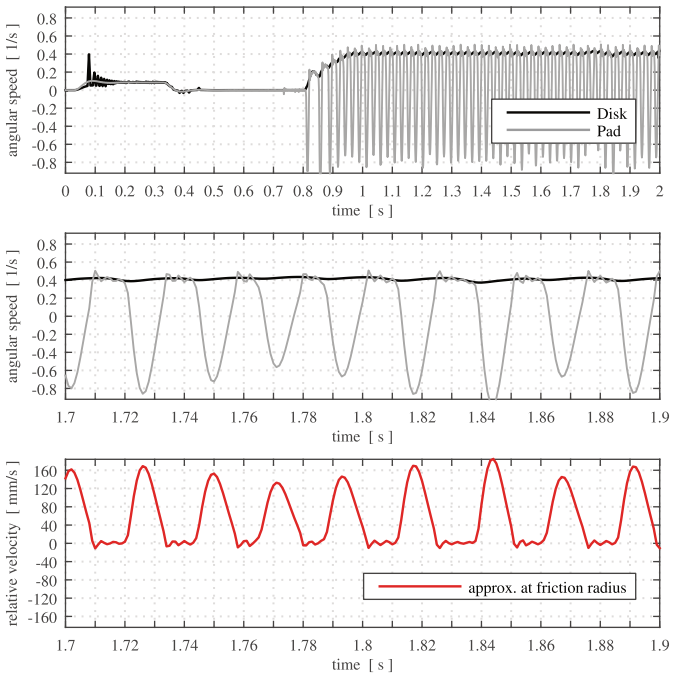

Figure 4.1.: Angular speed and relative velocity of the brake disk and pads of DPM_Impl_06

The top subfigure shows the time curve of the angular speed for both the brake disk and the brake pad throughout the entire simulation from $t = 0$ s to $t = 2$ s. The first 0.5 seconds show the effect of the application of the torque to the model, according to Figure 3.4. It is applied at $t = 0.03$ s, linearly increased until $t = 0.33$ s, and then held constant. The angular speed plot confirms this, by showing positive values between exactly those points in time. The elasticities in the model allow for deformations and thus an angular speed of the components due to the increasing torque that is applied. At $t = 0.33$ s, from where the torque is held constant, the angular speed goes back to zero and the model is quasi in a state of static equilibrium, until the beginning of the stick-slip phase due to the subsequently reduced brake pressure (see also Figure 3.4). From here, at about $t = 0.8$ s, the pads slip backwards in opposite direction of the disk, thus showing a momentary negative angular speed, while the disk begins to rotate at an almost constant angular speed. Small deviations occur due to the effect of the pads on the disk, as one can see in the middle subfigure in Figure 4.1. The pads then start to stick again to the disk until they have the same angular speed and tension can be built up until the next transition to slipping. This process was already explained in Chapter 2.1.1 and is now validated by the results of the simulation.
The phenomenon can be best observed by calculating the relative velocity at the

approximated friction radius between the disk and pad as it is depicted in the bottom subfigure of Figure 4.1. The stick-slip frequency can then be determined by counting the number of these events happening between two points in time, as it is shown in Figure 4.2. If the relative velocity is approximately zero, as it is the case after every peak in Figure 4.2, the movement of the brake disk and pads is identical, which indicates static friction, or sticking. During the peaks, relative motion between the disk and pads occurs, thus indicating kinetic friction or slipping.

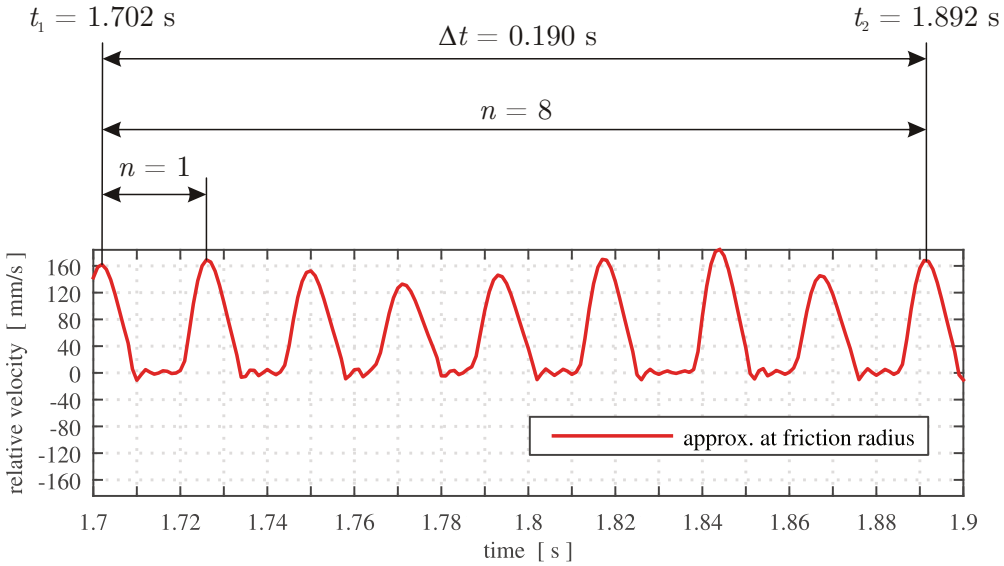

Figure 4.2.: Determination of the stick-slip frequency of DPM_Impl_06

Information gained from the graph:

- Time of the first peak in the interval: $t_1 = 1.702$ s
- Time of the last peak in the interval: $t_2 = 1.892$ s
- Number of stick-slip transitions between t_1 and t_2: $n = 8$

Now the stick-slip frequency can be calculated as:

$$f_{\mathrm{SS}} = \frac{n}{t_2 - t_1} = \frac{8}{1.892 - 1.702} = 42.11\,\mathrm{Hz}\,. \tag{4.1}$$

This frequency does not represent the frequency of the actual audible creep groan. The resulting audible noise consists of a wide spectrum of frequencies with distinct peaks at the aforementioned stick-slip frequency and its super-harmonics, as test rig experiments in [21], [20] revealed. The frequencies gained with this method represent thus the fundamental creep groan frequencies.

This method was used to determine the stick-slip frequencies of every variant of the Disk Pad Model DPM and the Axle-corner Model ACM.

4.2. Results for the Disk Pad Model DPM

4.2.1. Creep groan frequencies

4.2.1.1. DPM in Abaqus/Explicit

Initially the already existing results for the DPM in Abaqus/Explicit, shown in Figure 4.3, were meant to be used for comparison of the explicit and implicit solution.

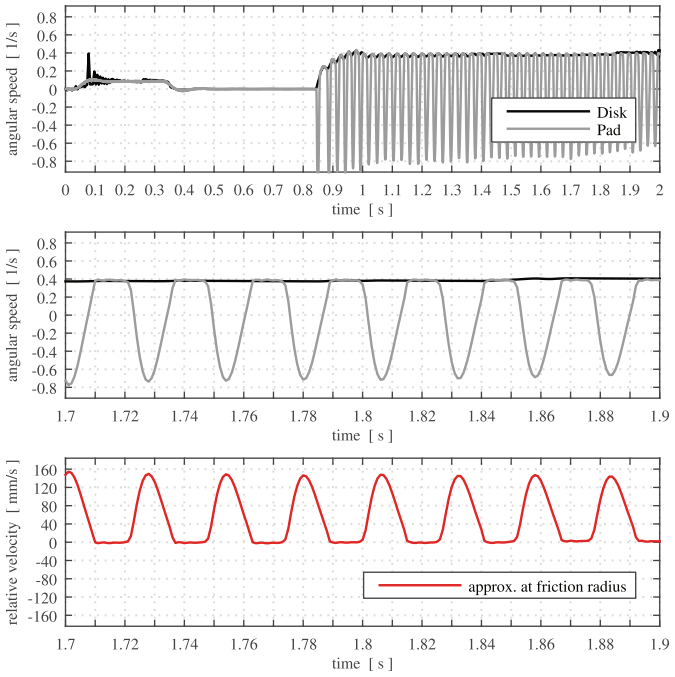

Figure 4.3.: Angular speed and relative velocity of the brake disk and pads of DPM_Expl_01

However, changes according to Chapter 3.2.1 were made, due to an initial noise in the signal for certain outputs. This noise can be observed e.g. by using the output data for the velocity in z-direction (DOF 3) of node 967 on the virtual brake carrier (not shown). This node represents the position of the acceleration sensor on the test rig and is depicted in Figure 4.4. Velocity was chosen instead of acceleration, since the phenomenon in question is better visible. Figure 4.5 shows the effect of the changes to the model for all 3 variants of the DPM in Abaqus/Explicit.

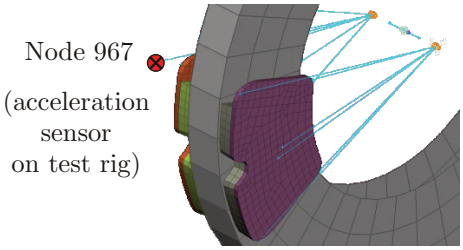

Figure 4.4.: Position of the virtual acceleration sensor (node 967) on the DPM

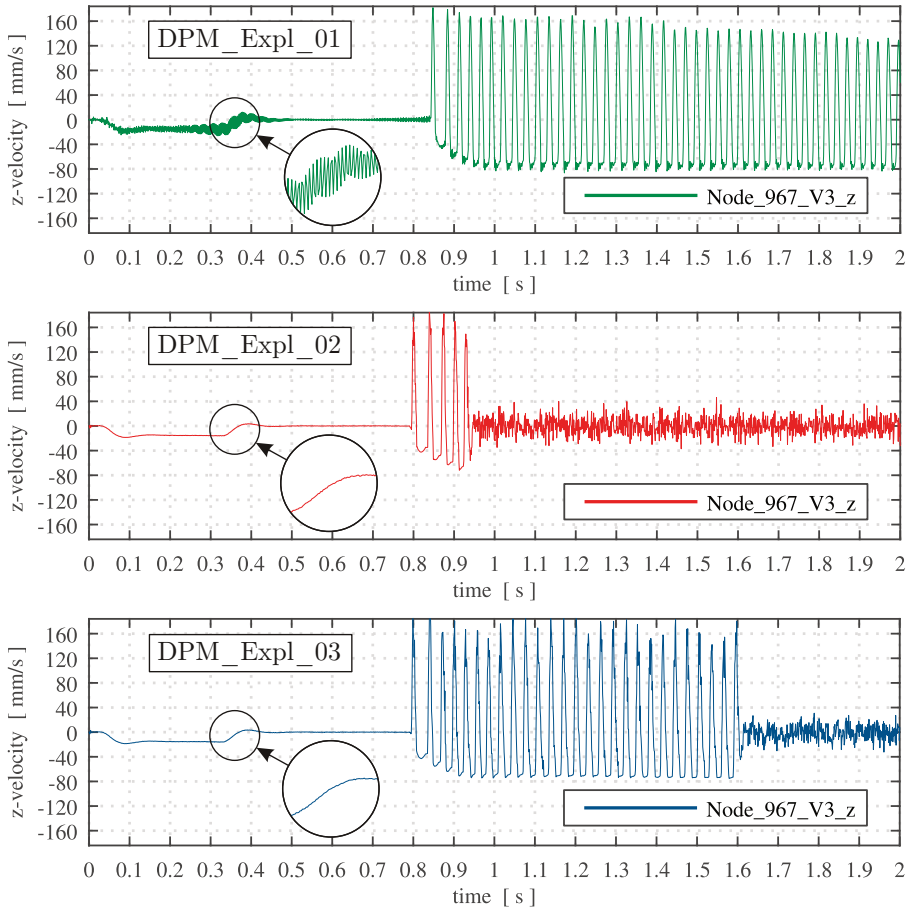

Figure 4.5.: Velocity in z-direction of the virtual acceleration sensor (node 967) of DPM_Expl_01, DPM_Expl_02 and DPM_Expl_03

Especially the first 0.5 seconds reveal the noise in the signal of DPM_Expl_01. As one can see, the noise was successfully removed in variant 02 and 03, but at the cost of a collapsing stick-slip effect. The delay of the collapse of the creep groan due to the updated amplitude function for the brake pressure (see Figure 3.14) can be observed as well.

The graphs for the angular speed of disk and pad according to Chapter 4.1 contain information similar to Figure 4.5. The noise however is not visible as one can see in Figure 4.6.

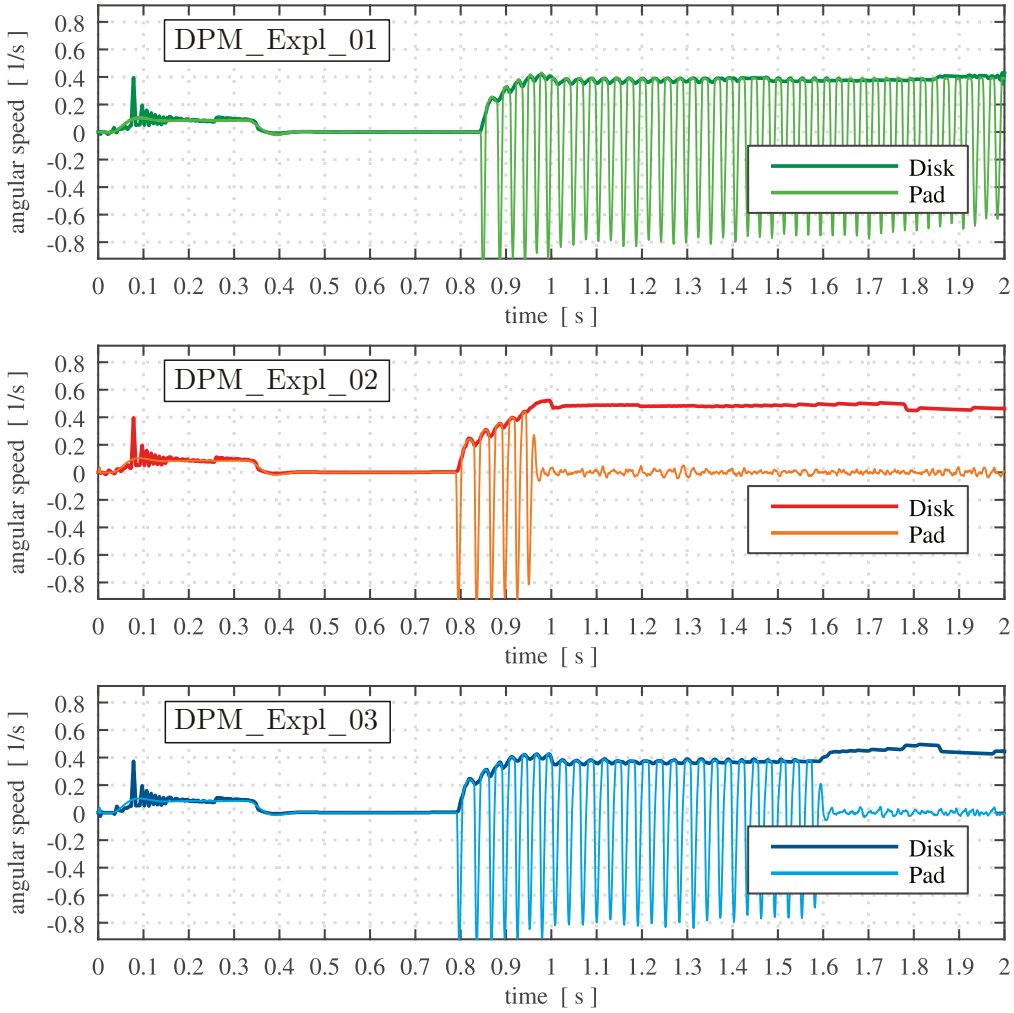

Figure 4.6.: Angular speed of the brake disk and pads of DPM_Expl_01, DPM_Expl_02 and DPM_Expl_03

Since the main interest is the creep groan towards the end of the simulation time, DPM_Expl_02 and DPM_Expl_03 are unsuitable. Thus, DPM_Expl_01 was used for means of comparison to the implicit model.

The following results were obtained for DPM_Expl_01:

- Computing time: 9.8 h
- Creep groan frequency: 38.46 Hz

4.2.1.2. DPM in Abaqus/Standard

Applying the method of determining the frequencies to every variant of the DPM in Abaqus/Standard according to Figure 3.15, the stick-slip frequencies listed in Figure 4.7 were found.

Group	1	2	3	4	5	6	7
Variant number	01 - 05	06 - 10	11 - 15	16 - 20	21 - 25	26 - 30	31 - 35

SIM	**Stick-slip frequency [Hz]**						
Augm. Lagrange	40.70	42.11	-	35.71	41.67	40.23	42.42
Penalty nonlinear	41.67	41.18	-	36.14	41.67	40.00	42.11
Penalty linear	42.25	42.68	-	37.50	41.18	40.00	40.00
Direct linear	41.42	41.67	-	35.71	41.18	40.00	40.00
Direct hard	41.18	42.42	-	37.04	41.67	40.00	42.94

Figure 4.7.: Stick-slip frequencies of the DPM in Abaqus/Standard

As one can see in Figure 4.7, for group 3 (variant 11 - 15) no creep groan frequencies could be determined, since no stable stick-slip motion could be produced with the parameter "Time integrator = HHT-TF". The equivalent plot to Figure 4.1 for DPM_Impl_11 is depicted as an example in Figure 4.8. The first half is very similar to the other variants, but the initial stick-slip behaviour starts to collapse at around $t = 1$ s and transitions to a state of constant kinetic friction. The pads remain in place, while the disk is rotating at a constant angular speed. Without stick-slip transitions, no creep groan occurs and a frequency cannot be determined.

The point in time at which the stick-slip effect collapses differs between the variants from $t = 0.92$ s for DPM_Impl_13, to $t = 1.33$ s for DPM_Impl_14. Thus, the variants of group 3 with "Time integrator = HHT-TF" were judged to be unsuitable for the simulation of creep groan on this model and were excluded from any further investigations. With all other variants, stable stick-slip reproductions relatable to creep groan could be produced as one can see in Figure 4.9 with the relative velocities for e.g. all variants with "Penalty linear" surface interaction. Small deviations between the stick-slip periods can be observed, but generally the results are similar for all variants.

