

User guide for CONTACT, Rolling and sliding contact with friction

Technical Report

TR09-03, version “v18.1.5”

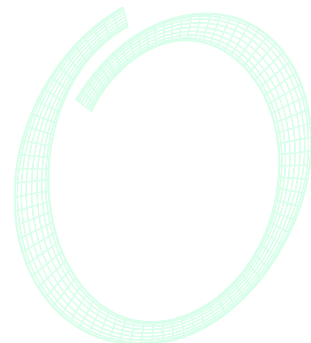
Date

May 1, 2018

Author(s)

Dr.ir. E.A.H. Vollebregt

© VORtech CMCC.



Contents

1	Introduction	6
1.1	Prerequisites	6
1.2	Purpose of CONTACT	7
1.3	History of the program	8
1.4	Free and premium versions	8
1.5	Structure of this report	9
1.6	Guidance to new users	9
2	Overview of the computational model	11
2.1	The role of contact in multi-body dynamics	11
2.2	Overall problem versus the contact problem	13
2.3	Contact-fixed and world-fixed coordinate systems	15
2.4	Using local coordinates	16
2.5	Formulation of the contact problem	17
2.5.1	Continuum mechanics	17
2.5.2	Surface quantities	18
2.5.3	The half-space approach	18
2.5.4	The contact conditions	19
2.6	Discretisation of the problem	20
2.7	Specification of a case	22
2.8	Solution processes	23
3	Operation of the program	26
3.1	Starting the program	26

3.1.1	Terminal-based, textual interface	26
3.1.2	Windows-based interface, the CONTACT GUI	28
3.2	Multiple cases, sequences	29
3.3	The control integers	30
3.3.1	Modules	30
3.3.2	Module 1 – wheel-rail contact	30
3.3.3	Module 3 – basic Hertzian and non-Hertzian contact	35
4	Module 1: wheel/rail contact	37
4.1	Wheel-rail coordinates and conventions	37
4.2	Track geometry	40
4.3	Wheelset geometry	41
4.4	Specifying the total forces	43
4.5	Potential contact area and discretisation	43
4.6	Global outputs for module 1	44
5	Module 3: in- and output quantities	46
5.1	Material parameters	46
5.1.1	Homogeneous elastic materials	46
5.1.2	Viscoelastic materials	47
5.1.3	Flexibilities of the simplified theory	48
5.1.4	Interfacial layer of contaminants	49
5.1.5	Numerically calculated influence coefficients	50
5.2	Friction description	51
5.3	Potential contact area and discretisation	54
5.3.1	Hertzian geometries	55
5.3.2	SDEC: simple double half-elliptical contact area	57
5.3.3	Direct specification of the potential contact area	58
5.4	Non-Hertzian geometry specification	59
5.5	Kinematic variables	60
5.5.1	Tangential quantities in case of a shift ($T = 1$)	61
5.5.2	Tangential quantities in case of rolling ($T = 2 - 3$)	61

5.6	Iteration constants	63
5.7	Global output quantities	65
5.7.1	Output in rolling problems ($T = 2 - 3$)	65
5.7.2	Output in case of shifts ($T = 1$)	66
5.8	Main solution arrays	66
5.9	Subsurface stresses	67
5.9.1	Control digits for the subsurface stress calculation	68
5.9.2	Input for the subsurface stress calculation	68
5.9.3	Output of subsurface stress calculation	70
6	Examples	73
6.1	The Cattaneo shift problem	73
6.2	The 2D Carter/Fromm problem	76
6.3	The 2D Bentall-Johnson test-case	77
6.4	Calculation of creep force curves	80
6.5	The Manchester wheel-rail benchmark	83
6.6	Conformal contact	86
6.7	The use of the FASTSIM algorithm	89
6.8	Instationary problems: from Cattaneo to Carter	90
6.9	The calculation of subsurface stresses	92
6.10	Dissimilar materials: Spence compression	94
6.11	Steady rolling of two viscoelastic cylinders	96
6.12	Transient rolling with velocity dependent friction	99
7	Matlab plot-programs	102
7.1	Prerequisites	102
7.2	Inspecting the surface stresses	104
7.2.1	Loading the results into Matlab	104
7.2.2	Plotting results for the entire (3D) contact area	105
7.2.3	Plotting results for 2D cross-sections	108
7.3	Inspecting subsurface stresses	109
7.3.1	Loading the results into Matlab	109

7.3.2	Plotting the subsurface stresses	110
8	The CONTACT library	112
8.1	Result elements and contact problems	112
8.2	Data units and sign conventions	113
8.3	Interface routines	114
8.3.1	Preparations	115
8.3.2	Use of parallel computing	116
8.3.3	Configuring basic contact problems (module 3)	117
8.3.4	Configuring wheel/rail contact problems (module 1)	119
8.3.5	Solving the contact problem	119
8.3.6	Global outputs for wheel/rail contact (module 1)	120
8.3.7	Global outputs per basic contact problem (modules 1 & 3)	120
8.3.8	Detailed outputs of the contact problem	121
8.3.9	Finalization	122
8.4	Calculation of creep force curves revisited	123
	Bibliography	125
A	Specification of in- and output-files	130
A.1	Files used by CONTACT	130
A.2	Specification of the input file <experim>.inp	130
A.2.1	Module 1 – wheel/rail contact	131
A.2.2	Module 3 – basic Hertzian/non-Hertzian contact	133
A.3	Subsurface-stress input in the file <experim>.inp	135
A.4	Specification of the file <experim>.<case>.mat	136
A.5	Specification of the file <experim>.<case>.subs	137

Chapter 1

Introduction

This report describes the usage of the computer program CONTACT for contact mechanical simulations. CONTACT is an advanced simulation program for the detailed study of three-dimensional frictional contact. For instance for investigating the wear of railway wheels and rails.

This program implements the famous theories for rolling contact by Prof. J.J. Kalker of Delft University of Technology. These were presented first in [12] and are described in full detail in [15], for an extended introduction see [17] or [46]. Notable extensions by Vollebregt concern the effects of third body layers [41] and falling friction [48] (together referred to as “Extended CONTACT”), speedup by dedicated iterative solvers, e.g. [40, 57], and the automated wheel/rail contact analysis.

1.1 Prerequisites

In order to understand this report and work with our program you should have a basic understanding of the deformation of solid objects. This involves the concepts of stress and strain from continuum mechanics. Further one should know about elastic, viscoelastic and plastic material behaviours, and of corresponding material parameters such as Young’s modulus and the yield strength. Introductory texts on these matters are provided on Wikipedia¹ and in text books such as [3] and [23].

In many cases applications of CONTACT incorporate the motion of the contacting bodies. Relevant background material on this is provided in text books on mechanics. A solid foundation on statics and dynamics of rigid bodies is provided for instance in [7]. The dynamics of deformable objects are surveyed among others in [4]. Reference texts on multi-body dynamics are provided by [28, 29]. The former of these is more general and pays attention to flexible bodies, the latter is targeted more on railway applications.

¹en.wikipedia.org/wiki/solid_mechanics

1.2 Purpose of CONTACT

Contact mechanics concerns the interaction of deformable bodies. A difficulty is that the contact area depends on the deformation of the bodies, whereas this deformation is function of the contact loading and hence the contact area. Therefore the determination of the actual contact area is an essential part of the problem. This is contrary to what is usually the case in finite element analysis, where the load and displacement are prescribed at different (fixed) parts of the boundary.

Different types of contact problems may be identified. For instance frictionless and frictional problems, stationary and instationary, and concentrated or diversified. The latter aspect concerns the size of the contact area with respect to the size of the contacting bodies. In concentrated contact problems, the contact area is relatively small compared to typical dimensions of the bodies, and the contacting bodies are “smooth edged” near the contact zone.

The program CONTACT is intended for concentrated contact problems. It solves the normal pressures (“frictionless”) and tangential (frictional) shear stresses (tractions) for stationary and instationary problems (shift, rolling). This is done for bodies of linearly elastic or viscoelastic materials. Both bodies are considered homogeneous –although possibly of different– materials. The geometry of the bodies is free (non-Hertzian). With respect to kinematic parameters (overall motion), total forces, approach and creepages may be prescribed.

After the surface loading (stresses) has been computed, the elastic field inside the bodies may be calculated. This gives the elastic displacements, displacement gradients and strains, and from those the stresses are deduced. In particular also the Von Mises stress is calculated, which is important in plasticity calculations.

CONTACT can be operated in different ways:

- detailed study of one or a few relevant cases, using detailed inputs and outputs, using the provided plotting routines,
- solving a large number of related cases, for instance for building up a table that can be used in another program,
- incorporating the calculating part of CONTACT as a subroutine in your own computing software. [The calculating part is provided as a library \(dll for Windows, so for Linux\) that can be interfaced from MATLAB, Fortran and C.](#)

The latter two options are relevant for wear calculations and for Vehicle System Dynamics (VSD) simulation codes. [Add-ons to GENSYNS, SIMPACK Rail \[50\] and Universal Mechanism² have been realized and are provided on a commercial basis.](#)

²www.gensys.se, www.simpack.com, www.universalmecanism.com

1.3 History of the program

The program CONTACT already has a long history. Version 1 was written in 1982, and followed the program DUVOROL [12, 34]. Since then there have been a number of versions. Several options were added, and some have been deleted too. Also the manual was revised over and over again.

In 1986–90, the direct method for steady state rolling was added to the program. This method is extensively described in [15]. The corresponding program (PC version) was called CONPC90. In 1992–4 the code has been restructured and modernized. New features of that time include fast solvers for the tangential contact problem [35] and the extension to viscoelastic contact problems (steady state rolling) [52, 53]. At that time the program was used for creating a table for the Hertzian creep-force law [16] and for the study of rough elastic contacts [18]. Since then the program has been distributed and supported for a long time without significant new extensions being made. This version is designated as CON93 or CONTACT’93 [47].

Since 2008 there has been renewed interest in this software and new developments have been made. The code has been modernized once again, simplifying its usage and improving its extendability. Also the robustness, accuracy and speed of operation have been improved [36]. The “Panagiotopoulos process” was found completely reliable, whereupon the slower KOMBI algorithm was removed. A first public version was published in 2009. Estimating that this could count as the ninth incarnation, this version was designated CONTACT v9.1 [37].

The range of applicability of CONTACT was extended since 2009 by incorporation of velocity dependent friction laws [48], modeling the effects of roughness and contamination in the contact interface (“third body layer”) [41], by making extensions for solving conformal contact problems [45, 49], and via new enhanced solvers on the basis of Conjugate Gradients and FFTs [40, 57].

Research is being carried out that should result in further enhancement of the program’s speed versus accuracy trade-off, and in its range of applicability. When such results are achieved, they will be included in future versions of this document.

1.4 Free and premium versions

The basic version of CONTACT is provided for free at www.kalkersoftware.org. Extended features are provided commercially on the basis of a licensing mechanism:

- The automated contact geometry analysis for given wheel/rail profiles (module 1), and
- the CONTACT library for MATLAB, Fortran and C.

The extended features are marked [blue](#) in this user guide.

1.5 Structure of this report

The remainder of this report is structured as follows.

- Chapter 2 presents an overview of the computational model that is used as far as is needed for working with the program. For further details and background information the reader is referred to [15, 17] and [46].
- The operation of the program is described in Chapter 3. This concerns the interactive and batch usage, terminology with respect to “*modules*” and “*cases*” (e.g. time steps), and the *control integers* that steer the execution of the program.
- Chapter 4 describes the input and output quantities for wheel/rail contact analysis.
- Chapter 5 gives a detailed specification of the input and output quantities of the program, and thereby also presents the input options that are available.
- The usage of the stand-alone program is demonstrated further through the examples in Chapter 6. Chapter 7 describes the Matlab scripts for visualization of the results.
- The use of the CONTACT library is documented in Chapter 8.
- Finally Appendix A specifies the files that are used.

1.6 Guidance to new users

It is acknowledged that CONTACT is not the easiest program to get acquainted with.

- The main problem is that there is little introductory material on contact mechanics as a whole. The books by Johnson [9] and Kalker [15] are excellent reference works, but present quite a challenge to get familiar with the material. A more gentle introduction to contact mechanical phenomena is provided by Popov [26].
- Further, there is a large diversity of situations to which the program can be applied. This is impeding the definition of a simple work-flow or the construction of simple plotting facilities.
- Finally, there’s the old-fashioned structure of input-file, with the many “control digits” and switches that are involved.