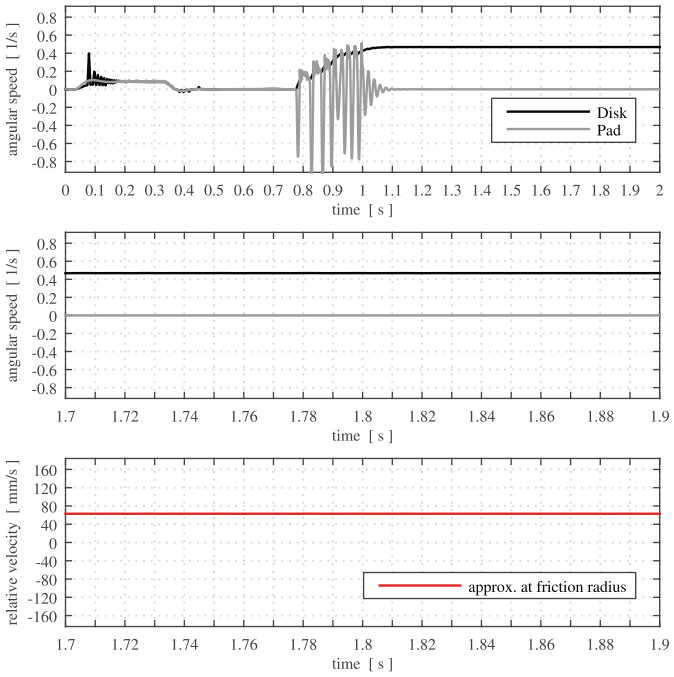

Figure 4.8.: Angular speed and relative velocity of DPM_Impl_11

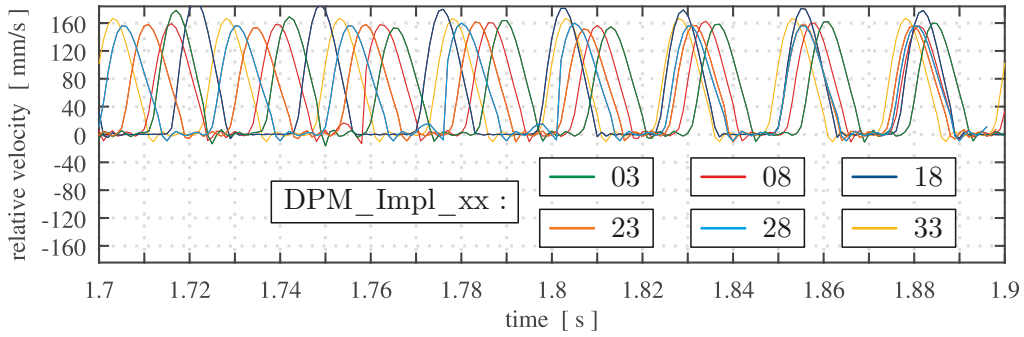

Figure 4.9.: Relative velocities of different variants with linear penalty contact

The stick-slip frequencies for all variants with a contact definition GENERAL CONTACT are very similar and within a range of 40 - 43 Hz. Defining the contact as CONTACT PAIR (group 4) leads to a reduction of the creep groan frequency of about 5 - 6 Hz compared to identical variants with GENERAL CONTACT across all surface interactions that were simulated.
Since the DPM is a solely academic model, we have no means of validating the frequencies via e.g. results of test rig experiments. It is thus not clear, if the results of CONTACT PAIR or GENERAL CONTACT are more realistic.

4.2.2. Computing times

As the previous chapter showed, stable stick-slip limit cycles could be produced with every variant of the DPM, with the exception of group 3. The corresponding computing times are listed in Figure 4.10. 4 CPUs were used in all cases.

Group	1	2	3	4	5	6	7
Variant number	01 - 05	06 - 10	11 - 15	16 - 20	21 - 25	26 - 30	31 - 35
SIM	**Computing time [h]**						
Augm. Lagrange	7.62	6.12	3.72	6.05	7.28	4.60	8.65
Penalty nonlinear	6.84	6.05	4.43	6.25	5.67	3.63	7.68
Penalty linear	5.97	6.47	3.67	5.75	5.78	4.67	5.23
Direct linear	6.67	6.98	5.25	4.43	6.00	6.57	6.33
Direct hard	6.43	6.03	6.07	5.75	6.65	5.50	6.83

Figure 4.10.: Computing times of the DPM in Abaqus/Standard

The computing times, in combination with the results for stick-slip frequencies in the previous chapter, can now be used to examine the effects of the different model settings and parameters on the performance of the simulation.
When comparing the computing times in the above table one can notice, that changes to the model can lead to increases or decreases in computing time depending on the surface interaction model (SIM) or solver setting. A more detailed investigation and comparison of the results is thus necessary and will be presented in the next chapter.

4.2.3. Comparison

The individual groups of variants were compared to one another in order to judge, which settings lead to the best and fastest results. Group 1 represents the most general settings of the model since all parameters are at default and shall be used as starting point for the comparison. As already mentioned, no means of validation for the stick-slip frequencies are available for this model. The DPM is an exclusively academic model that does not represent realistic parameters of disk brakes. Thus, the magnitudes of the frequencies are mostly irrelevant. A statement about whether or not a stick-slip limit cycle occurs is sufficient for this model. As already mentioned above, a stable cycle could be successfully produced with all variants of the DPM, with the exception of group 3.

Effects of "Convert SDI" (Group 1 vs. Group 2):
Small changes to the frequencies can be observed. For the SIM "Penalty nonlinear" the change leads to a decrease, and to an increase for all other SIMs.
The computing time was significantly reduced for "Augmented Lagrange" and "Penalty nonlinear" and slightly increased for "Penalty linear" and "Direct linear". For "Direct hard" a slight reduction can be observed as well. No significant differences in the characteristics of the stick-slip motion were found. It was thus concluded, that the setting "Convert SDI = no" leads to an overall improvement in performance of the simulation. Therefore it was kept for all further investigations on this model and the following groups of variants will be compared to group 2 instead. In Table 4.1 and 4.2 the changes in frequency and computing times relative to group 2 are listed.

Table 4.1.: Relative changes to the creep groan frequencies of the DPM in Abaqus/Standard

Groups	2 vs. 1	4 vs. 2	5 vs. 2	6 vs. 2	7 vs. 2
Relative change of the frequency [%]	+3.46	-15.18	-1.04	-4.45	-1.63
	-1.18	-12.22	+1.19	-2.86	+2.26
	+1.02	-12.14	-3.53	-6.29	-6.29
	+0.60	-14.29	-1.18	-4.00	-4.00
	+3.03	-12.70	-1.79	-5.71	+1.23

Table 4.2.: Relative changes to the computing times of the DPM in Abaqus/Standard

Groups	1 vs. 2	4 vs. 2	5 vs. 2	6 vs. 2	7 vs. 2
Relative change of the comp. time [%]	+24.51	-1.14	+18.95	-24.84	+41.34
	+13.22	+3.31	-6.28	-40.00	+26.94
	-7.73	-11.13	-10.66	-27.82	-19.17
	-4.44	-36.53	-14.04	-5.87	-9.31
	+6.63	-4.64	+10.28	-8.79	+13.27

Effects of time integrator (Group 3 vs. Group 2):
As it was already documented in Chapter 4.2.1, the change from "HHT-MD" to "HHT-TF" results in a collapse of the stick-slip for all variants of group 3. Thus, group 3 was declared unsuitable.

Effects of contact definition (Group 4 vs. Group 2):
The change from GENERAL CONTACT to CONTACT PAIR leads to a significant reduction of the frequencies across all SIMs. The computing time for "Direct linear"

was reduced drastically by 36.53 %, and for "Penalty linear" by 11.13 %. For the remaining SIMs the changes in computing time are very small and thus negligible.
As it was explained in Chapter 2.4.2.1, the main difference between GENERAL CONTACT and CONTACT PAIR in Abaqus/Standard is the contact discretization used. GENERAL CONTACT uses Surface-to-surface, and CONTACT PAIR uses Node-to-surface (see Chapter 2.3.4.1). Otherwise, the two options are mostly identical in Abaqus/Standard. It can thus be concluded, that a Node-to-surface discretization leads to lower stick-slip frequencies than Surface-to-surface. The overall improvement in computing time would suggest, that CONTACT PAIR is the better option for this model.
However, the two options are fundamentally different in Abaqus/Explicit, and thus a comparison between the explicit and implicit solution would not be reasonable for CONTACT PAIR. But since this comparison is topic of this thesis, the GENERAL CONTACT formulation was kept for further investigations. It is identical in both Abaqus/Explicit and Abaqus/Standard and allows for a more accurate comparison.
Furthermore, it is stated e.g. in [1] and [25], that the Surface-to-surface discretization is capable of modelling the contact condition a lot better than Node-to-surface at the cost of higher computing times. This was another reason to use GENERAL CONTACT, since the contact interaction is the most crucial part of these models.

Effects of numerical damping (Groups 5-7 vs. Group 2):
The amount of numerical damping of the algorithm can be controlled by the parameter α in the solver settings (see Chapter 2.4.1.2). The solver HHT-MD uses by default $\alpha = -0.41421$, which represents a significant amount of numerical damping.

Group 5: reduced numerical damping ($\alpha = -0.2$)
The stick-slip frequencies remained mostly unchanged. A slight reduction can be observed, with the exception of "Penalty nonlinear". Depending on the SIM, reduced numerical damping leads to significant increases or decreases in computing times.

Group 6: no numerical damping ($\alpha = 0$)
The absence of numerical damping leads to lower frequencies and drastic reductions of the computing times of up to 40%.

Group 7: increased (maximum) numerical damping ($\alpha = -0.333$)
Depending on the SIM, an increase of the numerical damping results in increases or decreases of both stick-slip frequencies and computing times.

As one can see, modifications of the numerical damping lead to rather unpredictable and partially contradictory results. For "Augmented Lagrange" and "Direct hard", both increases and reductions of α cause higher computing times. Reduced computing times are the result for "Penalty linear" and "Direct linear". Only for "Penalty nonlinear", an

increase in damping leads also to an increase in computing time, and a reduction of the damping to a decrease. It can thus be derived, that the presence of numerical damping in the algorithm leads to higher computing costs.
Abaqus writes information about a simulation, such as the solver parameters, into a message file that can be accessed. Upon investigation of these files it was discovered, that a manual change to the α parameter does not change the other parameters β and γ. Changes according to Eq. (2.69) were expected, but instead the default values of the time integrator, e.g. HHT-MD, were kept with only α being changed. Thus group 6 with $\alpha = 0$ still has the numerical dissipation introduced by the Newmark-β scheme, since γ was not set to $\frac{1}{2}$.
Based on these findings it was concluded, that modifications to the numerical damping lead to unpredictable results. Thus the parameters of the solver were left at default settings for all further simulations.

4.2.4. Conclusions of the parameter study

The motivation for the parameter study on the DPM was, to gain information about reasonable settings and parameters for the ACM. The study has shown, that slight modifications to the model can lead to severe changes of the results. Due to the greatly increased complexity of the ACM compared to the DPM, the findings of this study might not apply to the ACM in all cases. It does however provide a starting point for the simulations and helps to understand the effects of certain parameters to the model.
Based on the findings in this chapter, the following settings were chosen for simulations on the ACM:

- Contact definition: GENERAL CONTACT
- Numerical damping α: default
- Time integrator: HHT-MD

For the surface interaction model and the "Convert SDI" setting, a best option could not be found. The effects on the computing times are unclear and partly contradictory, depending on the SIM used. Based on the literature, either penalty contact, or augmented Lagrange contact are expected to be the most optimal for the problem at hand. Additionally, due to reasons related to contact status (see Chapter 4.5.2), the "Direct linear" contact was included as well. Since severe discontinuities, convergence issues and thus greatly increased computing times are expected for the ACM, both options for "Convert SDI" (YES or NO) were applied.

The following settings were simulated on the ACM:

- Surface interaction model (SIM):
  - Augmented Lagrange
  - Penalty nonlinear
  - Penalty linear
  - Direct linear
- Convert SDI:
  - YES
  - NO

This leads to a set of 8 model variants that were then applied to the 3 different operating points (brake pressure, velocity) presented in Chapter 3.3 resulting in a total of 24 variants for the ACM (see Figure 3.17).

4.3. Results of the Axle-corner Model ACM

As already mentioned in Chapter 3.3, test rig experiments were conducted in [21], [20] for this model. Creep groan could be produced with two different fundamental frequencies:

- 21.7 Hz with 32 bar and 0.12 km/h (low-frequency creep groan), and
- 87.1 Hz with 8 bar and 0.12 km/h (high-frequency creep groan).

These frequencies will be used as means of validation of the simulation results.

4.3.1. Stick-slip frequencies

4.3.1.1. ACM in Abaqus/Explicit

From previous investigations by my supervisor on this model, results of an explicit simulation were already available for both frequencies and were generated before any changes to the model were made, as it was explained in Chapter 3.1.2.2. This allows for an investigation of the effects of changes to the explicit model. The stick-slip frequencies gained from the original and modified ACM are listed in Figure 4.11.

Variant number	01	02	03	04	x1	x2	x3
SIM	**Stick-slip frequency [Hz]**						
Penalty linear	21.98	21.98	-	-	20.90	88.89	-

Figure 4.11.: Stick-slip frequencies of the original (x1, x2, x3) and modified (01, 02, 03, 04) ACM in Abaqus/Explicit

The variants 01 and 02 produce identical results since the only difference is the setting "Convert SDI" (YES or NO), which was expected to be irrelevant for simulations in Abaqus/Explicit. ACM_Expl_x1 is equivalent to ACM_Expl_01, and ACM_Expl_x2 to ACM_Expl_03 before and after the changes. ACM_Expl_x3 represents the operating point leading to constant kinetic friction, but was not simulated again. As one can see, results similar to the test rig were obtained for the low-frequency creep groan before (x1) and after (01, 02) the changes. The modification of the model resulted in a slightly lower frequency, that is now very close to the experimental result. In Figure 4.12 both versions of the model are compared, using the same content as for the DPM. The changes had little effect on the low-frequency creep groan.

For the high-frequency creep groan the changes to the model resulted in drastic changes of the results as depicted in Figure 4.13. With ACM_Expl_x2, a clean stick-slip effect and thus creep groan could be produced, whereas ACM_Expl_03 shows a different behaviour. This can be best observed by analysing the angular speed between 1.7 and 1.9 seconds in Figure 4.13. For ACM_Expl_03 no real stick phases occur, but instead both the brake disk and the pad show an alternating motion at different frequencies. The disks speed repeatedly reduces close to zero, where it gets very close to the pads speed, which constantly oscillates back and forth with positive/negative angular speed. Before the changes, the disk would rotate at an almost constant angular speed with an also alternating pad motion, but with a greater amplitude. The frequency of the pad oscillation of ACM_Expl_03 can be determined as 88.40 Hz, which matches the frequency obtained from ACM_Expl_x2 of 88.89 Hz and from the test rig of 87.1 Hz. However, the results of ACM_Expl_03 do not represent stick-slip and this variant has to be declared invalid. Further investigations and modifications to the model would be necessary in order to obtain results comparable to ACM_Expl_x2.

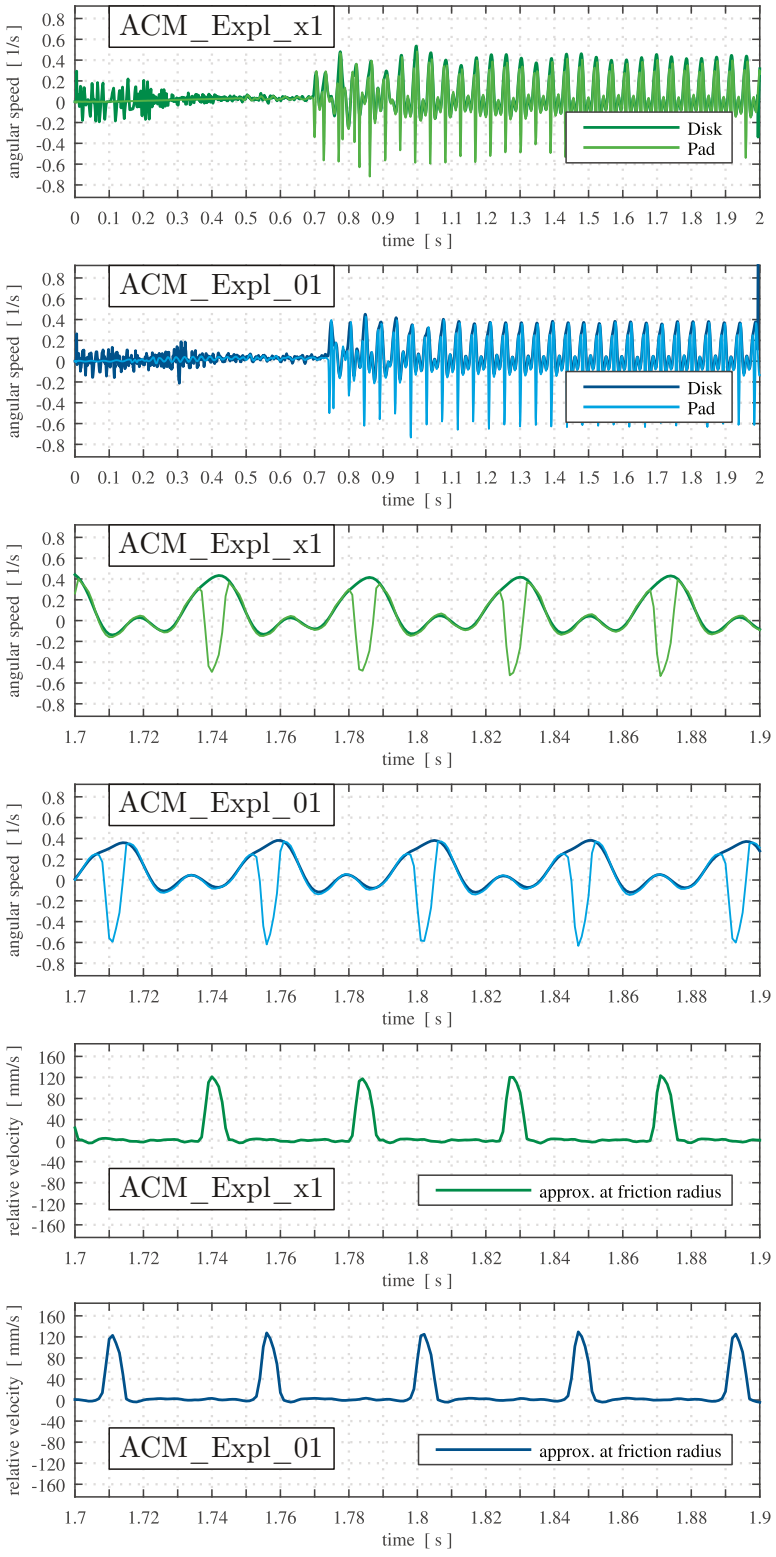

Figure 4.12.: Comparison of the ACM in Abaqus/Explicit before (x1) and after (01) the model changes (low-frequency creep groan)

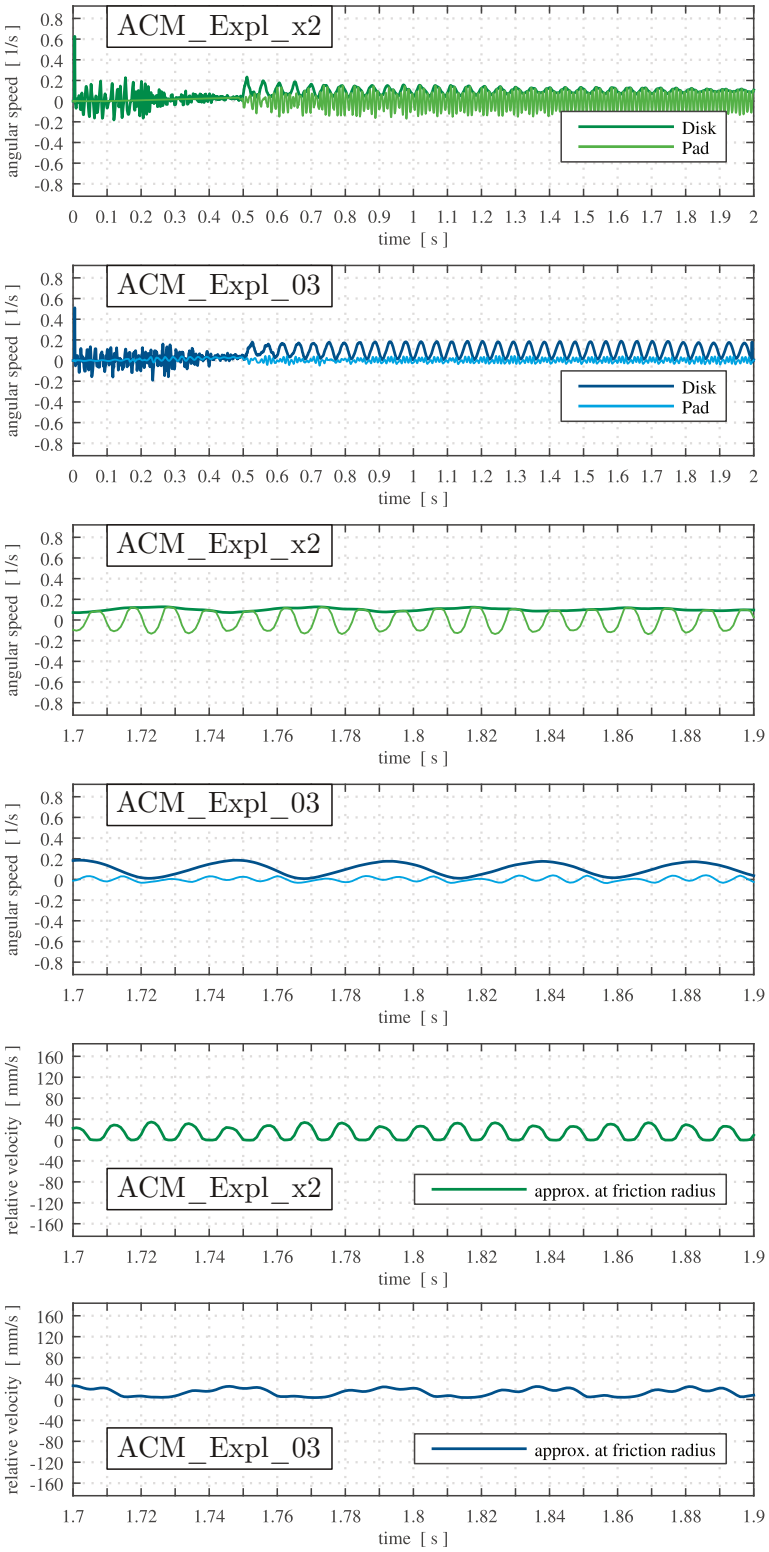

Figure 4.13.: Comparison of the ACM in Abaqus/Explicit before (x2) and after (03) the model changes (high-frequency creep groan)

4.3.1.2. ACM in Abaqus/Standard

For the variants of the ACM in Abaqus/Standard according to Figure 3.17, the following stick-slip frequencies, listed in Figure 4.14, were obtained.

Group	1	2	3	4	5	6
Variant number	01 - 04	05 - 08	09 - 12	13 - 16	17 - 20	21 - 24

SIM	**Stick-slip frequency [Hz]**					
Augm. Lagrange	17.78	87.50	-	19.35	-	-
Penalty nonlinear	20.00	87.50	-	20.00	-	-
Penalty linear	20.00	88.24	-	19.57	-	-
Direct linear	18.60	93.33	-	17.39	-	-

Figure 4.14.: Stick-slip frequencies of the ACM in Abaqus/Standard

As one can see in Figure 4.14, the variants of group 3, 5, and 6 did not produce creep groan. For group 3 and 6 this was expected, since the parameters were chosen to result in no creep groan and were only included for comparison. In Figure 4.15 the results of ACM_Impl_09 are depicted, which are representative of the other variants of group 3 and 6.

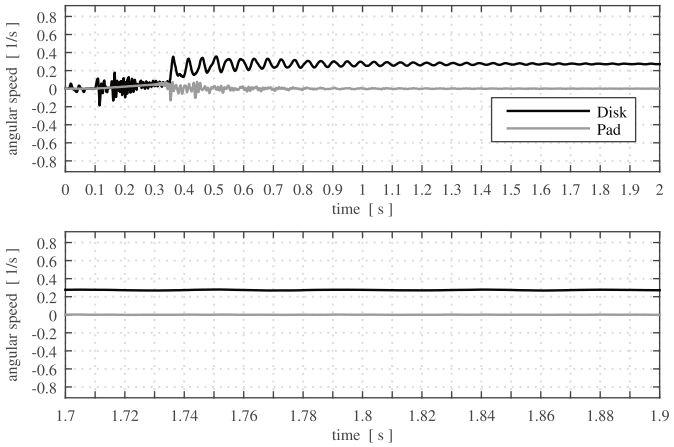

Figure 4.15.: Angular speed and relative velocity of ACM_Impl_09 (no creep groan)

The model transitions to a state, where the disk is rotating at a constant angular speed, while the pad stays in place. Constant kinetic friction between disk and pad is the result.

High-frequency creep groan (Group 2 and 5):

The variants of group 2 were able to reliably produce the high-frequency creep groan with similar results across all surface interaction models. Only for the SIM "Direct linear" a significantly higher frequency was obtained. The relative velocity graphs are depicted in Figure 4.16 for all four variants.
Group 5 is identical to group 2, but the setting "Convert SDI" was changed from YES to NO, which leads to results similar to ACM_Expl_03 in Figure 4.13. In Figure 4.17 a comparison of the variants ACM_Impl_07 and ACM_Impl_19 is provided, which are representative of their respective groups. As one can see, with ACM_Impl_07 a stable stick-slip effect was obtained, whereas ACM_Impl_19 produced rather chaotic results. Behaviour similar to stick-slip can be observed, but the characteristics differ vastly from the desired outcome and thus has to be declared invalid.

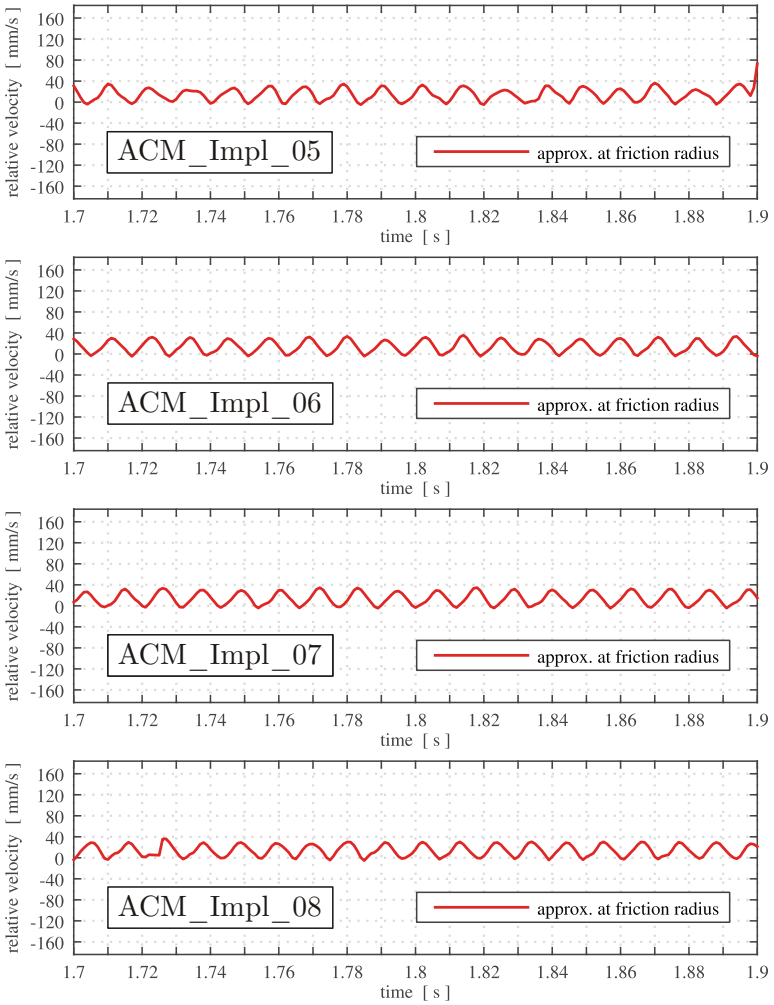

Figure 4.16.: Relative velocity of ACM_Impl_05, ACM_Impl_06, ACM_Impl_07 and ACM_Impl_08 (high-frequency creep groan)

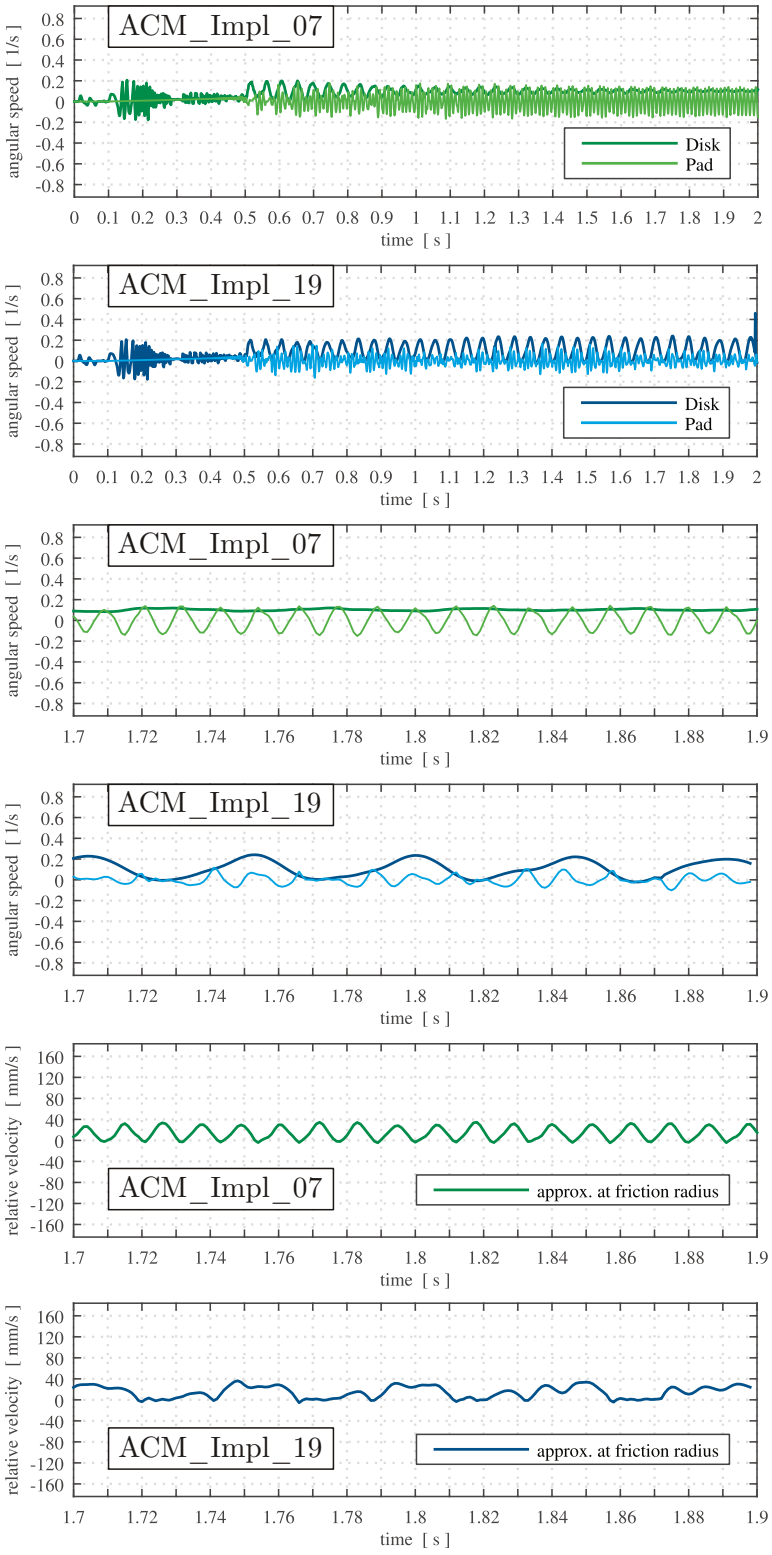

Figure 4.17.: Angular speed and relative velocity of ACM_Impl_07 and ACM_Impl_19 (high-frequency creep groan)

Low-frequency creep groan (Group 1 and 4):

One can see in Figure 4.12, that the explicit simulation produces for the chosen parameters very clean results with one distinct peak of relative velocity per stick-slip event for the low-frequency creep groan. Investigations of the results of the implicit simulations revealed an overshooting and transient after every stick-slip transition, that did not occur to a noticeable degree with the explicit simulation. Figure 4.18 provides a comparison of two representative variants, ACM_Expl_01 and ACM_Impl_03. The plot of the full time interval is omitted, since the phenomenon can be best observed with the more detailed graphs from 1.7 to 1.9 seconds.

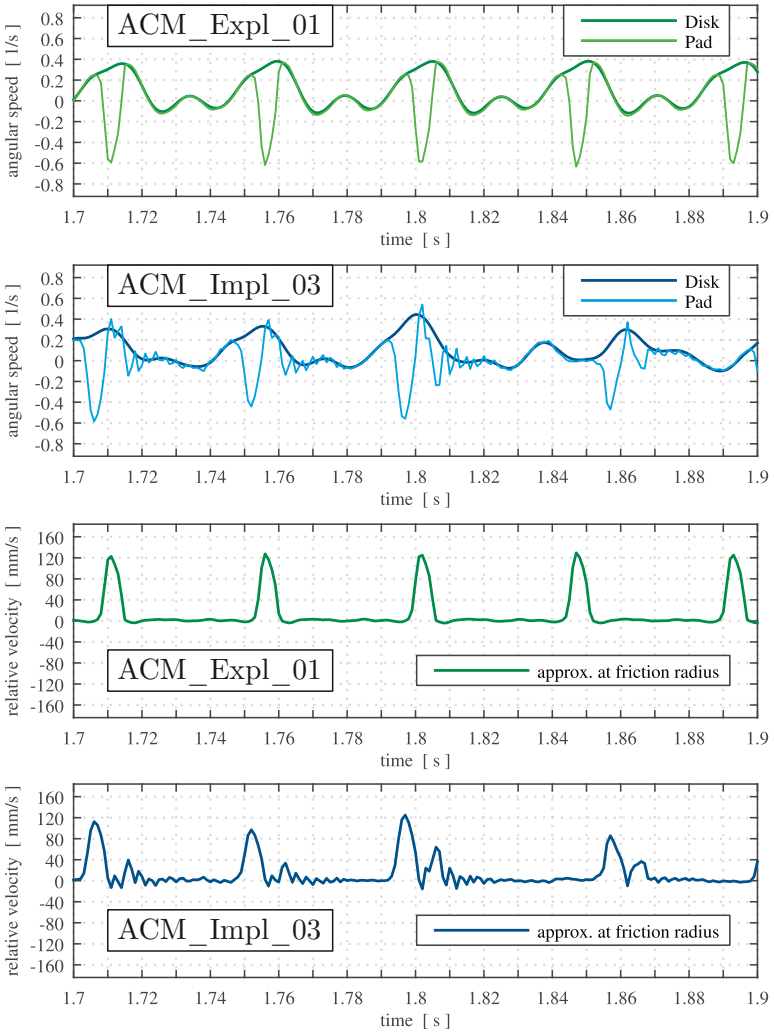

Figure 4.18.: Angular speed and relative velocity of ACM_Expl_01 and ACM_Impl_03 (low-frequency creep groan)

Since the implicit solution showed less consistent results, the time interval that was used to determine the stick-slip frequency was extended from 1.7 to 1.9 seconds to 1.4 to 1.9 seconds. In Figure 4.19 the relative velocities in the extended time interval of the variants of group 1 are depicted.
To group 4, the same extension of the interval was applied. When comparing these four variants one can see, that variant 01 with an augmented Lagrange contact formulation shows occasionally larger spikes in relative velocity, while the peaks are roughly equivalent in height for the other variants. The characteristics of the peaks however differ vastly throughout the observed interval which can be an indicator, that the system has not reached a steady state yet. Further simulations over a longer period of time than 2 seconds would be necessary, but were not performed for this thesis.