In our opinion the usage of the program is not so hard *if* one knows about contact mechanics terminology. It’s easy to get familiar with the input-file itself. For this the following steps are advised:

1. Quickly go over the contents of this user guide, particularly Chapters 3, 4 and 5.
2. Run the program for the examples that are provided and go through their description in Chapter 6 of this guide.
3. Put guide cards at Sections 3.3 and A.2 of this guide; this reference information is consulted frequently.

As stated above it is more difficult to get a good grasp of contact mechanics principles and terminology. This is why we set up a course that is advertised on www.kalkersoftware.org. With this course you also provide feedback to us and support the further development.

Chapter 2

Overview of the computational model

The program CONTACT is meant for:

- 3D homogeneous bodies of (linearly) elastic and viscoelastic materials, that may be different for the two contacting bodies,
- with concentrated contact, i.e. where the resulting geometries are essentially flat in and near the contact zone, but not necessarily Hertzian,
- with dry (Coulomb) friction or boundary lubricated situations (third body layer, falling friction, friction memory effects),
- solving shifts as well as rolling, transient as well as steady state problems, with creepages and/or total forces prescribed,
- solving for the surface tractions first, but capable of computing the elastic field in the interiors of the bodies as well.

2.1 The role of contact in multi-body dynamics

The main purpose of CONTACT is to assess the total forces between contacting bodies in different circumstances. These forces are needed to compute the dynamic behavior of mechanical systems, of which rail vehicle dynamics is the most important exponent with respect to steel on steel contacts. For instance when a train moves through a curve, the contact point on the outer rail shifts towards the rail gauge corner where the surface is inclined with respect to the horizontal plane. This changes the direction of the contact force, which facilitates steering of the vehicle through the curve.

It is important to realize that the physical phenomena at play differ greatly between normal and tangent directions, which introduces strong *anisotropy* in the contact problem (Figure

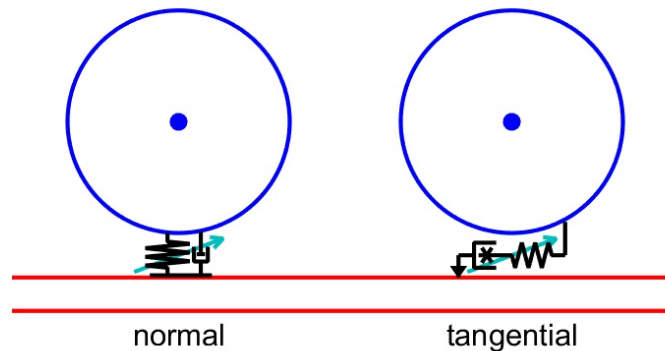


Figure 2.1: Schematization of the wheel-rail rolling contact forces: acting mainly as a variable spring in normal direction, Hertzian or non-Hertzian, and like a variable spring and dashpot tangentially that breaks loose if a large a force is required.

2.1). In the normal direction, one body is pressed onto the other. It tries to push through the other, and this is resisted strongly by the cohesive forces inside the other bodies' material. The force by which the bodies are pressed together causes (elastic) deformation to occur. The bodies seem to approach each other, by which it may seem as if penetration of the bodies occurs. If the force is released then the elastic deformation diminishes and the centers of the bodies are separating again. Consequently it is said that normal contact behaves like a spring. The stiffness of this spring is variable, dependent not only on the material and geometry but also on the contact load. There may be material damping as well, which would give a small dashpot *parallel* to the spring. Whether this damping is important or not depends on the materials that are used and the frequencies that are of concern.

Tangential contact is different in the sense that it allows for large displacements to occur between the two surfaces, with friction being the main physical phenomenon. If a small tangential force is applied to one body then it may deform slightly such that some tangential displacement seems to occur. This is like an elastic spring, i.e. the displacement is undone when the force is released. If the applied force is larger, then there may also be some micro-slip between the surfaces in the contact interface. This leads to an energy loss, i.e. this displacement is not undone upon releasing the force. Therefore this corresponds more to a dashpot as shown in Figure 2.1, *in series* with the tangential elasticity. Finally if the external force is large, compared to the normal load that presses the bodies together, then gross sliding occurs.

Tangential contact behaves differently between sliding and rolling. In sliding circumstances, the same material particles remain in contact during the overall motion. In this case, only so much elastic deformation can build up, after which no more displacement can occur (low/medium force) or gross sliding must set in (large force). During rolling, fresh material enters the contact all the time, allowing for some overall displacement to go on continuously. This introduces the so-called *creep phenomenon*, an apparent sliding velocity between the two surfaces. This creepage is resisted by a tangential force.

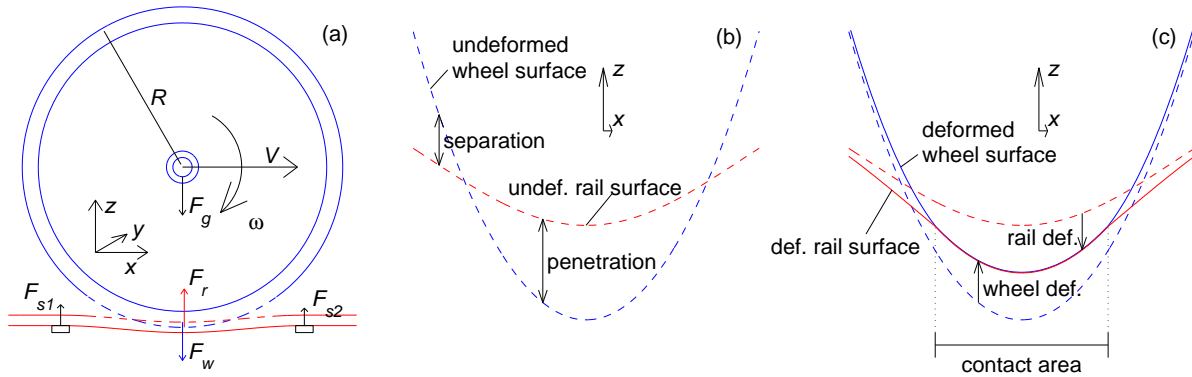


Figure 2.2: *Illustration of the wheel-rail contact problem [48]. (a): overall geometry. (b): wheel and rail geometries in so-called undeformed states. (c): wheel and rail geometries in the deformed state and corresponding deformations and contact area. Note: graphs (b) and (c) are stretched vertically.*

In steady state rolling and at low creepage, the force and percentual creepage are related according to the linear theory of Kalker [10, 15]. At larger creepages the creep force saturates at the friction maximum, upon which full sliding occurs, leading to the well-known creep versus creep force curve (see for instance Figure 6.4). Note that in steady state rolling, the tangential contact spring is held at constant elongation. Hence the creep-force curve describes the variable dashpot of Figure 2.1. The tangential contact spring comes into play during non-steady scenarios, and may then lead to tangential *rocking* to occur [42, 43].

2.2 Overall problem versus the contact problem

The geometry of the wheel-rail contact problem is introduced in Figure 2.2. In graph (a), the overall geometry is shown: the rolling wheel, with radius R , forward velocity V and angular velocity ω , positioned somewhere on a rail. The rail is bent due to forces exerted on it at the sleepers (F_{s1}, F_{s2}) and at the wheel-rail interface (F_w). In this picture the wheel and rail are considered rigid. The wheel is moved down such that penetration with the rail occurs. This is shown in detail in Figure 2.2 (b), stretched vertically. The distance by which the wheel is shifted down equals the maximum penetration and is called the *approach*, δ_n . The configuration shown in this figure is referred to as the “undeformed state”.

It is the purpose of multibody simulations to analyse the dynamic behaviour of such a system: to find the position and speed of the wheel and the bending of the rail as a result of all forces in the system, i.e. the forces $F_r(t)$ and $F_w(t)$ exerted on the wheel and the rail as function of time. These forces result from the elastic deformations in the contact patch, and can be found as a function of the wheel and rail geometries and the approach $\delta_n(t)$. This is illustrated in Figure 2.2 (b). At each time instance one obtains from the multibody simulation the positions and so-called “undeformed” shapes of the contacting bodies, and is asked for the resulting

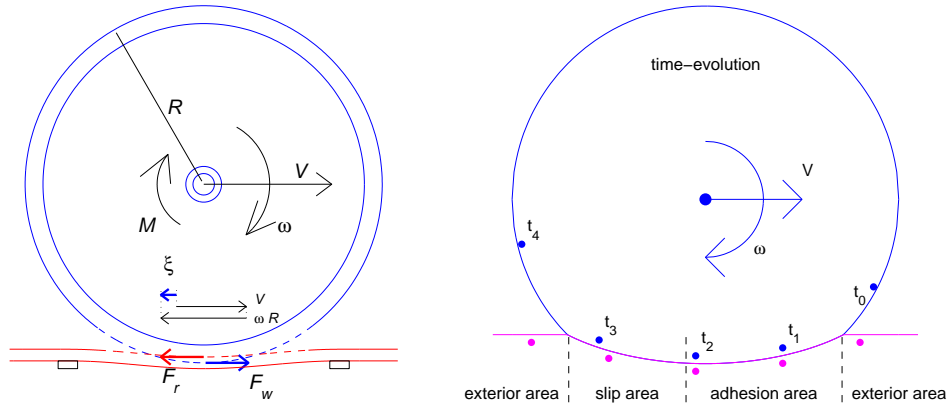


Figure 2.3: *Left: illustration of the creepage ξ , i.e. the relative velocity difference $(V + \omega R)/V$. Right: particles that move through the contact area (from right to left) tend to adhere to the opposite surface first, are strained more and more until local sliding (micro-slip) sets in.*

reaction forces in the contact patch.

The elastic deformations and resulting contact patch are illustrated in Figure 2.2 (c). The normal pressure between the surfaces is compressive and vanishes outside the contact area. Its integral over the contact area is the total force $F_w = -F_r$ that is sought for. It pushes the wheel surface upwards and the rail surface down. If the wheel and rail have identical elastic parameters then their local deformations are equal and opposite functions of position x and time t , and the maximum deformation is $\pm\delta_n/2$. Due to the strength of the material the deformations also extend outside the contact area, even though there is no pressure working there.

A typical size for the contact between steel wheel and rails is $10 \times 10 \text{ mm}$, and the corresponding approach is 0.01 mm . The elastic deformations decrease with $1/r$, with r the distance to the contact area, and the stresses and strains decrease in proportion to $1/r^2$. Therefore the stresses and strains are negligible at distances of a few centimeters outside the contact patch. This is the reason why the overall multi-body simulation and local contact problems may be decoupled [15].

If the profiles of rail and wheel are smooth, quadratic surfaces then the normal pressure can be solved via Hertz' theory. In case of other shapes (e.g. varying curvatures, including roughness) CONTACT's non-Hertzian capabilities come into play. Next the tangential forces have to be obtained as well. These cause *creepage* between wheel and rail, as illustrated in Figure 2.3, left. When a tractive force (torque M) is applied on a railway wheel, a small difference arises between the overall forward velocity V and circumferential velocity ωR (note: $\omega < 0$ for a wheel moving in positive x -direction). The relative difference is the (longitudinal) creepage $\xi = (V + \omega R)/V$. It is restricted by the friction force F_w acting on the wheel.