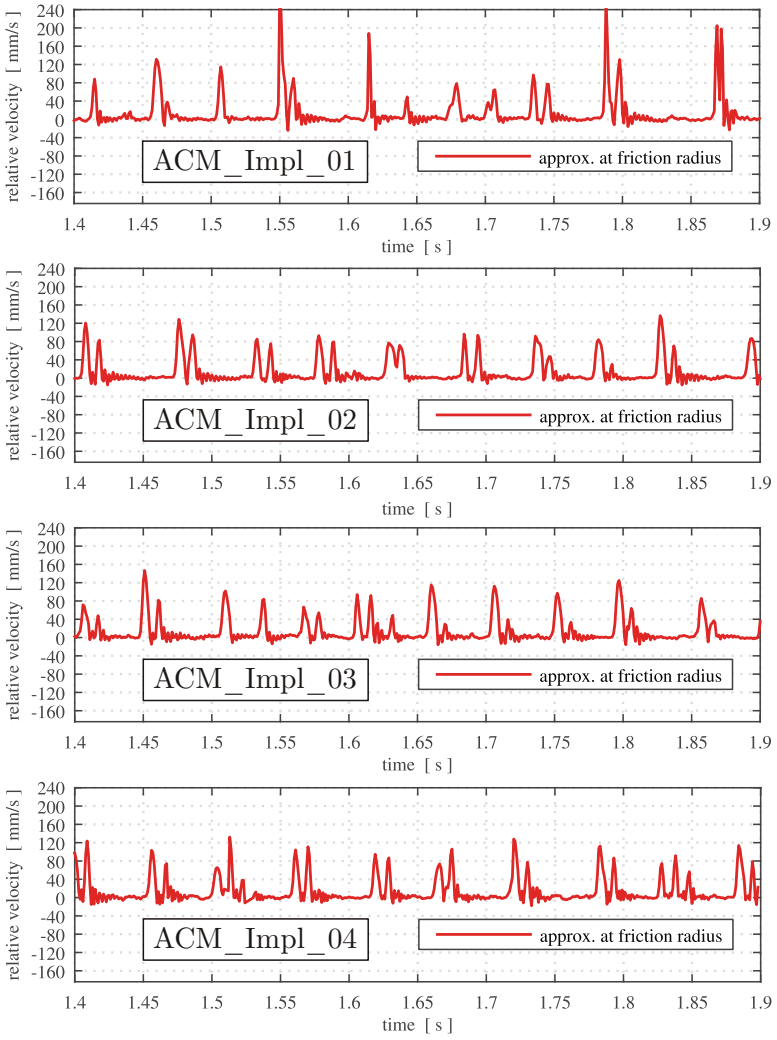

Figure 4.19.: Relative velocity of ACM_Impl_01, ACM_Impl_02, ACM_Impl_03 and ACM_Impl_04 (low-frequency creep groan)

Figure 4.20 shows the four variants of group 4. When comparing the results of group 1 and group 4, the setting "Convert SDI = NO" does not lead to the disappearance of the stick-slip effect as it was the case for the high-frequency creep groan. Instead it seems to lead to more regular peaks and thus a more stable stick-slip limit cycle. For ACM_Impl_13 this is not the case, since here the creep groan comes to a halt at about 1.75 seconds. As one can see, the penalty contact (variant 14 and 15) delivers the clearest stick-slip repetitions and therefore results most similar to a steady-state creep groan behaviour.

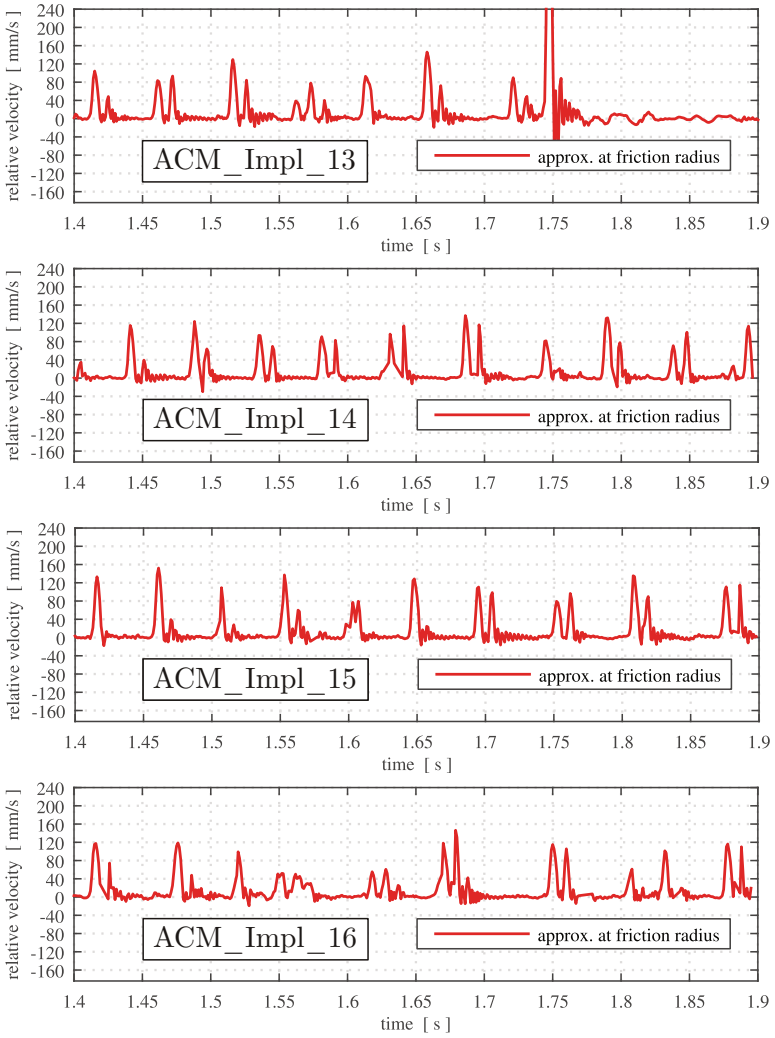

Figure 4.20.: Relative velocity of ACM_Impl_13, ACM_Impl_14, ACM_Impl_15 and ACM_Impl_16 (low-frequency creep groan)

4.3.2. Computing times

4.3.2.1. ACM in Abaqus/Explicit

The computing times of the ACM variants in Abaqus/Explicit are listed in Figure 4.21. Compared to the DPM, almost no increases occurred, despite the greatly increased complexity of the ACM. For all variants the same time increment of 3.61221 E-07 s and 4 CPUs were used.

Variant number	01	02	03	04	x1	x2	x3
SIM	**Computing time [h]**						
Penalty linear	10.85	9.97	10.47	9.48	11.02	10.63	10.30

Figure 4.21.: Computing times of the ACM in Abaqus/Explicit

The setting "Convert SDI" showed to have very little effect on the explicit simulation, as it was expected. Changes from YES to NO resulted in slight reductions in computing times of roughly 10 %. The computing times are very similar for the low-frequency creep groan (variant 01, 02, and x1) and the high-frequency creep groan (variant 03, 04, and x2), as well as for variant x3 with no creep groan. All variants were run on 4 CPUs.

4.3.2.2. ACM in Abaqus/Standard

The computing times for all variants of the ACM in Abaqus/Standard are listed in Figure 4.22 below.

Group	1	2	3	4	5	6
Variant number	01 - 04	05 - 08	09 - 12	13 - 16	17 - 20	21 - 24
SIM	**Computing time [h]**					
Augm. Lagrange	19.58	25.43	5.85	13.57	15.70	6.78
Penalty nonlinear	28.90	54.67	4.92	53.77	65.12	6.43
Penalty linear	32.57	35.10	5.40	14.50	10.20	5.80
Direct linear	15.10	64.18	4.97	61.35	100.57	4.77

Figure 4.22.: Computing times of the ACM in Abaqus/Standard

As briefly mentioned in Chapter 3.2.2, a different number of CPUs was used for the groups 1-3 (16 CPUs) and 4-6 (4 CPUs) respectively. Based on the findings in Chapter 4.5.1 regarding this topic, according to Table 4.6 an uncertainty of about ± 5% has to be considered for the computing times in Figure 4.22.
When comparing the computing times of the ACM in the above table to the ones of the DPM in Figure 4.10, a drastic increase can be observed. For the explicit simulations this was not the case. It can thus be concluded, that the complexity of the model barely influences the explicit scheme with a fixed time increment size and no iterations. For the iterative implicit scheme on the other hand, the complexity affects the convergence behaviour severely. One can also notice that group 3 and 6 show very fast computing times. These groups do not produce creep groan, but constant kinetic friction instead (see Figure 4.15). This is an indicator of how severely the stick-slip transitions effect the implicit numerical solution. Depending on the model configuration, increases in computing time by a factor of up to 10 or more can be the result. For the explicit simulation this was not the case.

4.4. Validation with post-processor

After a simulation was completed, the results were analysed in the post-processor META. Special attention was paid to a correct representation of the stick-slip effect, as well as a physically accurate deformation of the components. The former can be investigated on both models, DPM and ACM, the latter only on the ACM, since auxiliary components are missing in the DPM. The expected behaviour was derived form test rig experiments in [21], thus the ACM will be used for validation. The general stick-slip behaviour of the DPM is very similar to the ACM, thus the following applies to it as well. The displacement of the ACM at two points in time, t_0 and t_1 according to Figure 4.23, before beginning of the stick-slip phase is depicted in Figure 4.24. t_1 was chosen exactly before the first global slip, thus representing the state of maximum pretensioning in the model. A scale factor of 20 was applied.

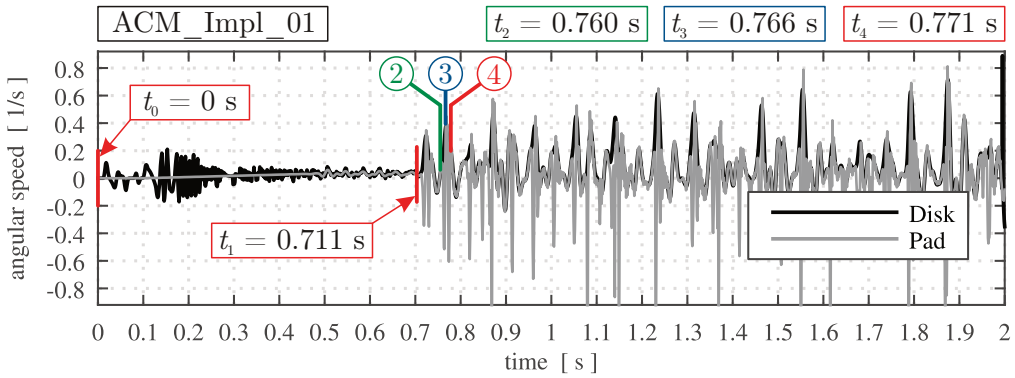

Figure 4.23.: Points in time for visualisation of the results in META

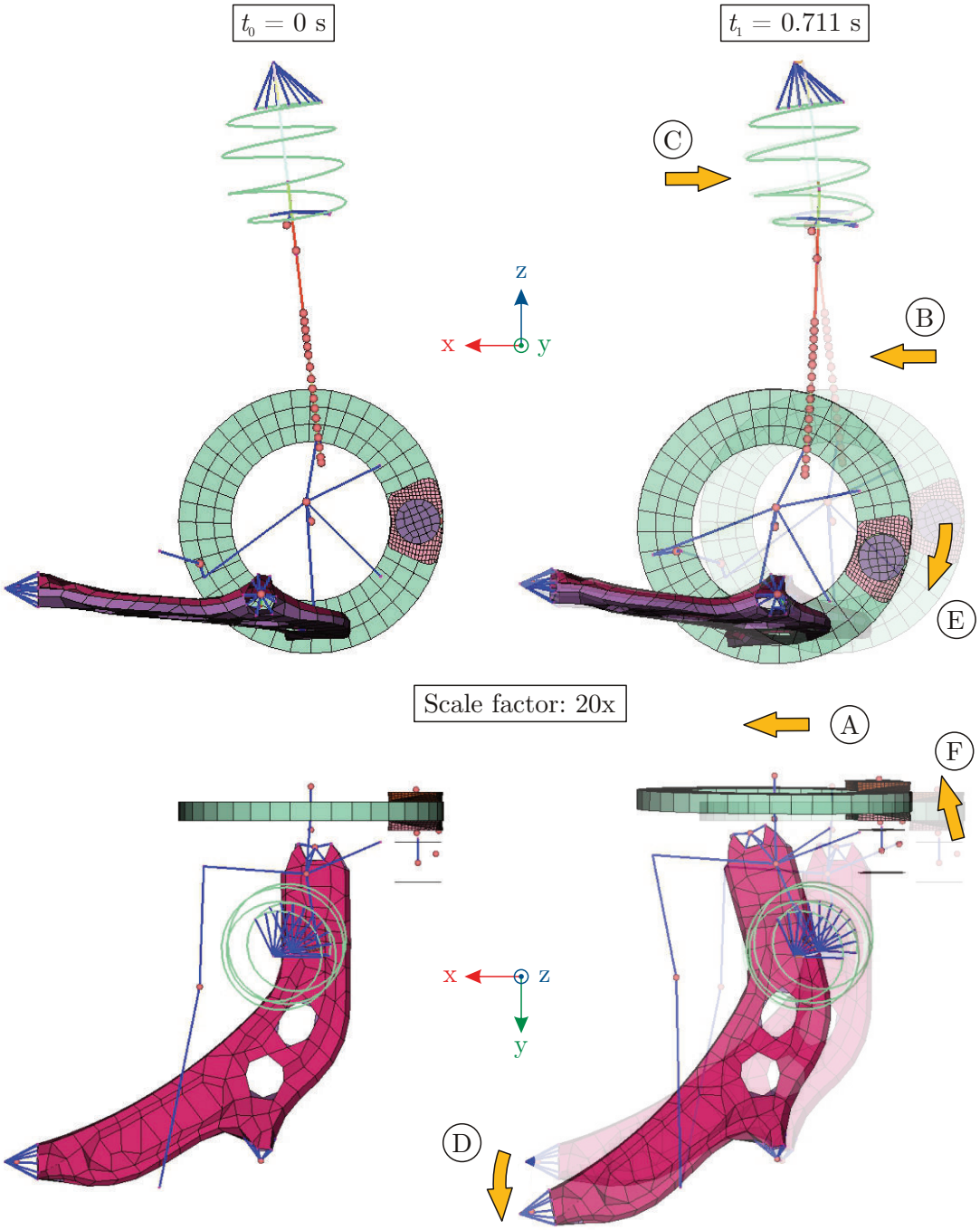

Figure 4.24.: Displacement of ACM_Impl_01 before beginning of the stick-slip phase (scale factor: 20 ; low-frequency creep groan)

The arrows in Figure 4.24 denote the directions of displacement due to load application. Elastic properties of structures and bushings allow for the model to deform as follows. One can see, that the brake disk, pads and caliper are pushed backwards (A) and the lower part of the strut is bent in the same direction (B). Consequently, the coil spring and upper part of the strut move forward (C). The lower control arm follows the displacement of the wheel, thus rotating around its locked boundary. This rotation is followed by the hydro bushing of the control arm (D), since the control arm was modelled as a rigid. Additionally, the disk undergoes a rotatory displacement (E), that is followed by the pads and caliper. Lastly, the disk gets slightly tilted outwards as well (F). This general behaviour is shared by all other variants of the ACM. However, the amount of pretensioning and thus the magnitude of deformation differ between the low- and high-frequency creep groan due to a different brake pressure. The lower pressure used for the high-frequency variants (8 bar instead of 32 bar, see Figure 3.17) leads to an earlier start of stick-slip, less pretensioning and smaller deformations. The displacement until the beginning of the first global slip is depicted in Figure 4.25 for ACM_Impl_05. The same scale factor of 20 was applied.

As one can see, the general displacement in Figure 4.25 is very similar to Figure 4.24. The only qualitative difference can be observed on the upper part of the trust. Due to the earlier beginning of stick-slip at $t_1 = 0.505$ s (before 0.711 s), the disk underwent less rotation (E) until this point. Consequently, less bending moment is applied to the strut and the upper part now gets bent backwards as well (C). The tilting of the disk (F) is also reduced for variants with high-frequency creep groan.

Immediately after t_1, the creep groan phase begins. One full stick-slip repetition is illustrated in Figure 4.26, using the following points in time for low-frequency creep groan, according to Figure 4.23: t_2 as a state of global sticking, right before slipping begins. t_3 denotes the last point of slipping before transition to sticking, which ends at t_4. Thus, t_4 can be understood as the "t_2" of the following cycle.

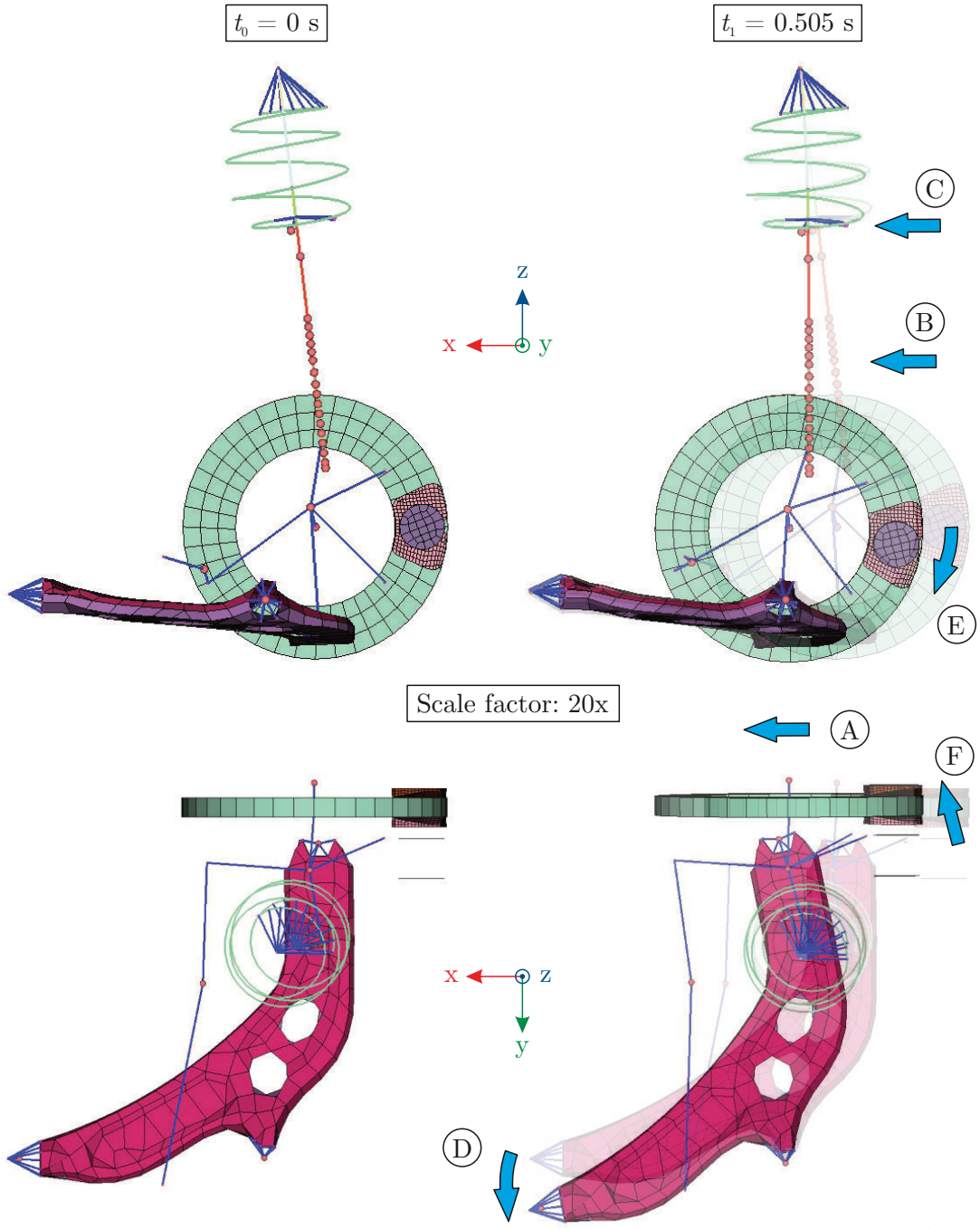

Figure 4.25.: Displacement of ACM_Impl_05 before beginning of the stick-slip phase (scale factor: 20 ; high-frequency creep groan)

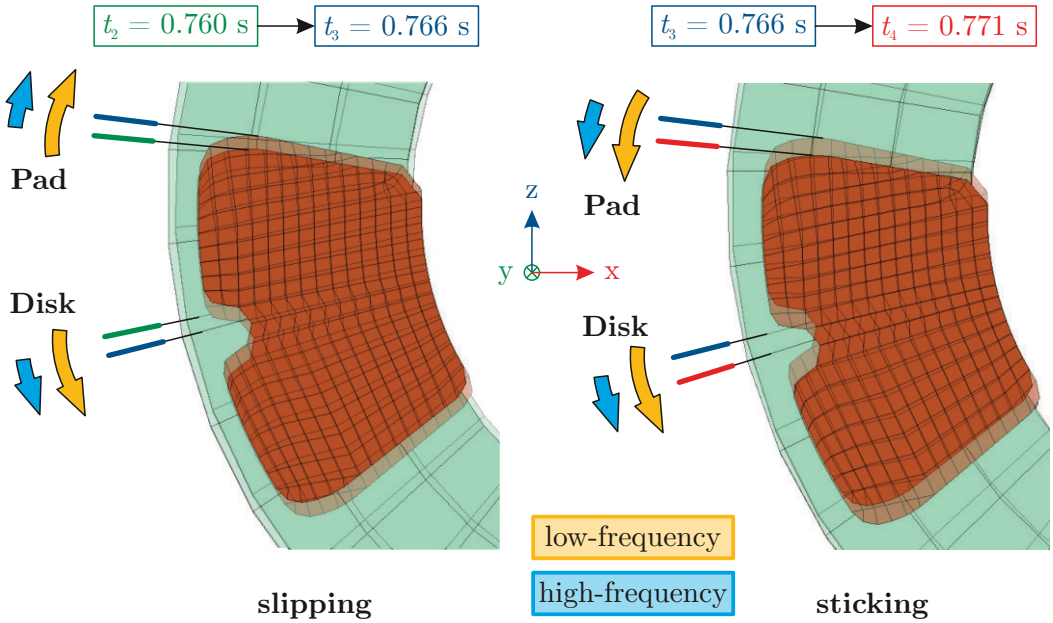

Figure 4.26.: Displacement of ACM_Impl_01 during a stick-slip transition (scale factor: 20)

As one can see, a state close to the beginning of stick-slip was chosen. This was necessary due to an incorrect display of the model in META when applying a scale factor for the displacements. Once the brake disk starts rotating, its diameter expands steadily while the remaining components, including brake pads, are still displayed correctly. As a result, at later stages the disk and pad cannot be properly displayed anymore when using a scale factor. For Figure 4.26, this effect had already begun, but was still at an acceptable level.

The coloured lines represent the positions of the respective components at the given points in time, with the yellow arrows denoting the direction of displacement. As one can see, the process follows the behaviour explained in Chapter 2.1.1. During slipping the disk continues to rotate forward, while the pad slides backwards in opposite direction. During sticking, their directions of rotation match. For the remainder of the simulation, the pads continue to oscillate back and forth for every model variant that produced stick-slip.
ACM_Impl_01 in Figure 4.26 represents low-frequency creep groan. For high-frequency variants, a similar oscillation results, illustrated by the blue arrows. The directions are the same, but a lower amplitude and higher frequency is obtained.

These stick-slip oscillations invoke vibrations in the other components of the model as well. The qualitative behaviour is depicted in Figure 4.27, once again using two sets of arrows for the different frequencies.

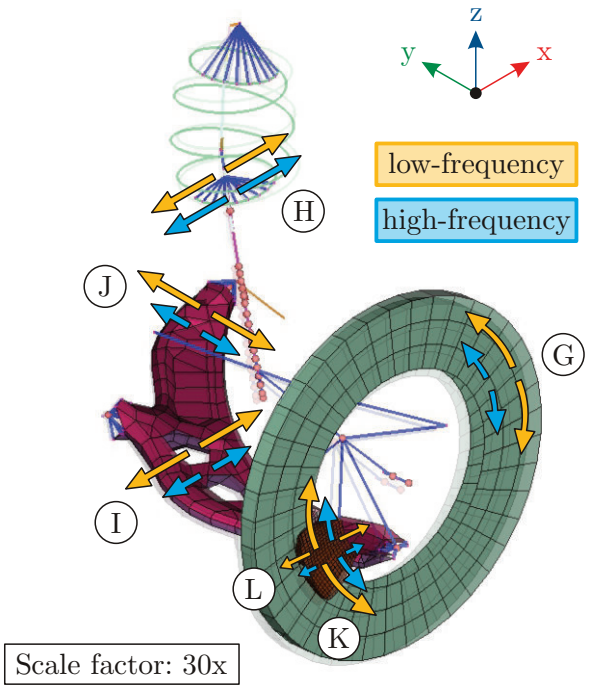

Figure 4.27.: Oscillations of the ACM during stick-slip for low- and high-frequency creep groan (scale factor: 30)

The aforementioned rotatory pad oscillation is included in Figure 4.27 denoted by (K), as well as an additional radial oscillation (L) with significantly lower magnitude that was observed. Once again, the longer yellow arrows indicating greater displacements for low-frequency, and the shorter blue arrows smaller displacements for high-frequency groan. The same applies to rotatory oscillations of the disk (G) and to translational oscillations of the lower control arm in longitudinal (I) and lateral direction (J). Additionally, the brake pads show small radial oscillations as well (L). Only for the strut both versions show similar oscillations (H).

The observed model behaviour matches the expectations well, see [21], therefore the obtained results can be considered an accurate representation of creep groan processes.

4.5. Further investigations

During the evaluation of the results, several additional factors were discovered that will be dealt with in this chapter.

4.5.1. CPUs and computing times

Due to an upgrade of the Abaqus license, a different number of CPUs were used for certain simulations. The initial licence that was used for most simulations on the DPM allowed for up to 4 CPUs to be used simultaneously. During the course of this thesis, a more comprehensive licence was acquired that allowed for any number of CPUs to be used. Since the processor of the simulation computer has a total of 24 logical cores, it was decided, to now run simulations on 16 CPUs to leave some capacities for parallel simulations if needed. Faster computing times were expected as a result of these changes.
However, it was found that the opposite is the case. A higher number of CPUs led to an increase in computing time. This discovery inspired a closer examination of this effect on both models, DPM and ACM.

Disk Pad Model DPM:

The variants of group 2 of the DPM in Abaqus/Standard (see Figure 3.15), that were run on 4 CPUs, were simulated again on 16 CPUs. The computing times and relative changes are listed in Table 4.3.

As one can see, using more CPUs for a simulation results in an increase in computing time for all variants. By comparing the graphs for angular speed and relative velocity it was verified, that the results are identical for both cases. This can also be validated e.g. by comparing the total number of increments that were necessary for the simulation. Table 4.4 shows, the numbers of increments are identical in all cases.

Table 4.3.: Influence of CPUs on computing times of the DPM in Abaqus/Standard

Variant	06		07		08		09		10	
CPUs	4	16	4	16	4	16	4	16	4	16
Comp. time [h]	6.12	6.53	6.05	6.83	6.47	6.85	6.98	7.80	6.03	6.78
Rel. change [%]	+6.81		+12.95		+5.93		+11.69		+12.43	

Table 4.4.: Comparison of the number of increments per simulation of the DPM in Abaqus/Standard

Variant	06		07		08		09		10	
CPUs	4	16	4	16	4	16	4	16	4	16
Number of increments	3136	3136	3154	3154	3197	3197	2966	2966	3080	3080

Axle-corner Model ACM:

For certain variants of the ACM according to Figure 3.16 and 3.17, the same investigations were conducted to determine the influence on this model. The results are listed in Table 4.5 and 4.6 for Abaqus/Explicit and Abaqus/Standard respectively. Due to the greatly increased computing time per variant of the ACM, fewer variants were analysed compared to the DPM.

Table 4.5.: Comparison of computing time and number of increments of the ACM in Abaqus/Explicit

Variant	ACM_Expl_01	
CPUs	4	20
Comp. time [h]	10.85	15.32
Rel. change [%]	+41.17	
Number of increments	5590609	5590609

For simulations of the ACM in Abaqus/Explicit a higher number of CPUs leads to a significant increase in computing time of over 40 % for variant 01. The results and number of increments are identical for both cases. Compared to implicit simulations, explicit simulations use a very large number of increments (several million vs. several thousand). The reason for this is the functionality of their respective integration schemes (see Chapter 2.2.3). The explicit scheme does not rely on iterations per increment, but instead uses a very small time step, thus leading to a tremendous number of increments that are necessary. The implicit scheme on the other hand performs many iterations per increment if necessary, which allows for much larger time steps and a smaller number of increments.

Table 4.6.: Comparison of computing time and number of increments of the ACM in Abaqus/Standard

Variant	ACM_Impl_01		ACM_Impl_05	
CPUs	4	16	4	16
Comp. time [h]	19.87	19.58	26.97	25.43
Rel. change [%]	-1.43		-5.69	
Number of increments	4928	4748	8478	7568

Implicit simulations on the ACM in Abaqus/Standard show a different behaviour. More CPUs resulted in slightly reduced computing times for both the low-frequency creep groan (variant 01), and the high-frequency creep groan (variant 05). This contradicts the behaviour that was observed on the DPM, see Table 4.3, that showed an increase in all cases.
Another difference can be noticed when comparing the total number of increments. Where previously the numbers for 4 CPUs and 16 CPUs were identical, now a slight deviation occurs. Fewer increments correlate with reduced computing time. This deviation suggests, that the number of CPUs impacts the convergence behaviour of simulations on the ACM.
Due to this finding, the solutions were compared to verify identical results, as it was the case for the DPM. However, it was discovered that for the ACM a different number of CPUs has a significant impact on the results as well.
Figure 4.28 provides a comparison of ACM_Impl_01 (low-frequency creep groan) on 4 and 16 CPUs, using the relative velocity between disk and pad in the extended time interval from 1.4 to 1.9 seconds. Both models are identical, besides the number of CPUs as their only difference. As one can see, different results were obtained. The model is still capable of producing creep groan, but the characteristics have slightly changed. The stick-slip frequencies were determined as:

- 17.78 Hz on 16 CPUs
- 18.42 Hz on 4 CPUs

Similar, but not identical frequencies, were obtained for both variants for the low-frequency creep groan.

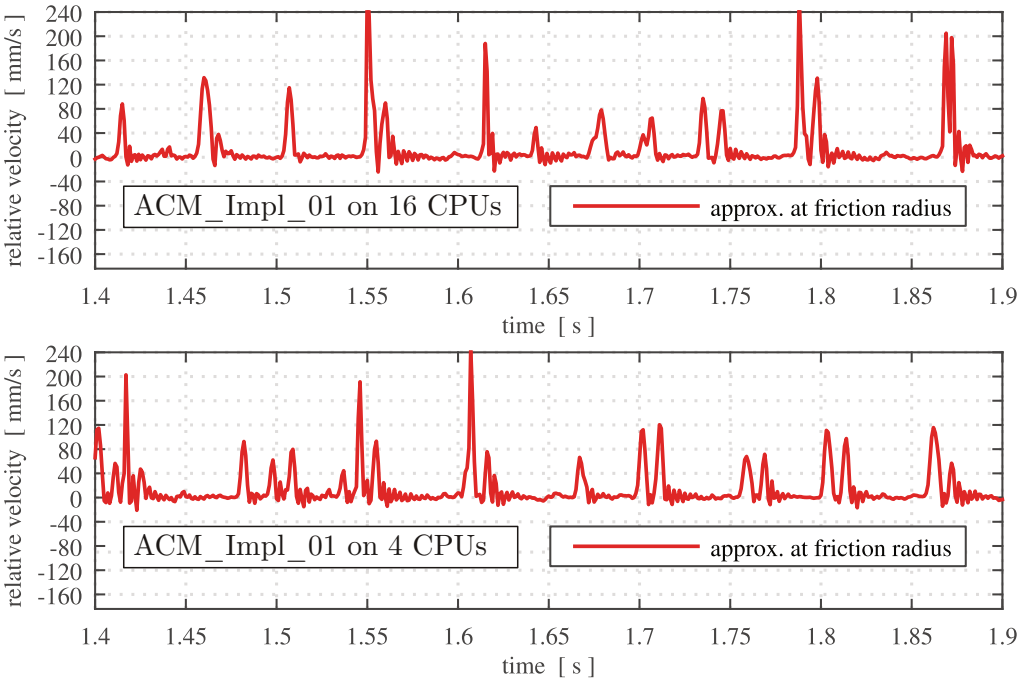

Figure 4.28.: Relative velocity of ACM_Impl_01 for 4 and 16 CPUs (low-frequency creep groan)

A comparison for ACM_Impl_05 (high-frequency creep groan) is provided in Figure 4.29, using the angular speed and relative velocity in the interval from 1.7 to 1.9 seconds. As one can see, running the identical simulation on 4 CPUs instead of 16, severely effects the results. Creep groan can no longer be produced, a chaotic behaviour is the result instead. Thus a stick-slip frequency can no longer be determined.

In order to understand this behaviour, a closer look at the parallelization in Abaqus is necessary:
According to [1], two options for parallel execution are available: threads and message passing. The iterative solver in Abaqus/Standard always uses MPI-based parallelization (MPI = message passing interface). The user has to define a number of CPUs requested for the simulation. Abaqus then splits the model into an equal amount of individual domains. These domains are then run parallel on the assigned CPUs.
It is stated in [1], that inconsistencies of the results can occur for models, that are highly sensitive to small perturbations. Finite precision effects can lead to small numerical differences when running simulations on different numbers of processors. It is thus recommended, to use the same number of CPUs for consistent results.

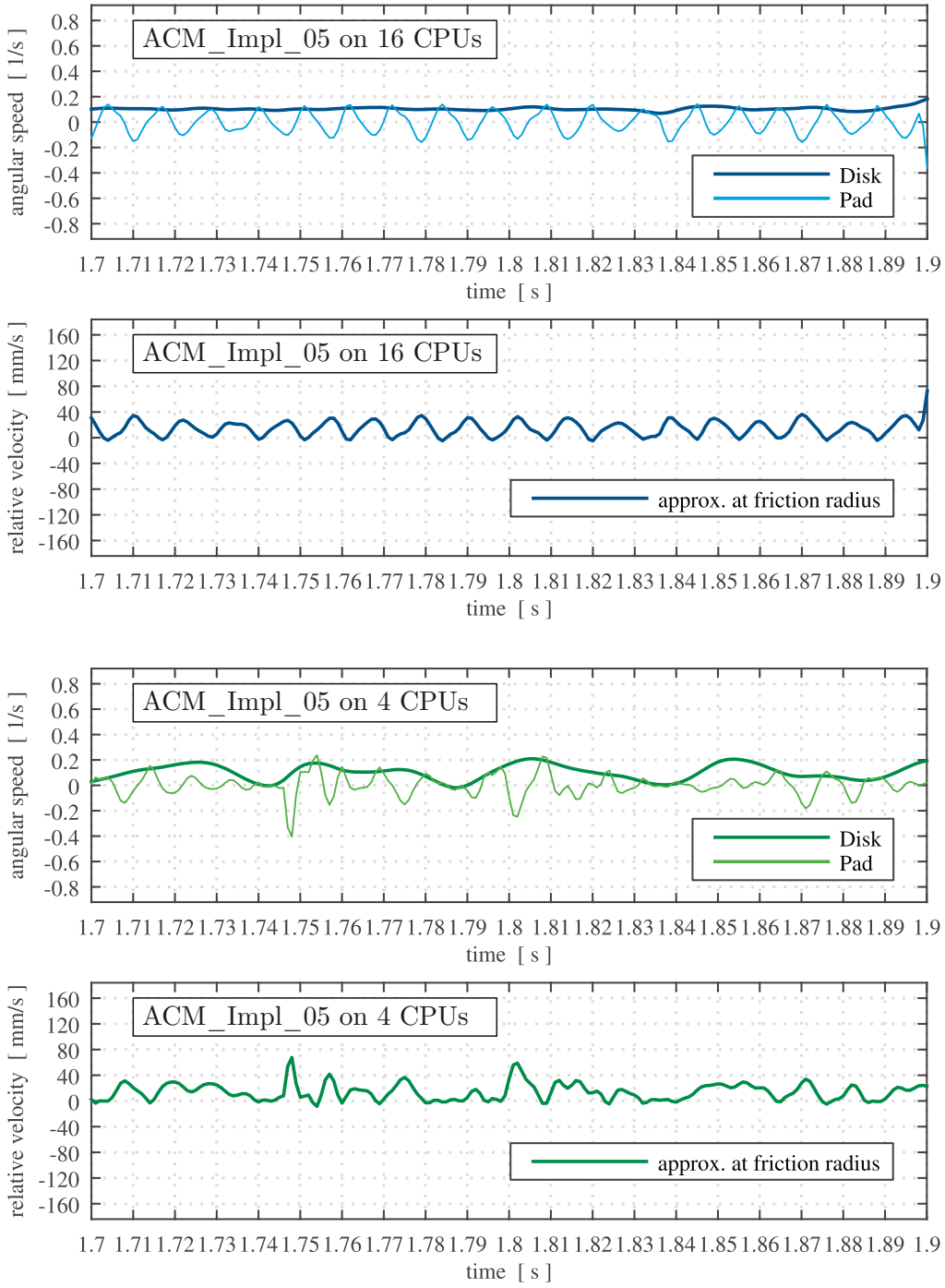

Figure 4.29.: Relative velocity of ACM_Impl_05 for 4 and 16 CPUs (high-frequency creep groan)

Conclusion:

It was discovered, that the number of CPUs used to run a simulation can significantly impact the computing time. Depending on the model, either increases of decreases can be the result. In certain cases, it can furthermore lead to drastic changes of the results in general.