This overall creeping motion of the wheel tries to drag the wheel surface particles over the rail, i.e. it can be seen as an average amount of slip. This relative movement is resisted by

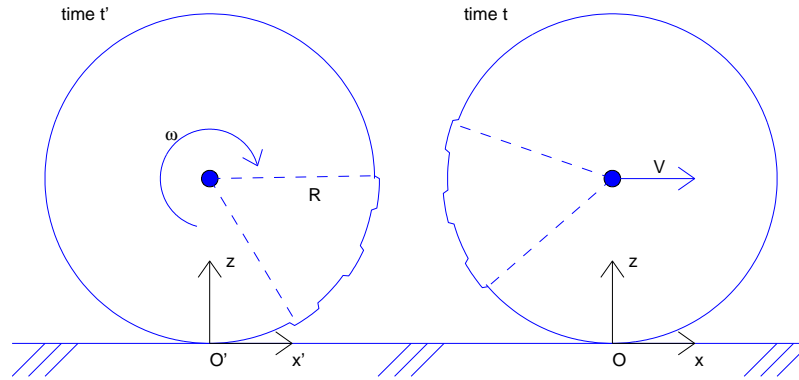


Figure 2.4: *Schematic example illustrating non-stationary rolling of a wheel with some irregularities on a rail.*

frictional shear stresses in the contact interface. If the circumferential speed of wheel particles is larger than the forward velocity ($\xi < 0$, $\omega R < -V$), the net tangential force on the wheel is pointing forward ($F_w > 0$) and accelerates the train. On the other hand, if the train is braking ($F_w < 0$), then the circumferential velocity is lower than the forward velocity ($\omega R > -V$, $\xi > 0$). Note that the situation is more complicated in reality, where lateral creepage η between wheel and rail may occur. Also, rotation of the wheel about the normal direction occurs (a cone rolling over a plane makes a circular trajectory), which is described by spin creepage ϕ . These combined creepages lead to combined forces F_x, F_y and the spin moment M_z .

Figure 2.3 (right) shows what happens to surface particles when they traverse through the contact area. They are free of stress when entering the contact area at the right side, the leading edge of the contact area. There they adhere to a particle of the opposing surface. Next they are strained by the overall motion difference between the two bodies. This introduces shear stresses, which increase until the local traction bound is exceeded and local slip sets in. This process is in different stages for different parts of the contact area.

If the overall motion of the bodies is constant, then an overall steady state may be attained. Here the state of each surface particle is varying in time, but the overall distribution can be constant. This is formalised by using a coordinate system that is moving along with the contact patch.

2.3 Contact-fixed and world-fixed coordinate systems

A typical geometrical configuration is presented in Figure 2.4. This example concerns a wheel with some irregularities at two different instances in time. In the time-period shown in the picture the wheel rotates over an angle of 140° . At the same time it moves forward in positive x -direction with velocity V . Under rolling conditions we have $V \approx -\omega R$, i.e. the creepage is

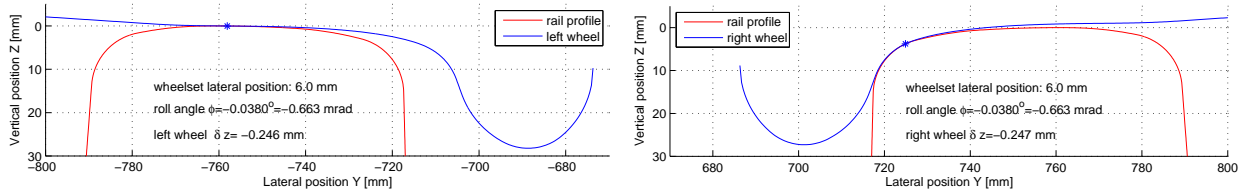


Figure 2.5: *Initial contact points for left and right wheels at wheelset lateral displacement $\delta y = 6 \text{ mm}$, yaw $\psi = 0^\circ$ (Manchester contact benchmark, see Section 6.5).*

small, at most a few percent. On the other hand when $V + \omega R$ is rather large compared to V , we have sliding or “rolling with sliding” circumstances.

The picture shows different coordinate systems for the different time instances. At time t' the coordinate system $O'x'y'z'$ is used, at time t the coordinate system is $Oxyz$. This is the *contact-fixed* coordinate system. In this example the y - and z -directions stay the same, but generally these may vary over time too. Particularly the z -axis is required to be normal to the two bodies’ surfaces. An important aspect of this coordinate system is that particles of the bodies’ surfaces move through the contact area with velocity $-V$. On the other hand, the bodies themselves appear to be fixed in space.

An alternative approach is used in sliding or rolling with sliding circumstances. In such a case there need not be a rolling velocity V . Then a *world-fixed* coordinate system is used. In terms of Figure 2.4 this consists of using coordinate system $O'x'y'z'$ for time t' as well as for time t . In this description the axle moves forward with velocity V , whereas particles in the contact area more or less stay at the same coordinate.

When modeling two rollers in a machine the situation is a little different. In that case the contact-fixed coordinate system is also a world-fixed one. The coordinate system to be used for sliding problems should then be a moving one, defined such that the particles in contact are almost stationary with respect to the coordinates used. This then defines “*material-fixed*” coordinates. A complete description of particles, coordinates and velocities is given in [15].

2.4 Using local coordinates

The consequence of anisotropy (Figure 2.1) is that contact phenomena must be studied in suitably defined coordinates, with “ n -direction” aligned with the normal to the contact plane. This is a complicating factor for wheel-rail contact analysis, due to the contact angle δ that arises when the contact moves from the wheel tread towards the flange, as illustrated in Figure 2.5.

The program CONTACT is based on a boundary element method (BEM), suited for concentrated contacts where no sharp geometric features occur near the contact zone. Originally this was based on so-called “planar contact” using the half-space approach. This relies on a

so-called “initial contact point”, where the normal direction and tangent plane are used to define local coordinates.¹ The steps for solving a contact problem are then:

1. Determine a suitable “initial contact point” or “contact reference”;
2. Determine the normal direction common to the contacting surfaces, and define the tangent contact plane;
3. Determine the contact geometry (esp. the undeformed distance function) and kinematics (“rigid slip velocity”) relative to the contact plane;
4. Solve the normal and tangential contact problems;
5. Convert results to the global coordinates.

All the steps are automated for wheel-rail contact analysis as discussed in Chapter 4. For other applications, the user takes care of the necessary conversions and preparations (steps 1–3 and 5), and uses CONTACT for step 4, the pure contact problem.

2.5 Formulation of the contact problem

The contact problem consists of determining the various aspects of the deformed state: the contact area, the distribution of surface tractions (pressures and frictional shear stresses), the deformations, and the stresses inside the materials. These are the result of a complex interaction between:

1. the overall motion (approach in normal direction and tangential creepage),
2. the elastic deformation of the two bodies (local motion), and
3. the friction processes (interaction between surface particles).

2.5.1 Continuum mechanics

When the bodies are brought into contact, stresses $\boldsymbol{\sigma}^{(a)}(\mathbf{x}, t)$, strains $\boldsymbol{\epsilon}^{(a)}(\mathbf{x}, t)$ and displacements $\mathbf{u}^{(a)}(\mathbf{x}, t)$ arise in the bodies and at their surfaces. These are related by Newton’s second law and the material behaviour. For instance, assuming linear elasticity, these relations read

$$\rho^{(a)} \ddot{\mathbf{u}}^{(a)} = \nabla \cdot \boldsymbol{\sigma}^{(a)} + \mathbf{f}^{(a)}, \quad \boldsymbol{\sigma}^{(a)} = \mathbf{C}^{(a)} : \boldsymbol{\epsilon}^{(a)}, \quad \boldsymbol{\epsilon}^{(a)} = \frac{1}{2} [\nabla \mathbf{u}^{(a)} + (\nabla \mathbf{u}^{(a)})^T]. \quad (2.1)$$

¹In recent years CONTACT has been extended towards conformal contact using numerical influence coefficients, see Section 5.1.5.

Here $a = 1, 2$ is the body number, with $a = 1$ for the upper body ($z > 0$). $\mathbf{f}^{(a)}$ represents body forces, $\rho^{(a)}$ is the mass density, and $\mathbf{C}^{(a)}$ is a fourth-order stiffness tensor. The first equality of (2.1) is the equation of motion (Newton), the second equality describes the material behaviour according to Hooke’s law, and the latter is the strain-displacement relation. These equalities must hold everywhere in the bodies’ interiors $\mathbf{x} \in \Omega^{(a)}$, where suitable boundary conditions are required on $\partial\Omega^{(a)}$.

2.5.2 Surface quantities

In formulating the contact problem, we are particularly interested in the surface quantities: the displacement $\mathbf{u}^{(a)}(\mathbf{x})$ of the surface particles \mathbf{x} of bodies $a = 1, 2$ ($z \approx 0$) and the surface tractions

$$\mathbf{p}^{(a)}(\mathbf{x}) = \boldsymbol{\sigma}(\mathbf{x}) \cdot \mathbf{n}^{(a)}(\mathbf{x}), \quad (2.2)$$

with $\mathbf{n}^{(a)}(\mathbf{x})$ the outer normal on body a at \mathbf{x} . Now, since $\mathbf{p}^{(2)}(\mathbf{x}) = -\mathbf{p}^{(1)}(\mathbf{x})$ for all surface positions \mathbf{x} where the bodies are in contact, we may eliminate $\mathbf{p}^{(2)}$ and consider a single variable $\mathbf{p}(\mathbf{x}) = \mathbf{p}^{(1)}(\mathbf{x})$ for the contact area. Furthermore, the displacements enter the contact problem mainly through their differences. Therefore, we introduce the so-called *displacement difference* $\mathbf{u}(\mathbf{x})$ at position \mathbf{x} :

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}^{(1)}(\mathbf{x}) - \mathbf{u}^{(2)}(\mathbf{x}). \quad (2.3)$$

The quantities introduced above are tensors ($\boldsymbol{\sigma}, \boldsymbol{\epsilon}$) and vectors ($\mathbf{x}, \mathbf{u}, \mathbf{p}$) in three-dimensional space. We assume that the contact area C is in the plane Oxy ,² such that $\mathbf{n}^{(a)} = [0, 0, (-1)^a]^\text{T}$, and concentrate on the surface points $\mathbf{x} = [x, y, 0]^\text{T}$. In the following we often deal with the normal and tangential vector components separately. The normal coordinate direction is indicated with subscript n and the tangential directions by t . The normal pressure (scalar function) and tangential tractions (2-vectors) are denoted by

$$p_n(x, y) = \mathbf{p}(\mathbf{x})^\text{T} \mathbf{n}, \quad \mathbf{p}_t(\mathbf{x}) = [p_x(x, y), p_y(x, y)]^\text{T}. \quad (2.4)$$

The normal and tangential components of the displacement difference are indicated similarly by u_n and \mathbf{u}_t .

2.5.3 The half-space approach

The program CONTACT is based on an *influence function method* or *boundary element method*, which in our case is also called the *half-space approach*.² The elasticity equations (2.1) for the interiors of the two contacting bodies are converted to equations for their bounding surfaces. The main resulting equation is the relation between surface tractions \mathbf{p} and

²For simplicity we do not consider conformal contact situations here. For information on this consult the paper [49] that describes the extensions with respect to the half-space approach.

displacements \mathbf{u} :

the tractions-displacement relation:

$$\mathbf{u}(\mathbf{x}, t) = \int_{\mathbf{x}' \in C} \mathbf{A}(\mathbf{x}, \mathbf{x}') \mathbf{p}(\mathbf{x}', t) dC \quad (2.5)$$

This relation depends on the constitutive equations that describe the material behaviour, as well as on the geometries of the bodies.

In CONTACT the following assumptions are made [15]:

- the bodies are formed of linearly elastic materials, and are homogeneous,
- the contact area is essentially flat and small with respect to typical dimensions of the bodies' geometries,
- no sharp variations exist in the geometries of the bodies,
- inertial effects ($\rho \ddot{\mathbf{u}}$) are small with respect to the contact stresses ($\nabla \cdot \boldsymbol{\sigma}$) and may be ignored.

These assumptions allow for using the so-called *half-space approach*. The actual response of the bodies to the surface loading is approximated by that of the elastic half-space, which was presented in analytical form by Boussinesq and Cerruti. With the half-space solution the contact problem is brought into surface-mechanical form. In this form no reference is made to the stresses $\boldsymbol{\sigma}$, strains $\boldsymbol{\epsilon}$, and displacements \mathbf{u} in the bodies' interiors.

2.5.4 The contact conditions

Two important quantities of the contact problem are:

normal problem :

$$\text{deformed distance} \quad e := h + u_n \quad (2.6)$$

tangential problem :

$$\text{relative slip velocity} \quad \mathbf{s}_t := \mathbf{w}_t + \dot{\mathbf{u}}_t/V \quad (2.7)$$

The function $h(x, y)$ describes the profiles of the two bodies as well as the approach denoted by δ_n . The relative rigid slip \mathbf{w}_t describes the velocity by which the surfaces move with respect to each other in the undeformed state. Together with the time derivative of the displacements ($\dot{\mathbf{u}}$) this yields the slip velocity \mathbf{s}_t of two opposing particles of the two bodies with respect to each other.