Regarding computing time:

One possible explanation for the observed behaviour could be the following:
As mentioned above, the model has to be split in domains for parallel execution. At certain points during the simulation, e.g. when outputs are requested, information has to be gathered from the CPUs and reassembled. Once all the output data is generated, the data has to be spread out to the domains again. This process can occur many times throughout a simulation. For this work outputs were generated at a frequency of 1 kHz over a simulation time of 2 s, leading to a total of 2000 output requests that need to be generated.
The distribution to and gathering from the CPUs imposes an additional effort, while the parallel calculation on multiple cores should lead to faster results. If now the additional effort exceeds the time saved from using parallelization, the result is an overall increase in computing time (DPM), or vice versa a decrease can occur (ACM). The greatly increased complexity of the ACM compared to the DPM is suspected to be the reason for this phenomenon. Here the parallel calculation of the model outweighs the additional costs of distributing and reassembling. For the simpler DPM and the explicit ACM the opposite seems to be the case.

Regarding differences of results:

As mentioned above, the possibility of deviations of the results due to parallelization is pointed out in [1]. It depends on the sensitivity of the model.
For the simplistic Disk Pad Model (DPM), identical results were obtained for different numbers of CPUs used. The more complex Axle-corner Model (ACM) on the other hand showed severe deviations. For the investigation of creep groan, this effect can be potentially fatal, as Figure 4.29 showed. When working with sensitive models like the ACM, it is thus of utmost importance to ensure the usage of the same number of CPUs in very case for comparable results.
Unfortunately this effect was only discovered in later stages of this thesis and all 12 variants of group 1, 2 and 3 of the ACM would have to be simulated again on 4 CPUs. However, due to long computing times, new simulations could not be performed anymore. Thus, small uncertainties have to be considered for the results of aforementioned variants upon comparison to group 4, 5 and 6.

4.5.2. Contact status

After a simulation was completed, the displacement results were examined in the post-processor to confirm realistic oscillations. The visualization of the contact status (output variable: CSTATUS), in the contact interface revealed differences between explicit and implicit variants. The CSTATUS describes the current state of the contact. The following values are possible:

- CSTATUS = 2 : sticking contact (static friction)
- CSTATUS = 1 : slipping contact (kinetic friction)
- CSTATUS = 0 : no contact (open)

Each node in the contact interface between brake disk and pads has a CSTATUS assigned to it. The contact interface is depicted in Figure 4.30 for the DPM. The two contacts' surfaces are illustrated by the red and blue meshes in Figure 4.30. The finer red meshes represent the contact surface of the brake pad linings, while the partner surfaces of the brake disks are represented by the coarser blue mesh. Since the disk surfaces are rigid, a very coarse mesh is acceptable.

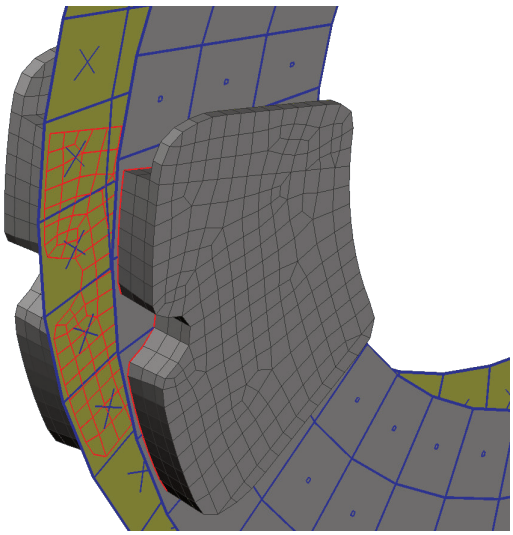

Figure 4.30.: Contact interface between brake disk and pads of the DPM

When analysing and comparing the results of different variants it was discovered, that an unexpected phenomenon occurs for the contact status of implicit simulations. It was expected that after contact has been established, the CSTATUS should alternate between values of 1 and 2 for the remaining simulation time. This was the case for the results of the explicit simulation. The CSTATUS of a single brake pad at different points in time of DPM_Expl_01 is depicted in Figure 4.31. The points in time were roughly chosen as the following: at the beginning (top left), before start of stick-slip (top right), during stick-slip in slipping phase (bottom left), and during stick-slip in

sticking phase (bottom right). As one can see, the results are as expected. Status 1 for the slipping phase, otherwise status 2.

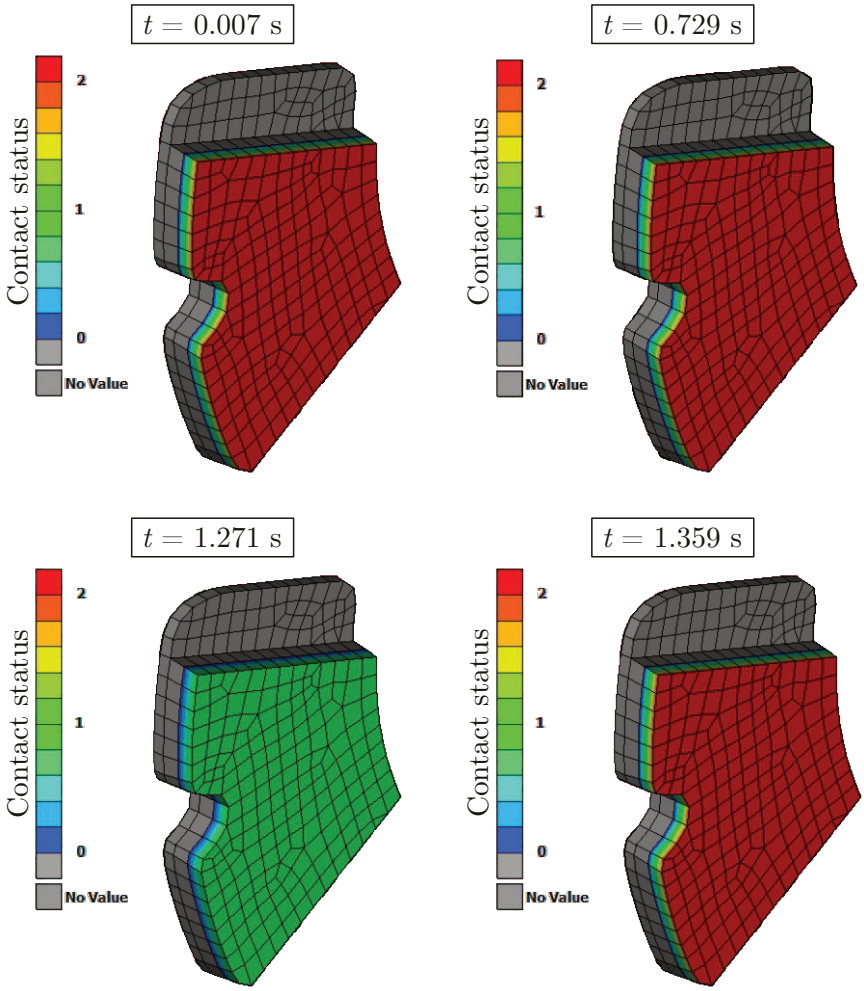

Figure 4.31.: Contact status at different stages during the simulation of DPM_Expl_01

The analogous figures for the implicit variant DPM_Impl_06 are depicted in Figure 4.32. Initially, all nodes were in contact (top left), as it was the case for DPM_Expl_01. But then certain nodes start to lose contact for the rest of the simulation. Information about the contact status for every node at every time step is written to the output file and can be accessed. Instead of a value of 1 or 2, the value -3.40282E+38 was gained for the nodes in question. This represents the maximum possible value for the data type "float" and was interpreted as a failed contact. As already mentioned, only values of 0, 1, or 2 should be possible. The faulty values were thus replaced by the value "0" instead, which means no contact.

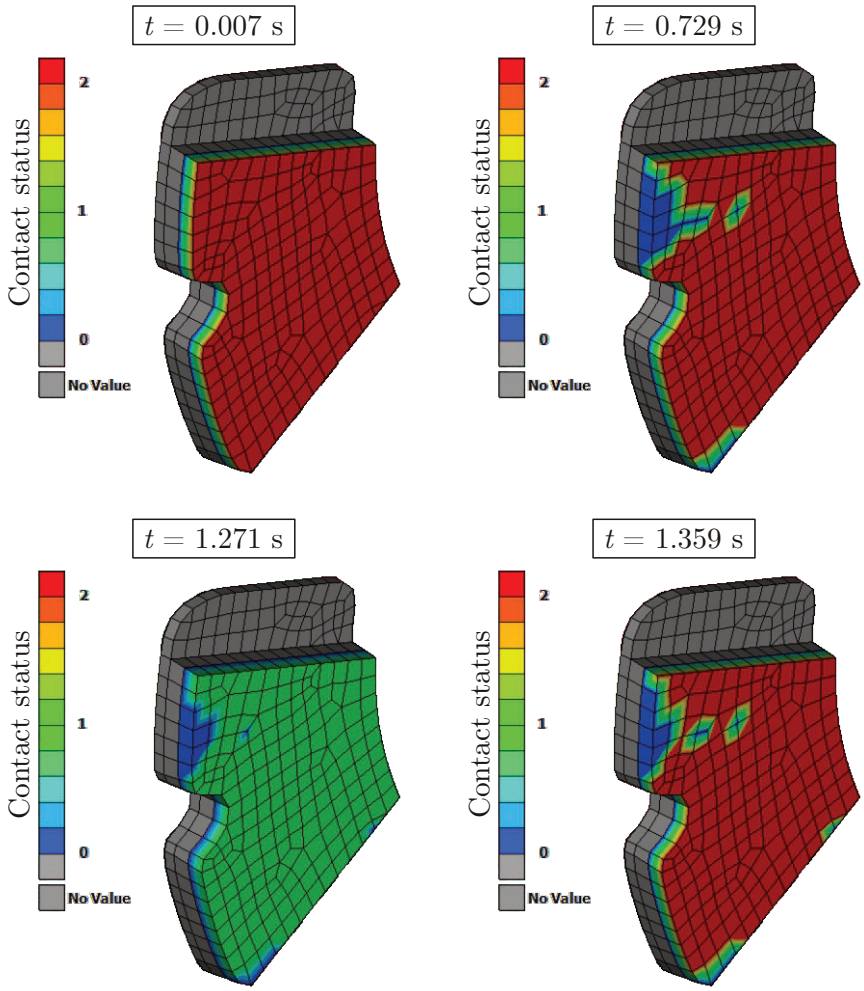

Figure 4.32.: Contact status at different stages during the simulation of DPM_Impl_06

After this was discovered, certain modifications to the model were made in an attempt to achieve a correct contact for all nodes. The following numerical options were explored for a variety of variants, but did not lead to a disappearance of this phenomenon:

- CONTACT STABILIZATION
- CONTACT INITIALIZATION:
  - INTERFERENCE FIT
  - MINIMUM DISTANCE
  - SEARCH ABOVE

Since the problem seemed to be mostly associated with nodes at the edge of the surface, the contact surface was newly defined. The surface was extended around the corner to also include the adjacent nodes. This new contact interface is depicted in Figure 4.33.

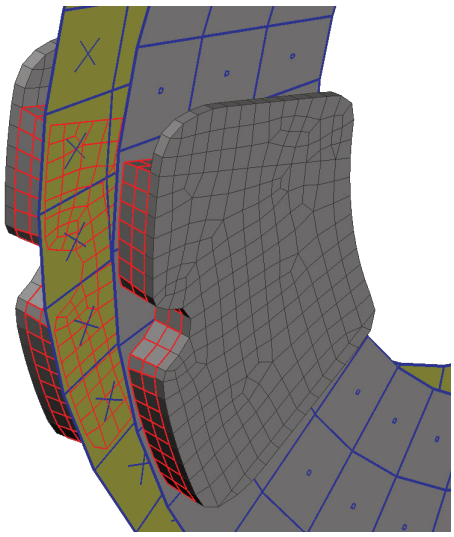

Figure 4.33.: New contact interface variant between brake disk and pads of the DPM

The results revealed however, that this change did not affect the contact status in any way. Therefore, the usual contact surface was continued to be used, see Figure 4.30.

The same effect was observed on the ACM as well. Here the deviation between explicit and implicit is severe and can be best illustrated by comparing the contact statuses during a stick-slip transition. The process is depicted in Figure 4.34 and 4.35 for an explicit and implicit simulation respectively. Three consecutive outputs during a transition are used for both variants.

When comparing the two figures one can notice obvious differences. The explicit variant (Figure 4.34) shows contact for all nodes in the contact interface. When transitioning from kinetic to static friction, first individual nodes start to stick. This is followed by more nodes until the majority is sticking, and the transition was successful.
The implicit solution (Figure 4.35) shows a different behaviour. Here certain nodes in the bottom right corner have no contact. The number of nodes without contact varies throughout the simulation, but a certain amount is present in nearly every output. The transition to static friction is very abrupt for this variant. All nodes with status 1 (slipping) switch to status 2 (sticking) within one output time step of 1 ms.

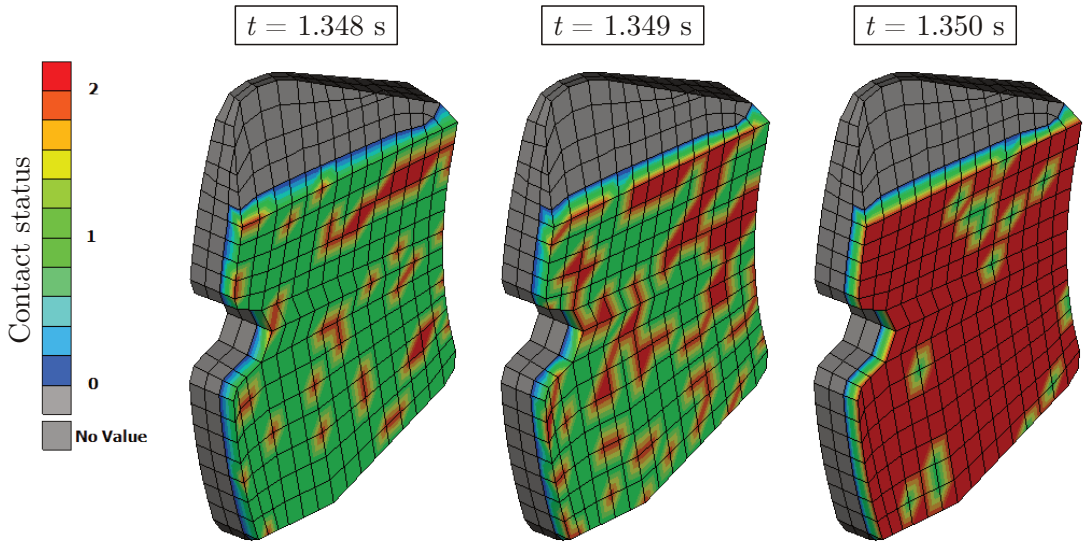

Figure 4.34.: Pad lining contact status during a slip to stick transition of ACM_Expl_01

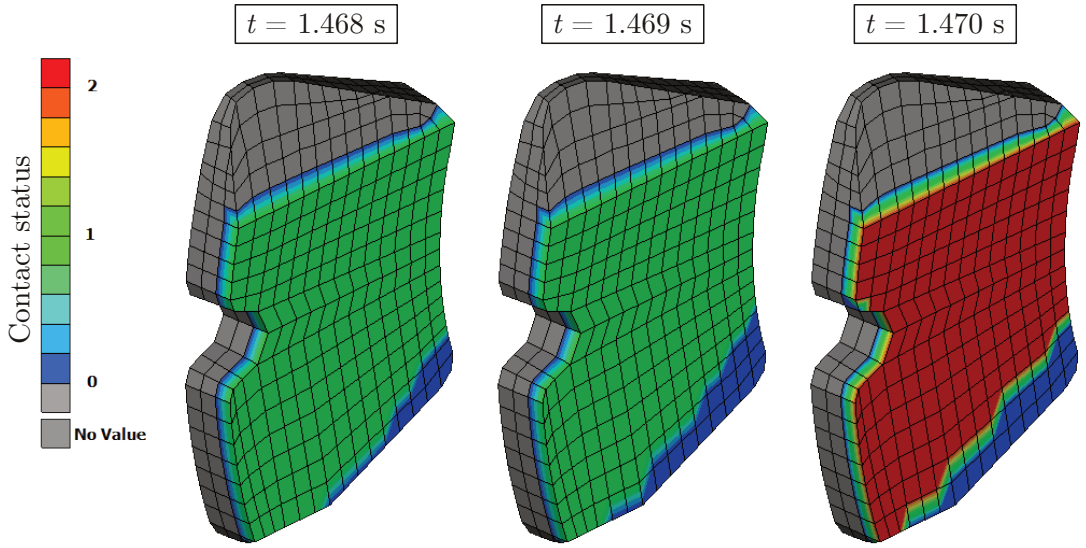

Figure 4.35.: Pad lining contact status during a slip to stick transition of ACM_Impl_01

The observation of nodes without contact in the implicit results inspired a closer examination of this phenomenon. The following method was used to describe and compare different variants:
For every output time step, the total number of nodes with status 0, 1 or 2 were calculated. An average number of nodes without contact throughout the entire simulation

time of a certain variant can then be determined. With a combined total of 432 nodes on both sides of the contact interface, this leads to a percentage of nodes without contact for a variant.
This percentage was found to be a reasonable tool for quantifying the quality of a contact interaction and was thus calculated for every variant of the DPM and the ACM in both Abaqus/Explicit and Abaqus/Standard. The results are listed in Figure 4.36 below for the DPM in Abaqus/Standard.
For the explicit variant DPM_Expl_01, a percentage of 0.32 % of total nodes without contact was calculated.