With these quantities the contact problem is to determine the contact region C , its subdivision into adhesion and slip areas H and S , and the tractions p_n , \mathbf{p}_t such that the following contact

conditions are satisfied:

normal problem :

$$\text{in exterior } E : e > 0, p_n = 0 \quad (2.8)$$

$$\text{in contact } C = H \cup S : e = 0, p_n \geq 0 \quad (2.9)$$

tangential problem :

$$\text{in exterior } E : \mathbf{s}_t \text{ free, } \mathbf{p}_t = \mathbf{0}, \quad (2.10)$$

$$\text{in adhesion } H : \|\mathbf{s}_t\| = 0, \|\mathbf{p}_t\| \leq g, \quad (2.11)$$

$$\text{in slip } S : \|\mathbf{s}_t\| > 0, \mathbf{p}_t = -g \mathbf{s}_t / \|\mathbf{s}_t\|, \quad (2.12)$$

$$\text{Coulomb friction : } g(\mathbf{x}, t) = \mu p_n(\mathbf{x}, t) . \quad (2.13)$$

These contact conditions state that:

- The bodies cannot interpenetrate in the deformed state: the distance between their surfaces is non-negative;
- The normal pressure is compressive, attraction between the surfaces is ignored;
- The exterior area is free of traction (the effect of auxiliary forces may be computed beforehand and incorporated in the geometry, i.e. h);
- The frictional shear stress cannot be larger than a space-varying maximum, the traction bound g ;
- No slip occurs where the tangential traction falls below the traction bound;
- If there is slip then the tractions are on the traction bound and opposite to the slip direction.

These contact conditions can be seen as a complex set of boundary conditions for the solid mechanics problems (2.1) for bodies $a = 1, 2$. The deformations in the two bodies' interiors cannot be solved independently but are connected, via the overall motion $(\delta_n, h, \mathbf{w}_t)$, the deformation $(\mathbf{p}, \mathbf{u}$, equation (2.5)) and the precise conditions at the interface (equations (2.6)–(2.13)).

2.6 Discretisation of the problem

In CONTACT a calculation starts by defining a potential contact area that encompasses the true contact area. The potential contact area is discretised into $N = mx \cdot my$ rectangular elements of size $\delta x \cdot \delta y$ (see Figure 5.6). The surface tractions are approximated by piecewise constant functions per element. This leads to

$$u_i(\mathbf{x}_I) = \sum_{j \in \{n, x, y\}} \sum_{J=1}^N A_{IiJj}(\mathbf{x}_I, \mathbf{x}_J) p_j(\mathbf{x}_J), \quad \text{for } I \in \{1 \cdots N\}, i \in \{n, x, y\}. \quad (2.14)$$

\mathbf{x}, \mathbf{y}	locations in 3D space, typically of the form $[x, y, 0]^T$ for points in the contact area C . Also used to identify particles of the bodies;
C, H, S	contact area, adhesion area, slip area;
$\mathbf{u}(\mathbf{x})$	displacement difference $\mathbf{u}^{(1)}(\mathbf{x}) - \mathbf{u}^{(2)}(\mathbf{x})$, with body 1 the upper body with $z > 0$;
$\mathbf{u}^{(a)}(\mathbf{x})$	displacement vector of the particle of body a (1, 2) that is at location \mathbf{x} in the undeformed state;
$u_i^{(a)}(\mathbf{x})$	i^{th} (scalar) component of the displacement vector $\mathbf{u}^{(a)}(\mathbf{x})$;
i, j	coordinate directions 1, 2, 3, with the first of these being normal to the contact area (n -, z -direction), the second the longitudinal (rolling) direction (x), and the third perpendicular to both (y -direction);
α	tangential coordinate direction: 2, 3 or x, y ;
$\mathbf{p}(\mathbf{x})$	surface traction (vector) acting on body 1 at position \mathbf{x} ;
$\mathbf{A}(\mathbf{x}, \mathbf{x}')$	3×3 matrix of influence functions;
$A_{ij}(\mathbf{x}, \mathbf{x}')$	influence function, describing the displacement difference in i -direction at \mathbf{x} arising due to a unit load in j -direction at \mathbf{x}' ;
$h(\mathbf{x})$	distance between the surfaces of the two bodies in normal direction in the undeformed state at location \mathbf{x} (positive: gap, negative: interpenetration);
$e(\mathbf{x})$	distance between the two bodies in the deformed state;
$\dot{\mathbf{u}}(\mathbf{x})$	material (particle fixed) time derivative of the displacement difference $\mathbf{u}(\mathbf{x})$;
\mathbf{s}_t	relative (tangential) slip velocity of two opposing particles of the bodies with respect to each other;
g	traction bound, maximum tangential traction (magnitude) that can be sustained with the surfaces adhering together;
$\delta x, \delta y$	the sizes in x - and y -directions of the rectangular discretisation elements;
\mathbf{x}_I	location of the center of discretisation element I ;
I, J	1D (i) or 2D ((ix, iy)) numbers of the discretisation elements (see equation (5.15));
\mathbf{u}_I	shorthand notation for $\mathbf{u}(\mathbf{x}_I)$, displacement diff. at the center of element I ;
u_{In}, \mathbf{u}_{It}	shorthand notations for $u_n(\mathbf{x}_I)$ and $\mathbf{u}_t(\mathbf{x}_I)$, (scalar) normal and (2-vector) tangential displacement differences at center of element I ;
$u_{I\alpha}$	shorthand notation for $u_\alpha(\mathbf{x}_I)$, (scalar) displacement difference in direction α (x or y) at center of element I ;
$t, t', \delta t$	time instances, time step;
V	in rolling problems: the rolling speed, in shifts $V = 1$;
δq	geometrical time step size: distance traversed per time step;

Table 2.1: Overview of notations and conventions used in this document.

Here \mathbf{x}_I and \mathbf{x}_J stand for the coordinates of rectangular elements I and J . A_{IiJj} stands for the influence coefficients. These are obtained by integrating (2.5) over a single element J with respect to an observation point at \mathbf{x}_I , which can be done analytically [15]. Due to the choice for rectangular elements, the influence coefficients A_{IiJj} are identical for all pairs I, J for which the relative positions are the same.

The slip \mathbf{s}_t at the surface of the contacting bodies involves a time-derivative. It is discretised using a “previous time instance” t' , with $\delta t = t - t'$. A related quantity is the traversed distance per time step $\delta q = V \cdot \delta t$, with V the rolling speed. δq is also called the “time step” for brevity. The displacements at the previous time instance are denoted by \mathbf{u}' .

The shift (distance) \mathbf{S}_t is the slip aggregated over a time step $\mathbf{s}_t \cdot \delta q = V \mathbf{s}_t \cdot \delta t$. It is discretised using a particle fixed, Lagrangian approach. In this approach the shift is expressed as the sum of rigid shift plus deformation shift, where the deformation shift is the change in deformation of two contacting particles over a time step from t' to t .

$$\mathbf{S}_{It} = \mathbf{W}_{It} + \mathbf{u}_{It} - \mathbf{u}'_{It}, \quad \text{for } I \in \{1 \cdots N\}. \quad (2.15)$$

Here \mathbf{W}_{It} is the rigid shift of the bodies, in rolling problems the creepage integrated over a time step δt . \mathbf{u}_{It} is the current tangential deformation difference $\mathbf{u}_{It}^{(1)} - \mathbf{u}_{It}^{(2)}$ of two contacting particles at element I . And \mathbf{u}'_{It} is the deformation difference of the same particles one time step earlier, at the position where they resided at time t' .

2.7 Specification of a case

With the formulations chosen as described above, a contact problem is specified completely by the following quantities:

- a coordinate system with positive z -direction normal to the contact plane and pointing into the upper body, body 1. In rolling problems this is a contact-fixed coordinate system, in sliding problems (shifts) material-fixed coordinates are used (Section 2.3);
- the potential contact region, a rectangular area in the plane $z = 0$ that encompasses the true contact region, and its discretisation step sizes $\delta x, \delta y$ or number of elements m_x, m_y (Figure 5.6);
- in rolling problems: the rolling velocity V , direction χ (“chi”) and “time step size” $\delta q = V \delta t$. Note: recent modifications of the program more or less require that $\chi = 0^\circ$ or 180° , i.e. rolling takes place in positive or negative x -direction;
- the material parameters $G^{(a)}, \nu^{(a)}$ for elastic materials for the two bodies $a = 1, 2$ and additional parameters for viscoelastic materials that are included in the influence function $\mathbf{A}(\mathbf{x}, \mathbf{y})$;

- the undeformed distance $h(\mathbf{x})$ between the surfaces, i.e. their separation in the direction normal to the contact area, up to a constant value, the approach δ_n ;
- either the approach δ_n or the total normal force F_n ;
- the rigid slip \mathbf{w}_t : the relative movement of opposing particles in tangential directions x, y in the undeformed state. These are characterized by creepages ξ (**ksi**) and η (**eta**) in x - and y -directions respectively, and the spin creepage ϕ (**phi**) with respect to the z -axis of the local coordinate system. Alternatively to the creepages total forces may be prescribed. An extra term to this rigid slip may be provided as well;
- configuration parameters for the solution processes, such as the required solution accuracy, number of iterations required etc.

These quantities are easily recognized in the variables that are specified in the user input, which are described in detail in Chapter 5.

2.8 Solution processes

The discretised problem is solved using five nested iteration processes.

1. The outer iteration.
 - (a) If the two bodies have identical elastic properties then they are called “*quasi-identical*”. This results in decoupling of the normal and tangential problems. The normal problem can be solved first and with that (and resulting traction bound g) the tangential problem can be solved. This is a one-step outer iteration procedure that is called “Johnson’s process”.
 - (b) On the other hand when the two bodies have different elastic properties, the tangential tractions affect normal displacement differences and vice versa. In such a case an outer iteration process called “Panagiotopoulos’ process” is used. This consists of solving normal and tangential problems alternately until the update between consecutive iterations is smaller than a tolerance.
2. The slip velocity iteration.
 - (a) The traction bound g in (2.11) and (2.12) can be made dependent on the slip velocity $\|\mathbf{s}_t\|$. In that case an additional iteration loop is used. In each iteration, a contact problem with a fixed traction bound g is solved. This yields a new estimate for the slip velocity \mathbf{s}_t , which is then used to compute a new traction bound g and iterate.

3. The active set algorithms NORM and TANG [15]. Note: although these algorithms are still maintained in the code, they are not active anymore. Today the element divisions are determined by the iterative solvers described below at item 5.

- (a) Kalker’s algorithm NORM was used for solving the normal contact problem. It consisted of guessing which elements should be inside the contact area C (“active” elements, active constraints $e = 0$), solving the corresponding equations, checking where the guess was appropriate or not, and iterating until the correct element division is found.
- (b) Kalker’s algorithm TANG was used for solving the tangential contact problem. In this case the active set algorithm determined the subdivision of the contact area C into adhesion and slip areas H, S . Each iteration required solution of $2n$ linear and nonlinear equations. In Kalker’s original approach the solution was done by Newton linearization in combination with Gaussian elimination [15].

4. The Newton-Raphson loop for the tangential forces.

- (a) If the total tangential forces F_x, F_y are prescribed, then the creepages ξ, η are to be adjusted. This cannot be done by the iterative solvers described below, which require that the rigid slip distribution is specified beforehand. This is solved by a small iterative procedure that estimates the derivatives $\partial F_x / \partial \xi - \partial F_y / \partial \eta$ and uses these to update the values of the creepages. This is called a Newton-Raphson process.

The 2×2 system for the tangential forces is complicated due to its inherent nonlinearities. Therefore the robustness of the approach is not 100%, particularly when both creepages are unknown and when the tangential forces are close to the maximum where full sliding occurs.

5. The iterative solvers NormCG, TangCG, ConvexGS, and SteadyGS [40, 57, 35, 38].

- (a) Solving the normal problem requires finding normal pressures p_{In} in each element satisfying equations like

$$e_I = h_I^* + \sum_{J \in C} A_{InJn} p_{Jn} = 0, \quad \text{for } I \in C \quad (2.16)$$

Here h^* optionally contains a term of the tangential tractions, which are fixed while solving the normal problem. A purpose-build fast solver is used for this problem that is called NormCG. This is an extension of the Bound-Constrained Conjugate Gradients method that is capable of dealing with constraints [39]. A preconditioner is constructed using the Fast Fourier Transform (FFT) and a prescribed total force is dealt with in the method via a deflation technique [40]. The underlying system matrix is dense, but has constant diagonals. It is not explicitly formed, a matrix-free implementation is used instead.

- (b) The tangential problem with prescribed creepages is non-linear when there are elements in the slip area. The equations are solved using iterative solvers particularly designed for this problem. In transient calculations this is the new TangCG solver [57], based on BCCG and using FFTs, or the older and slower ConvexGS solver [35]. For steady state problems the SteadyGS solver is used [38]. These approaches have in common that they all incorporate the active set strategy, that enforces the traction bounds in all elements. This means that the element division between slip and adhesion areas is updated along the way. Finally the methods are implemented in a matrix-free way, to avoid memory limitations and allowing large problems to be solved.

Chapter 3

Operation of the program

The basic structure of CONTACT is illustrated schematically in Figure 3.1.

3.1 Starting the program

3.1.1 Terminal-based, textual interface

The program CONTACT is a terminal-based program, with textual input from the keyboard or input-file and textual output to the terminal and several output-files. Consequently the program is started at the command prompt in a DOS-box or Linux terminal emulator window:

```
C:\Data\Carter2d\> contact.exe 2 carter2d
```

The later two arguments are optional. They prescribe the mode of operation (<imode>) of the program and the experiment name, which is denoted <experim> below. Note that this command line is hidden for the end-user when using the CONTACT GUI (paragraph 3.1.2).

When you start the program, you will first be asked whether you want to perform the actual run, or just want to check the input with no computation.

Mode of operation of this program.

IMODE=2: start from input file <EXPERIM>.inp,

IMODE=3: check input file <EXPERIM>.inp,

The program will use parallel computing (multi-threading) automatically when run on a multi-processor/core PC. This may be influenced via the OMP_NUM_THREADS environment variable. When set to 1 parallelisation is suppressed and only a single processor/core is used.

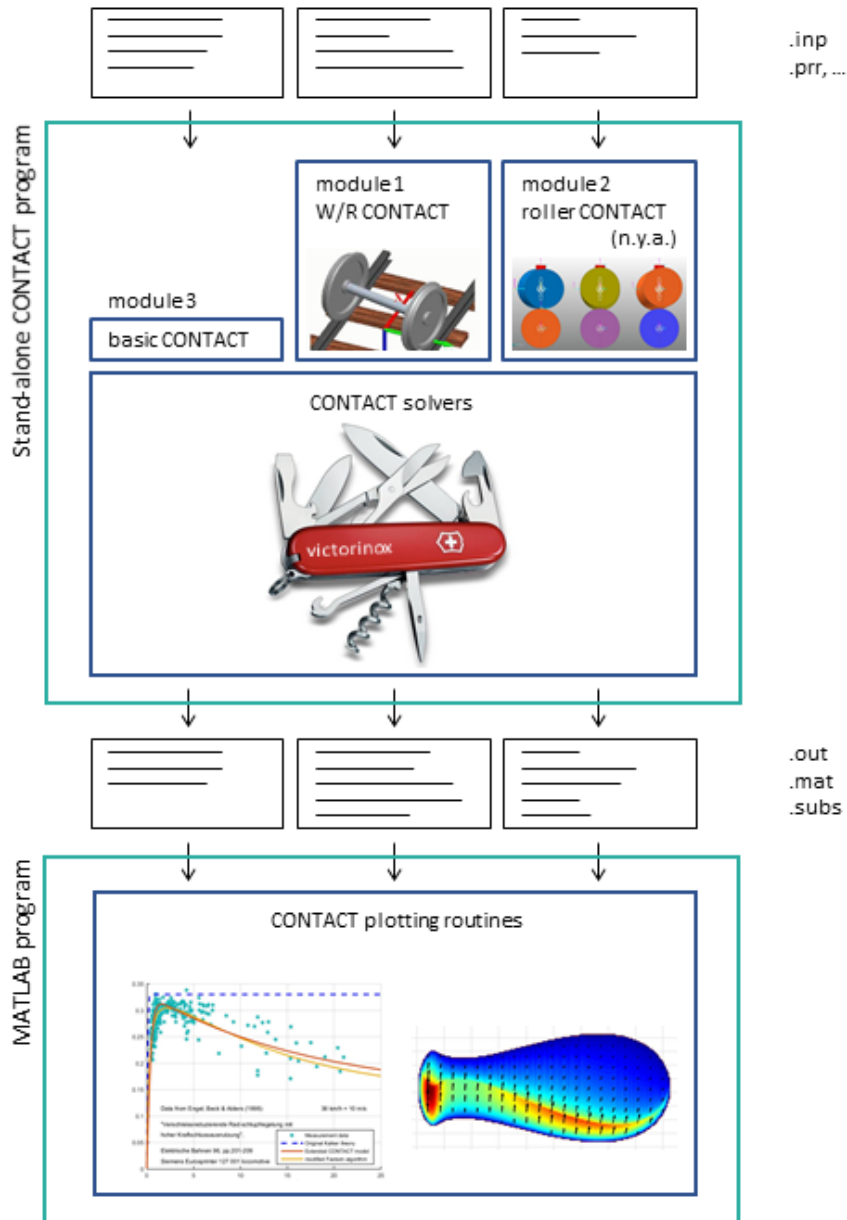


Figure 3.1: Structure of a calculation using the stand-alone CONTACT program.

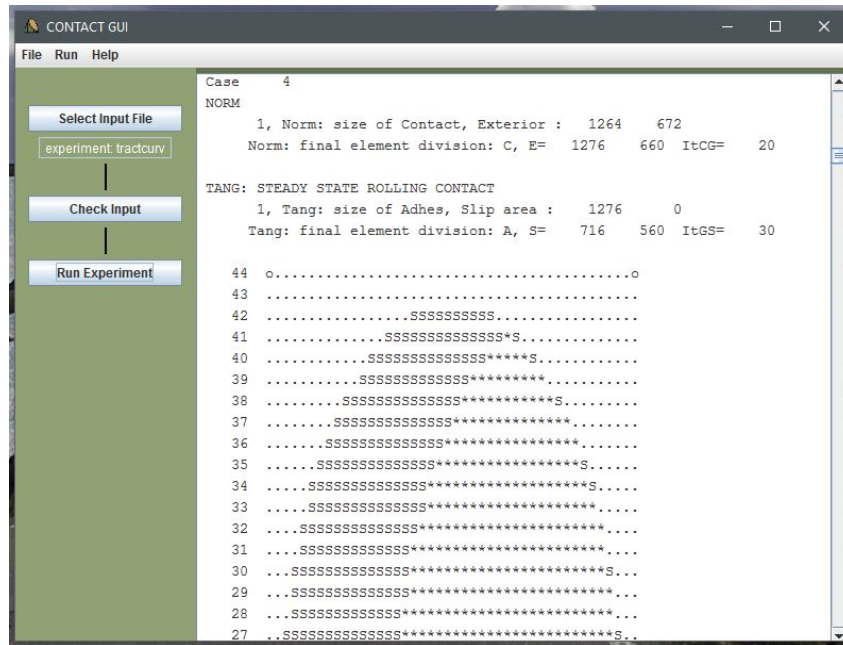


Figure 3.2: Main window of the CONTACT GUI.

3.1.2 Windows-based interface, the CONTACT GUI

The CONTACT GUI is a small auxiliary program by which CONTACT’s old-fashioned command line interface may be avoided. The main purpose of the GUI is to assist in selecting an input file and starting the calculation. No facilities are provided to create or edit the input-files themselves.

The GUI can be started from the Windows desktop short-cut (if created) or via the Windows Start menu. On Linux, the GUI is started using the script `start_gui.sh` in a terminal window (see the installation instructions in the file `README.txt`). This opens a new session as illustrated in Figure 3.2.

A typical run starts with selecting an existing input-file for CONTACT. The base name of this file (excluding the `.inp` filename extension) is the experiment name for the run. Next, the CONTACT program may be activated to check the input, using `imode = 3` as described previously. Using “Run Experiment”, CONTACT is started for the actual calculations, using `imode = 2`. The message output will be displayed in the GUI as shown in Figure 3.2. Other output files are created as well, that may be inspected for instance in MATLAB as indicated in Figure 3.1.

3.2 Multiple cases, sequences

In a single run of CONTACT multiple contact problems may be solved. These problems may be independent of each other or may form sequences.

A single case may be viewed as a single time instance. In this respect it is important to know that CONTACT uses an elastostatic approach. The contacting bodies may be accelerating or decelerating as a whole, yet the corresponding inertia terms are ignored from the stress balance ($\rho \ddot{\mathbf{u}}$ in equation (2.1)). By this approximation, the stresses inside the bodies are locally in equilibrium at all times, and any stress waves inside the bodies will be neglected.

Within this elastostatic approach, frictionless compression problems need just a single time instance. There is no memory of any sort in these problems. When a succession of loading steps is computed using different cases, the outcome of each case is independent of the order in which the cases are solved.

In frictional problems there is a dependency on time. This dependency arises through the discretisation of the slip velocity (equation (2.15)) that occurs between opposing particles of the two bodies in the contact zone. This time dependency may be dealt with in three different ways:

1. Contact formation: it may be assumed that there were no surface tractions at all at the previous time instance $t' = t - \delta t$.
2. True sequence: the surface tractions of the previous time instance t' may be defined through the solution of the previous case that was solved.
3. Steady state (rolling): the surface tractions of the previous time instance t' may be required to be identical to the solution of the new case for time t , but acting on different parts of the two bodies' surfaces.

It may be clear that cases of the first and third types may be solved by themselves, whereas the second way requires that another case has been solved before. A sequence of cases for a transient calculation should therefore start with one case of contact formation or with a steady state problem, before the actual transient cases are solved.

Even when two consecutive cases are independent of each other, the later one may re-use information of the earlier one. For instance when you compute the frictionless compression of two spheres by two normal problems. The later case may re-use the discretisation parameters and influence coefficients that have been computed by the earlier case. Moreover, the solution of the earlier case may be used as an initial estimate for the later case. In this way the computation time may be reduced.

3.3 The control integers

The specification of each case is largely governed by the “control integers”. These digits determine all kinds of things. Some specify the problem, others determine the solution method and so on. They are referred to by one letter or by a short word that describes their meaning. They are also grouped together in control words. The following list describes all possible values, and is very useful when you edit the input-file by hand.

The primary control digits are P, T, N, A and O. The other digits are modified less frequently. Do not be distracted by values that seem meaningless to you, they can be very useful when you program your own modules.

3.3.1 Modules

The program CONTACT is built up using different “modules” (sub-programs) that use a shared computational core, see Figure 3.1. The idea is that different usages of CONTACT require different “driver routines”, where different formats for the input file are required. So you first indicate to CONTACT which module you use for a case, and then specify the input which is read by the appropriate input-routine.

The modules that are available today are:

```
MODULE = 1: execute the WHEEL-RAIL processing program.  
        = 3: execute the basic, HERTZIAN & NON-HERTZIAN program.  
        = 0: STOP program execution.
```

Module 1 is targeted at wheel-rail contact analysis. It starts from a wheelset at a given track location, using generic wheel and rail profiles, locates the contact points, solves the contact patches, and then converts the results to the global coordinate frame. Module 3 activates the basic driver routine for solving Hertzian and non-Hertzian cases. When there are no more cases to compute this is signalled to the program by the artificial module number 0.

3.3.2 Module 1 – wheel-rail contact

CPBTNFS, LDCMZE, GIAOWR - the control words, compressed form

The following list is ordered the same as the letters in the control words. The first several digits concern the problem description, then follow the preprocessing actions for the computational core and lastly the solution methods and output options are described.

The first control word “**CPBTNFS**” describes the main aspects of the problem: the modes of operation regarding the configuration, time, the normal and tangential problems, and regarding the subsurface stress calculation.

C₁ - CONFIG : concerns the *configuration* or *composition* of the problem:

- 0 - wheelset on track, left wheel;
- 1 - wheelset on track, right wheel.