Group	1	2	3	4	5	6	7
Variant number	01 - 05	06 - 10	11 - 15	16 - 20	21 - 25	26 - 30	31 - 35

SIM	**Percentage of total nodes without contact [%]**						
Augm. Lagrange	5.46	5.44	4.93	3.28	5.50	5.55	5.50
Penalty nonlinear	3.45	3.43	3.35	2.17	3.50	3.55	3.48
Penalty linear	4.44	4.37	4.26	3.21	4.45	4.46	4.47
Direct linear	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Direct hard	5.49	5.45	5.17	3.29	5.51	5.52	5.51

Figure 4.36.: Percentage of total nodes without contact of the DPM in Abaqus/Standard on both brake pads

The following conclusions can be derived based on these results:
Problems with contact status only occur to a significant degree for implicit simulations. The quality of the contact is strongly correlated to the surface interaction model (SIM) applied. Specifically for variants with "Direct linear", or explicit simulations, the contact status problem does not occur. The value of 0.05 % originates from the very first increment, where all nodes are without contact. For the remainder of the simulation, all nodes are in contact across all groups of variants for this SIM. Furthermore, "Augmented Lagrange" and "Direct hard" show the worst contact quality in all cases. The Penalty contact is slightly better, with "nonlinear" showing improved quality compared to "linear". Between the groups, almost identical results are obtained for the same SIM. The exception is group 4 as the only one with "Contact pair" instead of "General contact". Here fewer nodes without contact are observed across all SIMs.
Upon closer investigation of the variants with "Direct linear" it was found, that the stiffness of the linear pressure-overclosure model (see Figure 2.21) influences the occurrence of the phenomenon. After running simulations with different slopes of the stiffness, the maximum allowed slope before the first nodes start to lose contact was determined as $\text{Slope}_{\text{max}} = 46$. The variant DPM_Impl_09 was used for this exemplary investigation.

For the variants of the ACM analogous values were calculated. Here the total number of nodes on both sides of the contact interface is given as 462. The results are listed in Figure 4.37 and 4.39 for the explicit and implicit variants respectively.

Variant number	01	02	03	04	x1	x2	x3
SIM	**Percentage of total nodes without contact [%]**						
Penalty linear	0.27	0.27	13.02	13.02	0.13	0.28	0.28

Figure 4.37.: Percentage of total nodes without contact of the DPM in Abaqus/Explicit

Unlike the DPM, problems with contact status were detected for explicit simulations as well on the ACM. For variant 03 and 04 the problem is severe with 13 % of all nodes without contact. As it was explained in Chapter 4.3.1.1, these variants represent variant x2 after changes were made to the model. It was already stated, that the changes led to incorrect results (see Figure 4.13) for the high-frequency creep groan. The information about contact status reveals an increase of total nodes without contact from 0.28 % to 13.02 % as a result of the changes. This was identified as the reason for the failed stick-slip behaviour. The contact status in the interface is depicted in Figure 4.38 for ACM_Expl_03 at later stages of the simulation. Here, stick-slip effect did not occur anymore, see Figure 4.13.
For the low-frequency creep groan, the results match before and after the changes (see Figure 4.12). This is also confirmed by similar results for nodes without contact of 0.13 % and 0.27 % for variant x1 and 01.

The implicit results of the ACM show the same behaviour as the ones of the DPM. Once again, "Augmented Lagrange" led to the highest number of nodes with failed contact, followed by "Penalty linear" and "Penalty nonlinear". For "Direct linear", the problem did not occur. The setting "Convert SDI" had almost no impact on the low-frequency creep groan (group 1 vs. 4). For the high-frequency creep groan slight differences can be observed (group 2 vs. 5). Since these differences are very small, this does not provide a satisfying explanation for the observed failure to produce stick-slip with group 5 (see Figure 4.17).
The ACM revealed further, that a problem with contact status does not only occur with stick-slip effects. For the variants of group 3 and 6 with constant sliding between disk and pads, the same problem was found.

An extensive investigation of the influences on the contact interface of disk brakes due to structural modifications using FEA was performed in [4]. The modelling of the connection between the brake piston (application of the pressure) and the backing plates of the pads was found to be of crucial importance. Significant improvements to contact area and pressure distribution could be made, [4].
For the matter of creep groan, especially the contact pressure distribution is suspected

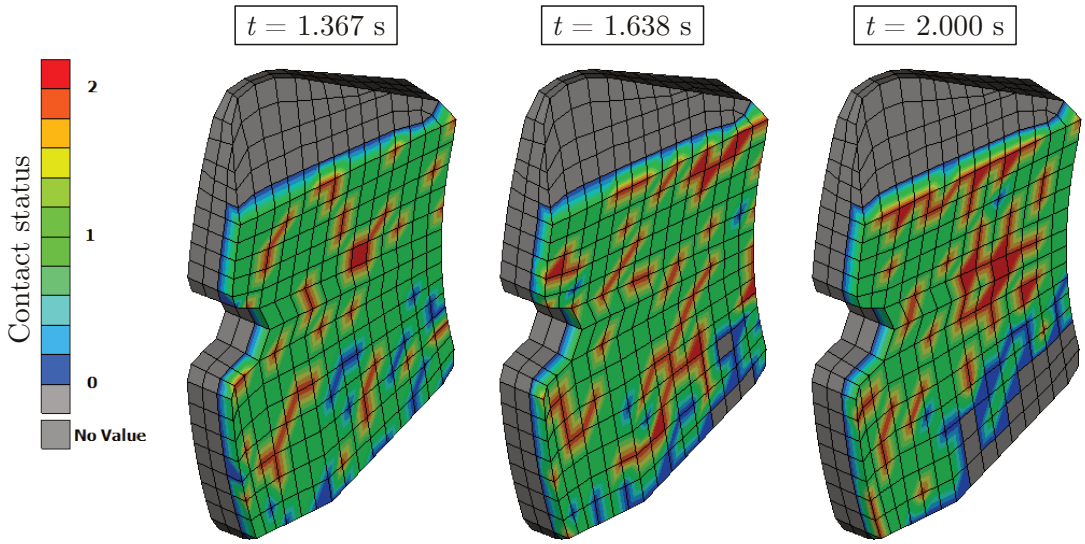

Figure 4.38.: Contact status at later stages of ACM_Expl_03

Group	1	2	3	4	5	6
Variant number	01 - 04	05 - 08	09 - 12	13 - 16	17 - 20	21 - 24
SIM	**Percentage of total nodes without contact [%]**					
Augm. Lagrange	3.46	4.45	3.60	3.47	5.75	3.16
Penalty nonlinear	2.42	3.21	2.59	2.44	2.91	2.62
Penalty linear	3.01	3.90	3.11	2.98	3.51	3.60
Direct linear	0.07	0.02	0.00	0.07	0.00	0.00

Figure 4.39.: Percentage of total nodes without contact of the ACM in Abaqus/Standard

to severely impact the resulting stick-slip behaviour. Changes to the contact status are also expected and modifications to the piston/pads connection may resolve the problem with nodes losing contact. For future investigations, careful attention should be paid to the modelling of these elements.

5. Discussion of the results

Based on the results presented in the previous chapter, a best option for implicit simulations will be sought in this chapter. It will then be compared to the explicit simulation and a final rating will be derived. The most important aspects for rating a variant are the computing time and the reliable production of a stable stick-slip behaviour.

5.1. Analysis of the Disk Pad Model DPM

The following conclusion were drawn from the DPM simulation results presented in Chapter 4.2:

- Stick-slip can be produced with every surface interaction model (SIM).
- GENERAL CONTACT is preferable to CONTACT PAIR, especially for means of comparison between implicit and explicit.
- "Convert SDI = no" can help to improve the implicit solution.
- Changes to the numerical dissipation can have unpredictable effects.
- Problems with contact status occur with implicit simulations. The exception is the SIM "Direct linear".

The most promising variants of the DPM in Abaqus/Standard were found to be group 1 and 2. The creep groan results are almost identical for all cases. Thus, the variant with the shortest computing time can be considered the best one. The following variants were the fastest:

1. DPM_Impl_03 with 5.97 h and 42.25 Hz
   Penalty linear; Convert SDI = yes; 4 CPUs;
2. DPM_Impl_07 with 6.05 h and 41.18 Hz
   Penalty nonlinear; Convert SDI = no; 4 CPUs;
3. DPM_Impl_06 with 6.12 h and 42.11 Hz
   Augmented Lagrange; Convert SDI = no; 4 CPUs;

As one can see, very similar computing times were achieved with different SIMs. The penalty contact however leads, in theory, to a less accurate representation of a realistic, physical contact interaction. This is due to the fact that a certain amount of penetration is necessary for this method. For augmented Lagrange this is not the case.

Since the results are almost identical for the variants listed above, this factor does not seem to matter for the DPM.

Comparison to explicit:

The following computing time and stick-slip frequency was obtained in Abaqus/Explicit:

- DPM_Expl_01 with 9.8 h and 38.46 Hz
  Penalty linear; Convert SDI = yes; 4 CPUs;

This means a reduction in computing time of 39.1 % was achieved with the fastest implicit variant compared to explicit.

The characteristics of the results differ slightly between explicit and implicit simulations, see Figure 5.1 below. Unlike the explicit variant, the implicit one shows a short transient after every stick-slip event. Further, the peaks of DPM_Expl_01 are almost perfectly equal in height, shape and frequency. For DPM_Impl_03 this is not the case. It is however unclear, whether the former or latter better represents realistic physical behaviour.

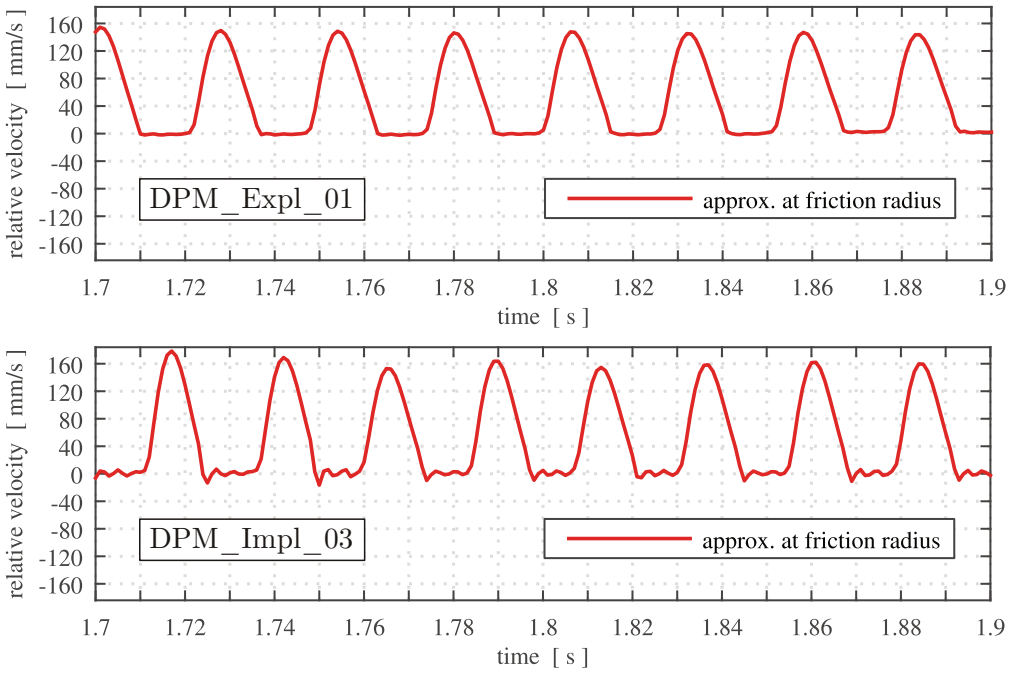

Figure 5.1.: Relative velocity of the brake disk and pads of DPM_Expl_01 and DPM_Impl_03

Conclusion:
Implicit direct time integration shows better performance than explicit direct time integration on the simpler DPM. It delivers similar results but is significantly faster. Reductions in computing time of up to 39.1 % were achieved.

5.2. Analysis of the Axle-corner Model ACM

The following conclusions were drawn from the ACM simulation results presented in Chapter 4.3:

- Stick-slip can be produced at both frequencies found on the test rig with explicit and implicit simulations.
- A complex creep groan model like the ACM is highly susceptible to any changes to its parameters and solver settings. This is especially the case for the high-frequency creep groan.
- The computing times for similar CPUs largely depend on the surface interaction model (SIM).
- "Penalty nonlinear" leads to significantly longer computing times than "Penalty linear".
- Explicit simulations take similar amounts of time for the DPM and for both frequencies of the ACM.
- Implicit simulations vary strongly in simulation time, depending on the model and solver settings.
- Problems with contact status, namely certain nodes not finding contact, can also occur with explicit simulations.

Additionally to the assessment factors used for the DPM (computing time and stick-slip stability), the similarity to the test rig results (21.7 Hz and 87.1 Hz) has to be considered for the ACM.

Low-frequency creep groan:

The results for every variant of the low-frequency creep groan are depicted in Figure 4.19 and 4.20. The fastest variant (computing time: 13.57 h), ACM_Impl_13, has to be disqualified due to a collapsing stick-slip motion. The SIM "Augmented Lagrange" in general shows deviations from the other SIMs in form of larger and more uneven spikes. ACM_Impl_16 leads to inconsistencies as well in comparison to the other variants with very similar behaviour.
With comparable stick-slip frequencies but vastly different computing times, the following variants showed the best performance for the low-frequency creep groan in Abaqus/Standard:

1. ACM_Impl_15 with 14.5 h and 19.35 Hz
   Penalty linear; Convert SDI = no; 4 CPUs;
2. ACM_Impl_04 with 15.1 h and 18.60 Hz
   Direct linear; Convert SDI = yes; 16 CPUs;

ACM_Impl_15 combines the fastest computing time and a stick-slip frequency closer to the measured 21.7 Hz for this model.

High-frequency creep groan:

The results for the valid variants of the high-frequency creep groan are depicted in Figure 4.16. Once again the SIM "Augmented Lagrange" deviates due to a less consistent pattern of the stick-slip oscillations. The degree of deviation is however still acceptable and represents a correct stick-slip motion.
The following variants showed the best performance for the high-frequency creep groan in Abaqus/Standard:

1. ACM_Impl_05 with 25.43 h and 87.50 Hz
   Augmented Lagrange; Convert SDI = yes; 16 CPUs;
2. ACM_Impl_07 with 35.1 h and 88.24 Hz
   Penalty linear; Convert SDI = yes; 16 CPUs;

ACM_Impl_05 is by far the fastest variant for this model and yields a stick-slip frequency almost identical to the measured 87.1 Hz.

Comparison to explicit:

The following computing time and stick-slip frequency was obtained in Abaqus/Explicit for the fastest variant with low-frequency creep groan:

- ACM_Expl_02 with 9.97 h and 21.98 Hz
  Penalty linear; Convert SDI = no; 4 CPUs;

As mentioned in Chapter 4.3.1.1, the changes to the explicit model resulted in a collapse of the high-frequency creep groan. Therefore, the already existing variant x2 has to be used as only option for comparison. The following was obtained:

- ACM_Expl_x2 with 10.63 h and 88.89 Hz
  Penalty linear, Convert SDI = yes; 4 CPUs;

Analogous to the DPM in the previous chapter, a comparison of the results of the fastest explicit and implicit variants for either creep groan frequency shall be given. The respective graphs for relative velocity are depicted in Figure 5.2 and 5.3.

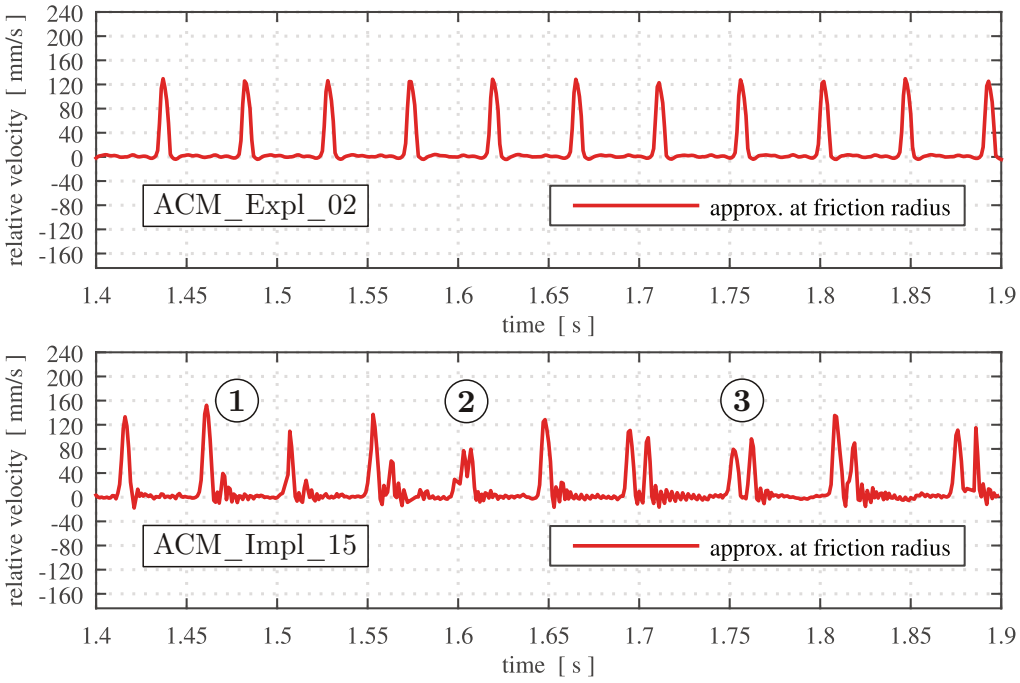

Figure 5.2.: Relative velocity of the brake disk and pads of ACM_Expl_02 and ACM_Impl_15 (low-frequency creep groan)

A phenomenon similar to the DPM can be observed when comparing the characteristics of the curves in Figure 5.2. The previously discovered transient occurs at implicit solutions of the ACM as well, but to a greater degree, e.g. (1). Additionally, the graph shows deviations from the regular single peak per stick-slip event. Double-peaks of different shapes, e.g. (2) and (3), are commonly found in all implicit variants with low-frequency creep groan (see Figure 4.19 and 4.20).
Upon comparison with the test rig results (see Figure 3.21), this phenomenon can be observed as well. It can thus be concluded, that the behaviour of the implicit solution is more realistic than the explicit solution.