A configuration with a single wheel can be described with both options.

P - PVTIME : the relation of the current to the *previous case* or *previous time instance*, particularly for transient contact problems with friction:

- 0 - full continuation; the resulting tractions of the last calculated case are used as previous time instance for the calculation of the current case;
- 1 - continuation for the normal part only; the tangential tractions of the previous time instance are set to zero;
- 2 - initiation of contact; the previous tractions vanish entirely (e.g. for a Mindlin shift);
- 3 - the previous tractions are not touched.

T - TANG : specifies the *type of problem* to be solved, especially concerning the *tangential part*:

- 0 - frictionless compression: no tangential tractions required, normal problem only;
- 1 - frictional compression, or shift, or transient rolling, with material-fixed coordinates. One-step (initiation) or multi-step (continuation) depending on the P-digit;
- 2 - transient rolling, using moving (contact-fixed) coordinates, also refer to the P-digit;
- 3 - steady state rolling, using the so-called direct method.

N₁ - NORM : specifies whether the *vertical wheelset position* or *vertical force* is prescribed:

- 0 - vertical position Z_{WS} prescribed;
- 1 - total vertical force FZ prescribed.

S - STRESS : determines the operation of the subprogram STRESS for calculating *subsurface stresses* per case:

- 0 - no subsurface stresses required for this case;
- 1 - compute the stresses in the points already stored in memory;
- 2 - read/ask new control-digits for the subsurface calculation, compute stresses in the points already stored in memory;
- 3 - read/ask new subsurface points, and compute subsurface stresses for these points.

The B- and F-digits are not yet available in module 1. They are described in the input for module 3 below.

The second control word “**LDCMZE**” is used to configure the different aspects of the contact problem: materials, friction, discretization and geometry.

L - FRCLAW : concerns the *friction law* to be used:

- 0 - Coulomb friction with static and kinetic friction coefficients;
- 1 - use the friction law from parameters in storage;
- 2 - Friction law with linear/constant dependency on slip velocity;
- 3 - Friction law with rational dependency on slip velocity;
- 4 - Friction law with exponential dependency on slip velocity.

D - DISCNS : concerns the potential contact area and its *discretisation*:

- 0 - maintain the potential contact area and discretisation of the previous case;
- 1 - form the discretisation from parameters in storage;
- 2 - read/ask input parameters and form the discretisation correspondingly.

C₃ - INFLCF : concerns the material parameters and corresponding *influence coefficients*:

- 0 - maintain influence coefficients of the previous case;
- 1 - generate influence coefficients from parameters in storage;
- 2 - using analytical influence coefficients for the half-space with piecewise constant tractions; read parameters from file and compute new influence coefficients;
- 3 - using analytical influence coefficients for the half-space with bilinear tractions; read parameters from file and compute new influence coefficients;
- 9 - using numerical influence coefficients; read parameters and filename, read pre-computed influence coefficients from file.

M - MATER : type of *material model* to be used

- 0 - purely elastic contact;
- 1 - viscoelastic contact;
- 2 - simplified theory with a single flexibility;
- 3 - simplified theory with three flexibilities;
- 4 - elastic contact with an elasto-plastic interface layer.

Z₁ - ZTRACK : concerns the *track design geometry*, profile(s) and optional rail deviations:

- 0 - maintain track dimensions, profile(s) and deviations;
- 1 - read new track dimensions;
- 2 - read new rail deviations for current side of the track (dependent on C_1 -digit);
- 3 - read new track dimensions, one profile for both sides of the track, and read rail deviations for current side of the track;
- 4 - read new track dimensions, separate profiles for the two sides of the track, and read new rail deviations for both sides of the track.

E₁ - EWHEEL : concerns the *wheelset geometry* and profile(s), and position and velocity data:

- 0 - maintain wheelset geometry, profile(s), position and velocity data;
- 1 - read new position data;
- 2 - read new position and velocity data;
- 3 - read new geometry, one profile for both sides of the wheelset, read new position and velocity data;
- 4 - read new geometry, separate profiles for the two sides of the wheelset, read new position and velocity data.
- 5 - same as 2, including flexible wheelset deviations for current side of wheelset;
- 6 - same as 2, including flexible wheelset deviations for both sides of wheelset;
- 7 - same as 3, including flexible wheelset deviations for current side of wheelset.
- 8 - same as 4, including flexible wheelset deviations for both sides of wheelset.

The third control word “**GIAOWR**” concerns the output of the program and the flow of the calculations.

G - GAUSEI : used for fine-tuning of the *iterative solvers*, particularly for the tangential problem.

- 0 - use the default solver (SteadyGS when $T = 3$, otherwise use TangCG) with default ω 's, read max. iterations from input file;
- 1 - use the solver settings from storage;
- 2 - use ConvexGS at all times, read ω 's from input file;
- 3 - use SteadyGS when possible ($T = 3$), read ω 's from input file.
- 4 - use the default solver, read parameters for slip velocity iteration (page 23) from input file.

I - IESTIM : governs the *initial estimate*:

- 0 - previous case does not provide a good initial estimate. Start from zero tractions and fill element division with a rough guess, based on the undeformed distance;
- 1 - use previous solution, regularize tractions. The element division of the previous case is used as initial estimate, and the tractions are adjusted to this state;
- 2 - use normal part of previous case only. The normal tractions are maintained, but all elements start with zero tangential tractions, in the adhesion area;
- 3 - previous case gives a good initial estimate. Use its tractions and element division (not touched).

A - MATFIL, surface tractions : governs the use of the *Matlab-file* `<experim>.<case>-.mat`:

- 0 - the `mat`-file is not created;
- 1 - the detailed results of the case are written to a Matlab-file `<experim>.<case>.mat`, for points inside the contact area;
- 2 - the detailed results are written for all points of the potential contact area.

Note: a separate `A`-digit is maintained to control the output of the subsurface stress calculation, as described in Section 5.9.1.

O - OUTPUT, surface tractions : governs the extent of the output to the *output-file* `<experim>.out`:

- 0 - no results are printed to the output-file (derived quantities *are* computed, and stored in internal memory for use in other calculations);
- 1 - minimum output is printed, just the global results;
- 2 - the global input and output quantities are printed;
- 3 - a picture is shown of the contact area and its division into adhesion and slip areas;
- 4 - the detailed solution inside the contact area (tractions, slip, and other quantities of interest) is printed as well, in addition to 2.

You may typically set `O = 2`, `A = 0` to get the global results of all cases and then use `A = 1` for the cases that you want to investigate in detail with the Matlab plot-programs.

Note: a separate `O`-digit is maintained to control the output of the subsurface stress calculation, as described in Section 5.9.1.

W - FLOW : governs the extent of the *flow trace* to the screen and the output-file:

- 0 - no flow trace printed, except the case number;
- 1-5 - number of iteration processes to show flow trace for: the outer loop (Panagiotopoulos' process), the loop for obtaining a slip velocity dependent traction bound, the NORM and TANG algorithms, Newton-Raphson procedures, and iterative procedures CG, ConvexGS and SteadyGS.

9 - full flow trace, including intermediate pictures of the element divisions.

Usually the levels 3 and 4 are most convenient.

R - RETURN : *return* to main program.

Modules 1 and 3 are organized so that you stay in one module until you give a sign that you do not want so. This sign can be given with the R-digit. It is also possible to skip cases in the input-file.

0 - calculate the solution, then stay in the same module for the next case;

1 - calculate the solution, return to the main program after this case;

2 - perform preprocessing actions only (skip this case), then stay in the same module for the next case;

3 - perform preprocessing actions only (skip this case), return to the main program after this case.

3.3.3 Module 3 – basic Hertzian and non-Hertzian contact

The input for module 3 is largely the same as described for module 1 above. Additional functionality is provided via the B- and F-digits, whereas the functionality of the N-digit is changed:

B - BOUND : selects the approach to be used for the normal problem, the *traction bound*:

0 - full linearly elastic model and contact conditions;

2 - elliptical traction bound derived from Hertzian problem data (requires IPOTCN < 0);

3 - parabolical traction bound derived from Hertzian problem data (requires IPOTCN < 0);

4 - simple double half-elliptical contact area and pressure distribution (SDEC [24], requires IPOTCN = -4).

Note: options 2 to 4 are approximate by nature. They are primarily meant for the FASTSIM approach, i.e. M = 2, 3 below.

N₃ - NORM : specifies whether the *normal force* or *approach* is prescribed:

0 - approach PEN (penetration) prescribed;

1 - total normal force FN prescribed.

F₃ - FORCE : specifies the number of *tangential forces* that are prescribed:

0 - creepages CKSI and CETA prescribed;

- 1 - total force **FX** and creepage **CETA** prescribed;
- 2 - total forces **FX** and **FY** prescribed.

In **module 3**, the **Z**- and **E**-digits are also used to describe the geometry and the rigid slip. However, the meaning of these digits is different from the interpretation in **module 1**:

Z₃ - RZNORM : concerns the right hand side of the normal problem, the *undeformed distance*:

- 0 - maintain undeformed distance and planform of the previous case;
- 1 - form undeformed distance from parameters in storage;
- 2 - read new parameters and compute undeformed distance.

E₃ - EXRHS : the *extra term* of the *rigid slip*, the right hand side of the tangential problem:

- 0 - set the extra term equal to zero;
- 1 - maintain the extra term of the previous case;
- 9 - read a new extra term from the input (file).

Chapter 4

Module 1: wheel/rail contact

This chapter describes the input and output quantities of CONTACT that are specific to module 1 for wheel/rail contact processing. The additional inputs that are needed for generic non-Hertzian contact (module 3) are described in Chapter 5.

4.1 Wheel-rail coordinates and conventions

Track coordinates Wheel-rail contact analysis starts from track coordinates as defined in Figure 4.1. These are formed by a plane resting on the (canted) rails, with origin at the track center line and x_{track} aligned with the track center line. In rail vehicle dynamics, the track origin may be defined as the point on the track curve closest to the wheelset center of mass (CM), see e.g. [21]. This is denoted here as s_{ws} , with s (s_1 , longitudinal) the parameter used to describe the track curve.

Gauge width computation The lateral position of the rails is defined by setting the position of the rail origin point. This uses the gauge width between the inner faces of the two rails (i.e. the grey steps in Figure 4.1). Figure 4.2 (left) shows the rail as seen within

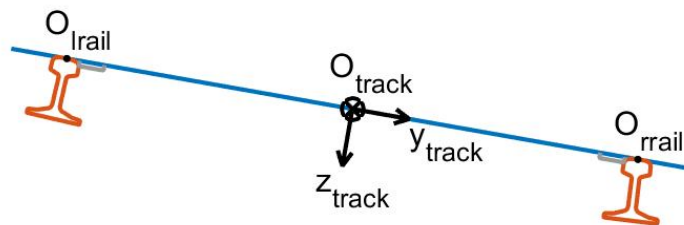


Figure 4.1: Definition of the track coordinate system, at the center of the plane resting on the (inclined) rails.

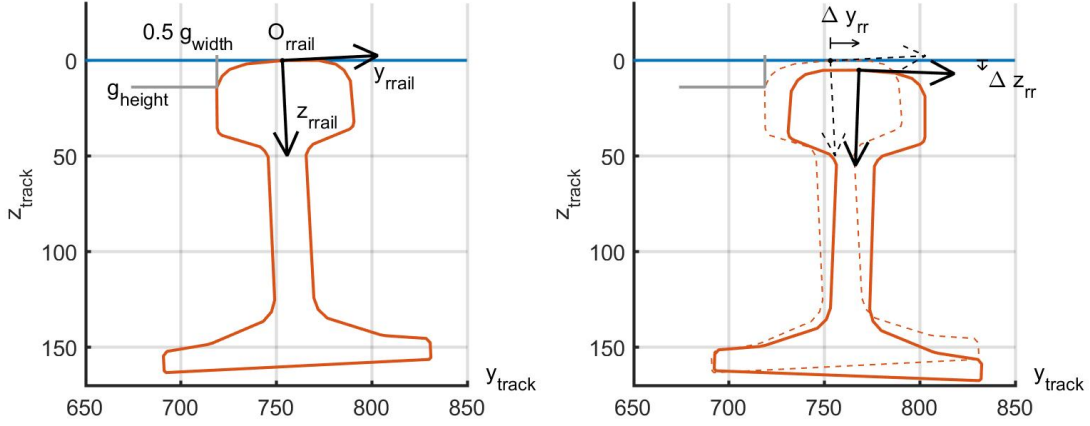


Figure 4.2: *Left: definition of rail coordinates with respect to the track system, with positive cant angle (right hand rule: installation angle < 0), with the gauge width and gauge measuring height. Right: displacement due to track irregularities for positive $\Delta\phi_{rr}$, Δy_{rr} and Δz_{rr} .*

the track system. In this case the rail has an installation angle of -2.9° corresponding to a cant of 1 : 20. The profile is specified with respect to the rail origin O_{rrail} . It is first rotated and then shifted up to just touch the track plane and shifted left to touch the gauge stop. Note that the rail origin can be the highest point in the uncanted rail, which need not be the highest point on the rail after cant is applied.

Track deviations Track irregularities may be defined that displace the rails with respect to their design positions and orientations, see Figure 4.2, right. The rails are rotated about their origins O_{lrail} , O_{rrail} by roll angles $\Delta\phi_{lr}$ and $\Delta\phi_{rr}$, and are shifted by Δy_{lr} , Δz_{lr} , Δy_{rr} , Δz_{rr} with respect to the track origin, which isn't affected by the irregularities. Note that the rails typically won't be precisely touching the track plane anymore.

These rail irregularities may be static/permanent, but may be due to track flexibility too. In such a case, the corresponding velocities $v_{lr} = \Delta\dot{y}_{lr}, \dots$ may be specified also.

Wheelset degrees of freedom The position of a wheelset is characterized by the position of its center of mass along the track curve (s_{ws}), and the position and orientation of its axle with respect to the track reference (Figure 4.3). The orientation is defined in the roll–yaw–pitch convention: starting with the axle parallel to the y_{track} -direction, the wheelset is rolled about its x -axis by ϕ_{ws} , then yawed by ψ_{ws} about the new vertical axis z_{wset} and then pitched by θ_{ws} about the axle, i.e. the new y_{wset} -axis. After this the wheelset is shifted to its position $[0, y_{ws}, z_{ws}]^T$.

Wheelset geometry The wheel profiles are specified relative to the wheel origins O_{lwheel} , O_{rwheel} , which are subsequently placed at $[0, \pm(y_{fbdist}/2 - y_{fbpos}), r_{nom,w}]^T$ with respect to the

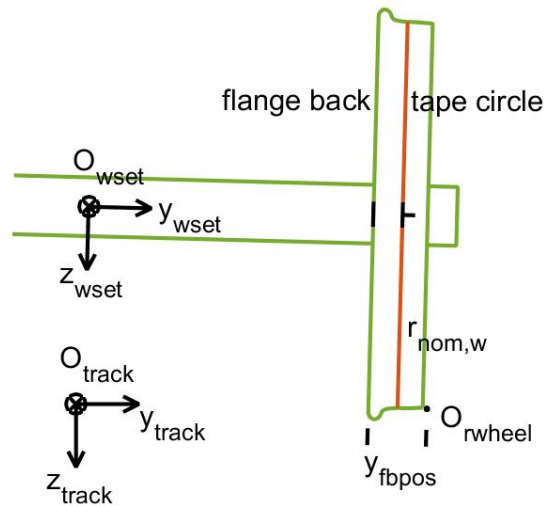


Figure 4.3: *Definition of wheelset reference with respect to track coordinates and wheel reference marker with respect to wheelset coordinates.*

wheelset center. Here y_{fbdist} is the flange-back distance, y_{fbpos} is the offset from wheel origin to the flange back position, and $r_{nom,w}$ is the wheel radius at the tape circle line. The y -position is negative for the left wheel, positive at the right side of the wheelset. It is convenient when the wheel origin is at the tape circle line, in which case the point $[y_w, z_w]^T = [0, 0]^T$ is on the wheel profile, yet other choices are possible for the wheel origin also.

Flexible wheelset deviations Flexible wheelset deviations may be defined that displace the wheel profiles with respect to their design positions and orientations. Increments may be specified for all six position and orientation variables to support axle bending and torsion. The corresponding velocities may be specified as well.

Using right-sided profiles In the input, all profiles must be specified for the right side of the wheelset and track, for which optional mirroring is provided. The calculation of the left side wheel/rail pair is then instructed to the program with the control integer CONFIG (C_1 , page 31). Inside the program, this left wheel/rail pair is mirrored and solved using the equivalent right wheel/rail pair. This shows up in the detailed outputs, especially the contact grid, that is given for the mirrored configuration.

Kinematic configuration Note that CONTACT focuses on the contact problem and not the the wheelset balance (equilibrium) or motion (dynamics). Each case of the simulation therefore concentrates on a single wheel/rail combination. The full position and velocity states of the wheelset are typically prescribed when using CONTACT from a vehicle dynamics simulation. Alternatively, some total forces may be specified also, which is particularly useful in stand-alone calculations.

4.2 Track geometry

The track geometry consists of one or two rail profiles, the track dimensions and rail irregularities. New geometry is input when the control integer ZTRACK is 1–4 (page 32). Using $Z_1 = 3$, one profile is read that’s used at both sides of the track. With $Z_1 = 4$, separate profiles are read for the left and right rails, respectively.

Rail profiles should be defined for right rails, i.e. with the points ordered from the track center to the field side, with increasing y_{rail} coordinates. If an actual file concerns a left rail, with the track center at a high y -value, mirroring may be enabled to convert into the required format. The easiest way of working is to define the profiles upright, without cant, with the rail origin centered in the rail surface.

MIRROR_Y	[–]	Use the profile as is (0) or mirror y -values (1).
RFNAME	[–]	Input filename for a rail profile.

The rail profiles may be provided in SIMPACK **pr**r-format or Miniprof **ban**-format, the format is taken from the file extension. SIMPACK profiles are processed according to the options that are specified in the file: mirroring, reordering, scaling, etc. Miniprof profiles are subject to some automatic corrections, but may need manual preparation as well.

Rails (profiles) are rotated by an installation angle which is +CANT for a left and –CANT for a right rail. After this the rail position is derived using the gauge width and gauge measuring height parameters. This gauge computation can be disabled by setting $g_{height} \leq 0$. The rail origin is then placed simply at $[0, \pm g_{width}/2, 0]^T$. In a configuration using one wheel on one rail, this may be used with $g_{height} = -1, g_{width} = 0$.

GAUGWD	[<i>mm</i>]	Distance g_{width} between the inner faces of the two rails.
GAUGHT	[<i>mm</i>]	Height g_{height} below the track plane at which the gauge width is measured. The gauge computation is disabled when $g_{height} \leq 0$.
CANT	[<i>rad</i>]	Rotation angle from track vertical (z_{track}) to rail vertical (z_{rail}), positive for inclination towards the track center.

Rail irregularities provide offsets with respect to the ideal (design) geometry. Track coordinates are used, such that $\Delta y > 0$ is towards the track center for a left rail, but to the field side when the right rail is considered.

DYRAIL	[<i>mm</i>]	Offset Δy_{rail} of the rail origin with respect to the design position in track coordinates.
DZRAIL	[<i>mm</i>]	Offset Δz_{rail} of the rail origin with respect to the design position in track coordinates.

DROLLR	[rad]	Rotation $\Delta\phi_{rail}$ of the rail origin with respect to the design orientation in track coordinates, positive using the right-hand rule.
VYRAIL	[mm/s]	Velocity v_y^{rail} of the rail origin with respect to the design position in track coordinates.
VZRAIL	[mm/s]	Velocity v_z^{rail} of the rail origin with respect to the design position in track coordinates.
VROLLR	[rad/s]	Angular velocity v_ϕ^{rail} of the rail origin with respect to the design orientation in track coordinates, positive using the right-hand rule.

4.3 Wheelset geometry

The wheelset geometry consists of the wheelset dimensions, one or two wheel profiles, and the position and velocity of the wheelset with respect to the track. New data are input when the control integer EWHEEL is 1–4 as described on page 33.

FBDIST	[mm]	Lateral distance y_{fbdist} between the inner faces (flange backs) of the two wheels of the wheelset.
FBPOS	[mm]	Lateral position y_{fbpos} of the flange back with respect to the wheel profile origin O_w . (In the configuration of Figure 4.3, y_{fbpos} could be -140 mm .)
NOMRAD	[mm]	Nominal radius $r_{nom,w}$ of the wheel, i.e. the radius at the tape circle line.

Wheel profiles should be defined for a right wheel, with the points ordered from the outside towards the flange, with decreasing y_{wheel} coordinate values. Mirroring may be enabled to convert into the required format.

MIRROR_Y	[–]	Use the profile as is (0) or mirror y -values (1).
WFNAME	[–]	Input filename for the wheel profile.

Wheel profiles may be provided in SIMPACK `prw`-format or Miniprof `wh1`-format, the actual format that’s used is determined from the file extension. SIMPACK profiles are processed according to the options that are specified in the file: mirroring, reordering, clipping, etc. Miniprof profiles are subject to some automatic corrections regarding the ordering of the points and the positive z -direction, but may need further manual preparation as well.

The contact geometry is governed by the position of the wheel with respect to the rail. This is computed using the position of the wheelset center of mass, and the wheelset roll, yaw and

pitch angles.

S_WS	[mm]	Wheelset position s_{ws} along the track center line, used mainly for output purposes. Note: $s = s_1$ refers here to the (longitudinal) rolling direction.
Y_WS	[mm]	Lateral position y_{ws} of the wheelset center of mass in terms of track coordinates.
Z_WS	[mm]	Vertical position z_{ws} of the wheelset center of mass in terms of track coordinates.
ROLL	[rad]	Wheelset roll angle ϕ_{ws} with respect to the track plane.
YAW	[rad]	Wheelset yaw angle ψ_{ws} with respect to the track center line x_{track} .
PITCH	[rad]	Wheelset pitch angle θ_{ws} , i.e. rotation about the wheelset axle. This is currently not used. It becomes of relevance when out-of-round wheels need to be described.

The orientation of roll, yaw and pitch angles is defined using the right hand rule. For instance, the configuration of Figure 4.3 shows a positive roll angle ϕ_{ws} , defined as the rotation about the x_{track} -axis from the positive z_{track} - to positive z_{wset} -axis. Likewise, the wheelset will tend to the right in the rear view of Figure 4.3 if a positive yaw angle is given. This shows up in the results of the Manchester benchmark example (Figure 6.7), where the contact patches on left and right wheels shifted to positive and negative x -values, respectively.

The creepages for the contact problem are obtained from the wheelset velocity. These are computed from the rates of change of the position parameters.

VS_WS	[mm/s]	Wheelset forward velocity $v_{s_1} = v_x$ along the track center line.
VY_WS	[mm/s]	Wheelset lateral velocity $v_y = \dot{y}_{ws}$ with respect to track coordinates.
VZ_WS	[mm/s]	Wheelset vertical velocity $v_z = \dot{z}_{ws}$ in terms of track coordinates.
VROLL	[rad/s]	Wheelset rate of roll $\dot{\phi}_{ws}$.
VYAW	[rad/s]	Wheelset yaw rate $\dot{\psi}_{ws}$.
VPITCH	[rad/s]	Wheelset angular velocity $\omega_{ws} = \dot{\theta}_{ws}$.

According to the right hand rule, the pitch velocity ω_{ws} is negative for forward rolling.

Flexible wheelset deviations provide offsets with respect to the ideal (design) geometry. Wheelset coordinates are used, such that $\Delta y > 0$ is towards the center of mass for a left wheel, but to the field side when the right wheel is considered.

DXWHL . . . DZWHL	[<i>mm</i>]	Offsets $\Delta x_{whl}, \Delta y_{whl}, \Delta z_{whl}$ of the wheel profile origin with respect to the design position in wheelset coordinates.
DROLLW, DYAWW, DPITCHW	[<i>rad</i>]	Rotations $\Delta \phi_{whl}, \Delta \psi_{whl}, \Delta \theta_{whl}$ of the wheel profile origin with respect to the design orientation in wheelset coordinates, positive using the right-hand rule.
VXWHL . . . VZWHL	[<i>mm/s</i>]	Velocities $\Delta v_x^{whl}, \Delta v_y^{whl}, \Delta v_z^{whl}$ of the wheel profile origin with respect to the design position in wheelset coordinates.
VROLLW, VYAWW, VPITCHW	[<i>rad/s</i>]	Angular velocities $\Delta v_\phi^{whl}, \Delta v_\psi^{whl}, \Delta v_\theta^{whl}$ of the wheel profile origin with respect to the design orientation in wheelset coordinates, positive using the right-hand rule.