For the high-frequency creep groan both versions are similar. The graph of the explicit solution shows a more "round" characteristic in comparison to the implicit one. This indicates smoother stick-slip transitions with explicit simulations.
Differences in shape of consecutive peaks can be observed for both variants, but to a greater degree for the implicit solution. When comparing to the results of the second fastest option, ACM_Impl_07, this behaviour is barely noticeable (see Figure 4.16). It can thus be concluded, that the SIM "Augmented Lagrange" is responsible for this phenomenon.

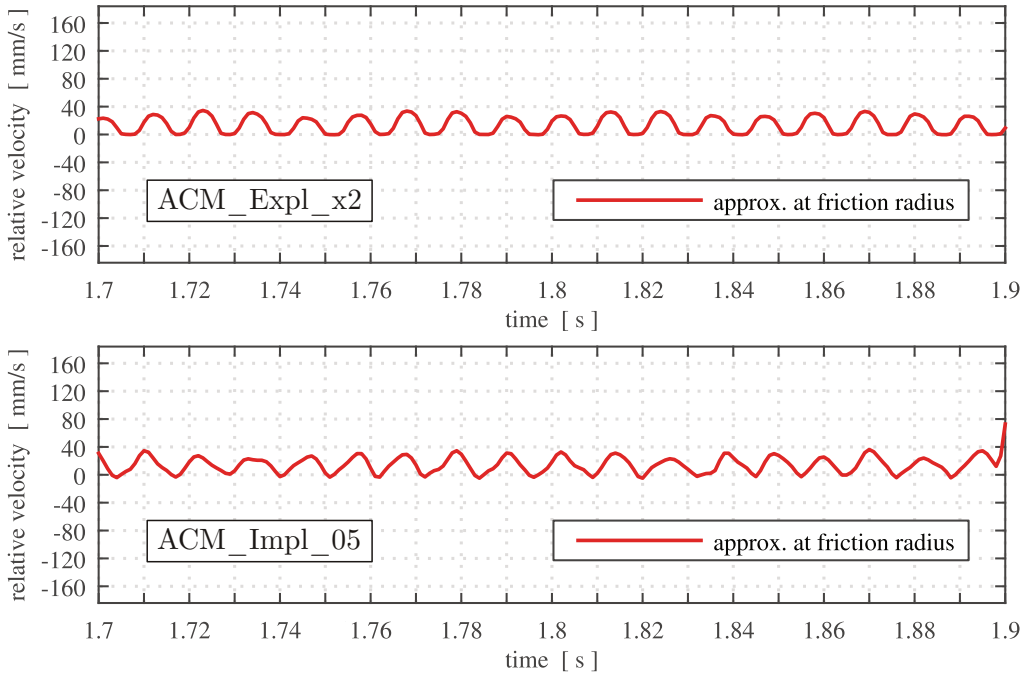

Figure 5.3.: Relative velocity of the brake disk and pads of ACM_Expl_x2 and ACM_Impl_05 (high-frequency creep groan)

Conclusion:

For the more complex ACM, explicit direct time integration shows better performance than implicit direct time integration in all cases analysed. Compared to the DPM, a larger deviation of the characteristics of the results can be observed. The obtained stick-slip frequencies are however very similar. For the low-frequency creep groan, implicit simulations lead to an increase in computing time of 45.4 % in the best case. For the high-frequency variant, increases of 139.2 % were found. For certain contact models, the computing times can increase tremendously, up to over 400 %.

5.3. Comparison: Explicit vs. Implicit

For the simulation of creep groan with direct time integration, the following aspects were found to have the most significant influences on the explicit and implicit method: the complexity of the model, and the stick-slip frequency. The frequency determines the amount of stick-slip transitions that occur throughout a simulation, which impose severe discontinuities on the model. These discontinuities are very demanding for the iterative implicit time integration scheme. Very small time increments and many iterations are necessary in order to achieve convergence. Thus, a high-frequency creep groan with a large number of stick-slip transitions results in very long computing times with implicit simulations.
The explicit scheme does not perform iterations but uses a very small, but constant, time increment instead. Therefore, the stick-slip frequency has very little impact on the computing time with explicit simulations.
Additionally, the overall complexity of the model strongly effects the computing times for implicit simulations. With more components and elements in the model it is more difficult for the implicit scheme to achieve an equilibrium. Longer computing times are the result.
For explicit simulations this is once again not the case. Here a great increase in complexity (DPM vs. ACM) does not lead to longer computing times.

The above statement can be validated by comparing the computing times in Figure 5.4. This figure provides an overview of the fastest variants for both models in Abaqus/Explicit and Abaqus/Standard. As one can see, the computing times for explicit simulations are very similar across all three models. For implicit simulations this is not the case. Here the computing times greatly increase with the complexity of the model. Regarding the high-frequency creep groan of the ACM, one has to keep small uncertainties in mind due to a difference in CPUs used (see Chapter 4.5.1.

Besides the computing times, another crucial differentiation between explicit and implicit simulation of creep groan was found to be the characteristics of the results. The explicit solution showed very clear, distinct peaks. These peaks are furthermore highly consistent in shape, amplitude and frequency.
For the implicit solution, these peaks show a more inconsistent behaviour. This is especially the case for the low-frequency creep groan on the ACM (see Figure 5.2).
Results of the test rig experiments conducted in [21], [20] suggest, that the implicit solution is a more accurate representation of realistic physical behaviour.

		Explicit simulation (Abaqus/Explicit)	**Implicit simulation (Abaqus/Standard)**
Disk Pad Model DPM	Name:	DPM_Expl_01	DPM_Impl_03
	SIM:	Penalty linear	Penalty linear
	Conv. SDI:	yes	yes
	Frequency:	38.46 Hz	42.25 Hz
	Comp. time:	**9.8 h** (4 CPUs)	**5.97 h** (4 CPUs)
Axle-corner model ACM (low-frequency creep groan)	Name:	ACM_Expl_02	ACM_Impl_15
	SIM:	Penalty linear	Penalty linear
	Conv. SDI:	no	no
	Frequency:	21.98 Hz	19.35 Hz
	Comp. time:	**9.97 h** (4 CPUs)	**14.5 h** (4 CPUs)
Axle-corner Model ACM (high-frequency creep groan)	Name:	ACM_Expl_31	ACM_Impl_05
	SIM:	Penalty linear	Augmented Lagrange
	Conv. SDI:	yes	yes
	Frequency:	88.89 Hz	87.5 Hz
	Comp. time:	**10.63 h** (4 CPUs)	**25.43 h** (16 CPUs)

Figure 5.4.: Overview of the fastest variants for the DPM and ACM in Abaqus/Explicit and Abaqus/Standard

6. Conclusions

In the course of this work, various methods for the direct, transient simulation of creep groan using Finite Element Analysis were investigated on two different models, DPM and ACM.

The occurrence of creep groan depends on a large number of influence factors such as operating parameters (brake pressure, velocity, etc.), frictional and material properties of the pad linings as well as their overall shape, damping and elasticities of auxiliary components, bushing properties, and many more that should be considered. Therefore, the development of FE models for the simulation of creep groan is a challenging task and their solution computation-intensive, with a wealth of solver settings and modelling options available.

With both given models used for this thesis, a stable stick-slip limit cycle characteristic for creep groan was successfully produced with explicit and implicit methods.
Many of the aforementioned influence factors are considered in the ACM but were mostly left unchanged, since the main objective was the comparison and evaluation of solution methods. The definition of the contact interface between brake disk and pad linings showed to be a crucial modelling aspect and various options were thus explored. The second main focus was the influence of solver settings, namely of the implicit HHT-α solver.

For identical model settings, differences in results, mainly regarding computing times and overall characteristics, were discovered based on the solution method used. The complexity of the model especially, as well as the obtained stick-slip frequency showed to significantly impact this deviation.

Regarding computing times:

Computing times for explicit simulations remain mostly constant regardless of model complexity, since it is primarily determined by the time increment size and total simulation time. The increment size is based on the fineness of the mesh, as well as the density and stiffness of the materials.

For implicit simulations, the computing times mainly depend on the convergence behaviour of the iterative solution scheme. In general, the higher the stick-slip frequency, and thus the larger the number of severe discontinuities, the longer the computing time. Finer meshes and stiffer materials are however expected to lead to a further increase as well. Additionally, special attention has to be paid to the surface interaction

model used, since it can tremendously increase the computing time. Based on findings within this work, the linear penalty method or augmented Lagrange method can be recommended. Both options should however be explored, since the outcome may differ depending on the model parameters.

For a realistic representation of creep groan, a certain complexity of the model will be necessary. Thus, if short computing times are desired, explicit simulations are expected to be significantly faster than their implicit counterpart. However, many additional solver settings can be modified and one might be able to obtain faster results with implicit, than what was achieved within this work. Overall, the implicit method provides a lot more options than explicit and further investigations on this topic would be needed.

Regarding characteristics:

Especially for the case of low-frequency creep groan on the ACM with a major stick-slip interval of approximately 20 Hz, significant differences between the graphs gained with explicit and implicit simulations were discovered. For the high-frequency creep groan of approximately 88 Hz, differences were observed as well, but to a lesser extent. Based on results available from test rig experiments, implicit was found to be a more accurate representation of realistic physical behaviour. Nonetheless, the obtained stick-slip frequencies are very similar between both options. Unlike the computing times, the characteristics of the results were found to be mostly determined by solver settings, while the influence of the surface interaction model was small in comparison.

Ultimately, a combined approach of computer simulation and test rig experiments is deemed necessary to gain a better understanding of the phenomenon creep groan and its prevention. At this point, no clear recommendation for an explicit or implicit approach can be given.

For further investigations on this subject the following strategy could be promising. At first, modifications to the model should be made (e.g. brake pressure, velocity, friction and material parameters, damping and elasticities, etc.) and their effects on the obtained stick-slip behaviour analysed. Based on these findings, analogous modifications can be made to the test rig. The experimentally obtained outcome can then be compared to the predicted behaviour from the simulation. With this approach, the simulation can either be validated, or valuable clues for model improvements gained.

List of Figures

List of Tables

Bibliography

[1] *Abaqus 2016 Online Documentation*. Dassault Systèmes, 2015. [Online]. Available: http://130.149.89.49:2080/v2016/index.html.

[2] M. K. Abdelhamid and W. Bray, "Braking systems creep groan noise: Detection and evaluation", *SAE Technical Papers*, 2009. DOI: 10.4271/2009-01-2103.

[3] H. Abendroth and B. Wernitz, "The integrated test concept: Dyno - vehicle, performance - noise", *SAE Technical Papers*, 2000. DOI: 10.4271/2000-01-2774.

[4] A. R. Abu Bakar, H. Ouyang, and Q. Cao, "Interface pressure distributions through structural modifications", *SAE Technical Papers*, 2003.

[5] A. R. AbuBakar and H. Ouyang, "Complex eigenvalue analysis and dynamic transient analysis in predicting disc brake squeal", *International Journal of Vehicle Noise and Vibration*, vol. 2, no. 2, pp. 143–155, 2006, ISSN: 14791471. DOI: 10.1504/IJVNV.2006.011051.

[6] *ANSA version 15.2.x User's Guide*. BETA CAE Systems S.A., 2015.

[7] J. Brecht, *Untersuchungen zum Bremsenknarzen: Ein Beitrag zur Beschreibung von Schwingungen in Bremssystemen: Zugl.: Siegen, Univ., Diss., 2000*. Aachen: Shaker, 2000, ISBN: 3-8265-8161-X.

[8] B. Breuer and K. H. Bill, Eds., *Bremsenhandbuch: Grundlagen, Komponenten, Systeme, Fahrdynamik*, 5th ed. Wiesbaden: Springer Fachmedien, 2017, ISBN: 9783658154882.

[9] J. Chung and G. M. Hulbert, "A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-alpha method", *Journal of Applied Mechanics, Transactions ASME*, vol. 60, no. 2, pp. 371–375, 1993, ISSN: 00218936. DOI: 10.1115/1.2900803.

[10] M. Esgandari and O. Olatunbosun, "Implicit–explicit co-simulation of brake noise", *Finite Elements in Analysis and Design*, vol. 99, pp. 16–23, 2015, ISSN: 0168874X. DOI: 10.1016/j.finel.2015.01.011.

[11] Hartmut Hetzler, *Zur Stabilität von Systemen bewegter Kontinua mit Reibkontakten am Beispiel des Bremsenquietschens: Dissertation*. Karlsruhe: Universitätsverlag Karlsruhe, 2008, vol. 8, ISBN: 1614-3914.

[12] H. M. Hilber and T. J. R. Hughes, "Collocation, dissipation and [overshoot] for time integration schemes in structural dynamics", *Earthquake Engineering & Structural Dynamics*, vol. 6, no. 1, pp. 99–117, 1977. DOI: 10.1002/eqe.4290060111.

[13] H. M. Hilber, T. J. R. Hughes, and R. L. Taylor, "Improved numerical dissipation for time integration algorithms in structural dynamics", *Earthquake Engineering & Structural Dynamics*, vol. 5, no. 3, pp. 283–292, 1976. DOI: 10.1002/eqe.4290050306.

[14] S. Huemer-Kals, "Complex eigenvalue analysis of friction induced low-frequency vibrations in vehicle disc brake systems", Master Thesis, Graz University of Technology, 2018.

[15] V. A. Leontyev, "Direct time integration algorithm with controllable numerical dissipation for structural dynamics: Two-step lambda method", *Applied Numerical Mathematics*, vol. 60, no. 3, pp. 277–292, 2010, ISSN: 01689274. DOI: 10.1016/j.apnum.2009.12.005.

[16] S. K. Mahajan, Y. K. Hu, and K. Zhang, "Vehicle disc brake squeal simulations and experiences", *SAE Technical Papers*, 1999. DOI: 10.4271/1999-01-1738.

[17] N. M. Kinkaid, O. M. O'Reilly, and P. Papadopoulos, "Automotive disc brake squeal", *Journal of Sound and Vibration*, vol. 267, no. 1, pp. 105–166, 2003, ISSN: 0022460X. DOI: 10.1016/S0022-460X(02)01573-0.

[18] R. Nunes, A. Shivaswamy, and M. Könning, "A time domain approach towards analysing creep groan noise in automobile brakes", in *EuroBrake 2016*, FISITA, 2016, ISBN: 978-0-9572076-4-6.

[19] H. Ouyang, W. Nack, Y. Yuan, and F. Chen, "On automotive disc brake squeal part ii: Simulation and analysis", *SAE Technical Papers*, 2003. DOI: 10.4271/2003-01-0684.

[20] M. Pürscher, S. Huemer-Kals, and P. Fischer, "Experimental and simulative study of creep groan in terms of macpherson axle bushing elasticities", in *EuroBrake 2018*, vol. EB2018-SVM-011, FISITA, 2018.

[21] M. Pürscher, S. Huemer-Kals, and P. Fischer, "Experimental investigation of low-frequency vibration patterns in automotive disk brake systems: Utilization study for modal simulation methods", *SAE Technical Papers*, 2018. DOI: 10.4271/2018-01-1513.

[22] W. Rust, *Nichtlineare Finite-Elemente-Berechnungen: Kontakt, Kinematik, Material*, 3., überarbeitete und erweiterte Auflage. Wiesbaden: Springer Vieweg, 2016, ISBN: 9783658133771. DOI: 10.1007/978-3-658-13378-8.

[23] K. Uchiyama and Y. Shishido, "Study of creep groan simulation by implicit dynamic analysis method of fea (part2)", *SAE Technical Papers*, vol. 2014-September, no. September, 2014. DOI: 10.4271/2014-01-2516.

[24] K. Uchiyama and Y. Shishido, "Study of creep groan simulation by implicit dynamic analysis method of fea (part 3)", *SAE Technical Papers*, vol. 2015-September, no. September, 2015. DOI: 10.4271/2015-01-2689.

[25] M. Wagner, *Lineare und nichtlineare FEM: Eine Einführung mit Anwendungen in der Umformsimulation mit LS-DYNA*, ser. Lehrbuch. 2017, ISBN: 3658178655.

[26] P. Wriggers, *Computational contact mechanics: With 12 tables*, 2. ed. Berlin [u.a.]: Springer, 2006, ISBN: 3540326081.

A. Appendix

The following list contains every output parameter that was requested for the DPM and ACM. The identifier i denotes the respective components in Cartesian coordinates ($i = 1$ is equivalent to the x direction, $i = 2$ to y, and $i = 3$ to z).

- Node outputs (For a selection of potentially relevant nodes):
  - COORDi: Coordinates of the node in x, y, z
  - Ui: Displacement of the node in x, y, z
  - Vi: Velocity of the node in x, y, z
  - Ai: Acceleration of the node in x, y, z
- Energy outputs (for whole model):
  - ALLAE: Artificial strain energy associated with constraints
  - ALLCD: Energy dissipated by viscoelasticity
  - ALLCW: Work done by constraint penalties
  - ALLDMD: Energy dissipated by damage
  - ALLFD: Energy dissipated through frictional effects
  - ALLIE: Total strain energy (ALLIE = ALLSE + ALLPD + ALLCD + ALLAE + ALLDMD)
  - AlLKE: Kinetic energy
  - ALLPD: Energy dissipated by plastic deformation
  - ALLPW: Work done by contact penalties
  - ALLSE: Recoverable strain energy
  - ALLVD: Energy dissipated by viscous effects
  - ALLWK: External work
  - ETOTAL: Total energy (ETOTAL = ALLKE + ALLIE + ALLVD + ALLFD – ALLWK – ALLPW – ALLCW)
- Contact outputs (for inner and outer contact):
  - CFNi: Total force due to contact pressure in x, y, z

  - CFNM: Magnitude of total force due to contact pressure
  - CFSi: Total force due to frictional stress in x, y, z
  - CFSM: Magnitude of total force due to frictional stress
  - CSTATUS: Contact status for general contact analyses
  - XNi: Coordinates of the center of the total force due to contact pressure in x, y, z
  - XSi: Coordinates of the center of the total force due to frictional stress in x, y, z

ALLCW (constraint penalties) and ALLPW (contact penalties) have to be removed for implicit simulations and replaced by the following:

- ALLCCDW: Contact constraint discontinuity work
- ALLCCE: Sum of:
  - ALLCCEN: Contact constraint elastic energy in normal direction due to penalty constraint enforcement
  - ALLCCET: Contact constraint elastic energy in tangential direction due to friction penalty constraint enforcement
- ALLCCSD: Sum of:
  - ALLCCSDN: Contact constraint stabilization dissipation in normal direction
  - ALLCCSDT: Contact constraint stabilization dissipation in tangential direction