4.4 Specifying the total forces

Using the N₁-digit, the total vertical force F_z may be given as part of the problem specification. In that case, the program will raise or lower the wheelset (z_{ws}) as needed to achieve the desired force. The wheelset roll angle will not be adjusted.

The F-digit is not yet supported in module 1.

4.5 Potential contact area and discretisation

The potential contact area is a rectangular area that should encompass the actual contact area. Within module 1, this potential contact is determined automatically. The parameters that must be set by the user concern the discretisation and a safety factor for adjusting the automatic calculation.

DX	[<i>mm</i>]	Size δx of each grid element in longitudinal (x_{track} or x_{rail}) direction.
DS	[<i>mm</i>]	Size δs of each grid element in lateral ($s = s_2$) direction tangent to the rail surface.
DQREL	[–]	Relative size $c = \delta q / \delta x$ of the rolling distance δq traversed per time step compared to the grid size δx .
SAFETY	[–]	Multiplication factor for enlarging (> 1) or reducing (< 1) the default size use for the potential contact area. The value 1 will be appropriate in most cases.

The program automatically sets $c = 1$ in the computation of steady rolling. For transient rolling, the value 1 is also preferred [36, 55].

4.6 Global outputs for module 1

Global output quantities are written to the output file `<experim>.out` when the 0-digit is set to values 1 – 4.

The global part of the output firstly contains a description of the problem that is solved ($0 \geq 2$), its control digits and the primary input values that are used (except discretisation and geometry). Then several output quantities are displayed for the wheel/rail pair as a whole: the total forces as determined by the program, or the wheelset positions and rates that had to be found.

With positive z pointing downwards (Figure 4.3), the rail is the output body (a) = (1).

FX_TR	[N]	Total force $F_{x_{tr}}$ on the rail in track longitudinal direction x_{track} .
FY_TR	[N]	Total force $F_{y_{tr}}$ on the rail in track lateral direction y_{track} .
FZ_TR	[N]	Total force $F_{z_{tr}}$ on the rail in track vertical direction z_{track} .

The forces are also rotated to the wheelset orientation:

FX_WS	[N]	Total force $F_{x_{ws}}$ on the rail in wheelset longitudinal direction x_{wset} .
FY_WS	[N]	Total force $F_{y_{ws}}$ on the rail in wheelset lateral direction y_{wset} .
FZ_WS	[N]	Total force $F_{z_{ws}}$ on the rail in wheelset vertical direction z_{wset} .

After the values for the wheel/rail pair as a whole, data are displayed for all the contact patches that have been detected and solved. This concerns the contact reference point, the creepages and penetration, and the forces and moment of each separate patch.

The *contact reference position* is the origin of the local coordinates used for solving the contact problem [44]. It is determined by a heuristic rule to be centered within the contact patch in a weighted sense. This may be different from the *initial contact point* where the undeformed distance is minimum.

The contact reference position is presented in terms of the track-, rail- and wheel coordinates.

XCP_TR, . .	[mm]	Position of the contact reference point in track coordinates.
XCP_R, . .	[mm]	Position of the contact reference point on the rail profile.
XCP_W, . .	[mm]	Position of the contact reference point with respect to the wheel profile.
S_R, S_W	[mm]	s (s_2) position of the contact reference point with respect to the profile origin measured along the curved rail or wheel surface.

DELTRR [*rad*] Contact angle: rotation from the track z -axis to the contact n -axis. Typically positive for left wheel/rail pairs and negative at right wheel/rail pairs.

For left wheel/rail pairs, the positions within the profiles ($.R$, $.W$) are given for the mirrored right wheel/rail combination, with positive y -coordinates at the field side and negative values towards the track center.

The creepages, approach, total forces and moments that are presented next are the same as the global outputs of module 3 (Section 5.7). Note that **FY** and **MZ** of module 3 are renamed here to **FS** and **MN**, to emphasize that the values are defined using the contact local coordinates.

Chapter 5

Module 3: in- and output quantities

5.1 Material parameters

5.1.1 Homogeneous elastic materials

The input parameters for elastic materials are:

GG(ia) $[N/mm^2]$ Modulus of rigidity $G^{(a)}$ of body ia, $G^{(a)} > 0$.

POISS(ia) $[-]$ Poisson's ratio $\nu^{(a)}$ of body ia, $0 \leq \nu^{(a)} \leq 0.5$.

The modulus of rigidity is also known as the (elastic) shear modulus. It is related to the modulus of elasticity (Young's modulus) E by

$$G^{(a)} = \frac{E^{(a)}}{2(1 + \nu^{(a)})} \quad (5.1)$$

From these values the program computes the following combined quantities:

GA $[N/mm^2]$ Combined modulus of rigidity G .

NU $[-]$ Combined Poisson's ratio ν .

AK $[-]$ Difference parameter K .

These values are defined by

$$\frac{1}{G} = \frac{1}{2} \left(\frac{1}{G^{(1)}} + \frac{1}{G^{(2)}} \right), \quad \nu = \frac{1}{2} \left(\frac{\nu^{(1)}}{G^{(1)}} + \frac{\nu^{(2)}}{G^{(2)}} \right), \quad \frac{K}{G} = \frac{1}{4} \left(\frac{1 - 2\nu^{(1)}}{G^{(1)}} + \frac{1 - 2\nu^{(2)}}{G^{(2)}} \right) \quad (5.2)$$

They characterize the combined influence function $\mathbf{A}(\mathbf{x}, \mathbf{x}')$ of the two bodies, see equation (2.5).

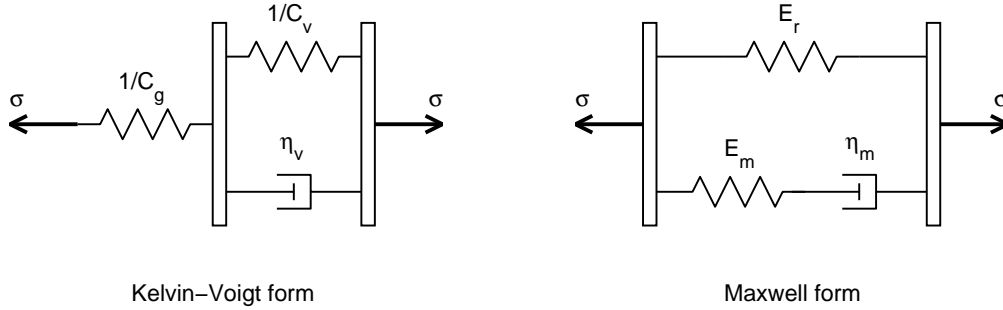


Figure 5.1: Mechanical analog networks of the Standard Linear Solid (SLS) model of viscoelastic materials, featuring the equivalent Kelvin-Voigt (left) and Maxwell forms (right).

5.1.2 Viscoelastic materials

An extension has been made to viscoelastic materials, but only for steady state rolling problems with rolling in positive x -direction ($\chi = \chi = 0$). The required influence coefficients are calculated by numerical integration as described in [15], appendix D, with further information provided in [52, 53].

The viscoelastic material model that is incorporated in CONTACT is the Standard Linear Solid (SLS) model. This model adequately describes the viscoelastic stress relaxation and creep phenomena for many linear viscoelastic materials in the first order [27].

For a standard creep test, the *creep compliance* is described with the SLS as

$$C_{crep}(t) = \varepsilon(t)/\sigma_0 = C_g + (C_r - C_g)(1 - e^{-t/\tau_c}). \quad (5.3)$$

This employs the initial compliance C_g , the final compliance $C_r > C_g$ and the creep relaxation time τ_c .¹

For a standard stress relaxation test, the *relaxation modulus* is described with the SLS as

$$E_{rel}(t) = \sigma(t)/\varepsilon_0 = E_r + (E_g - E_r)e^{-t/\tau_E}. \quad (5.4)$$

This employs the initial Young's modulus E_g , the final Young's modulus $E_r < E_g$ and the stress relaxation time τ_E .

Two mechanical analog network models that are equivalent and that are closely linked to these two standard relaxation tests, are the Kelvin-Voigt and Maxwell forms of the SLS, see Figure 5.1. The element parameters are the spring stiffnesses and compliances $E = 1/C [N/mm^2]$ and the damper viscosity $\eta [N s/mm^2]$. All material parameters, including the Poisson's ratio ν , are assumed to be constant over time.

Apart from the elastic parameters `GG(ia)` and `POISS(ia)` per body, the extra input quantities in the program for viscoelastic materials are

¹The subscripts $_g$ and $_r$ refer to the initial “glassy” and the final “rubbery” states of the viscoelastic material [27].

Model parameters	Inputs to CONTACT		
	GG	FG	TC
Maxwell form $E_r, E_m = E_g - E_r, \eta_m$	$\frac{E_r + E_m}{2(1 + \nu)}$	$\frac{E_m}{E_r}$	$\frac{E_r + E_m}{E_r E_m} \eta_m$
Kelvin-Voigt form $C_g, C_v = C_r - C_g, \eta_v$	$\frac{1}{C_g} \frac{1}{2(1 + \nu)}$	$\frac{C_v}{C_g}$	$C_v \eta_v$
Creep relaxation test C_g, C_r, τ_C	$\frac{1}{C_g} \frac{1}{2(1 + \nu)}$	$\frac{C_r - C_g}{C_g}$	τ_C
Stress relaxation test E_g, E_r, τ_E	$\frac{E_g}{2(1 + \nu)}$	$\frac{E_g - E_r}{E_r}$	$\frac{E_g}{E_r} \tau_E$

Table 5.1: Relations of the viscoelastic input quantities to the model parameters for a number of different descriptions of viscoelastic relaxation.

- FG(ia) [–] Ratio of Kelvin-Voigt spring compliance constants $C_g^{(a)}/C_v^{(a)}$ of body ia, $C_g^{(a)}/C_v^{(a)} \geq 0$.
- TC(ia) [s] Creep relaxation time $\tau_C^{(a)}$ of body ia, $\tau_C^{(a)} \geq 0$.

The relations between the input quantities and the different model parameters are summarized in Table 5.1. From these relations it follows that

$$f_g = \frac{E_g - E_r}{E_r} \leftrightarrow E_r = \frac{E_g}{1 + f_g}, \text{ and consequently } G_r = \frac{G_g}{1 + f_g}. \quad (5.5)$$

Note that when viscoelastic material behaviour is used, the calculation of subsurface stresses is still based on the elastic half-space approach. This means that the true surface stresses due to viscoelastic rolling contact are not propagated entirely correct into the subsurface.

5.1.3 Flexibilities of the simplified theory

CONTACT more or less incorporates the modified FASTSIM algorithm [33] for quickly solving for the tangential surface tractions \mathbf{p}_t using the simplified theory [11, 13]. In this approach the surface particles are assumed to be able to move independently of each other with response \mathbf{u}_t linear in the surface traction \mathbf{p}_t , as if the bodies consisted of a set of independent springs with (combined) flexibility parameter L . This is also called the Winkler foundation approach.

- FLX [mm^3/N] Flexibility parameter L .

In Hertzian problems (Section 5.3.1), three different flexibilities L'_ξ , L'_η and L'_ϕ may be used for longitudinal, lateral and spin creepage respectively. These flexibilities are computed automatically according to the original FASTSIM algorithm when the M-digit is set to 3.

The modifications proposed in [33] are based on a variable flexibility, increasing with the ratio of the slip area to the area of adhesion:

$$L_{eff} = \frac{L}{k}, \quad \text{with } k = k_0 \left(\alpha_{inf} + \frac{1 - \alpha_{inf}}{1 + \beta \varepsilon} \right). \quad (5.6)$$

This increases the flexibility when $k < 1$. The input parameters are as follows:

KO_MF	[-]	Initial value k_0 of Kalker’s reduction factor at creep values close to zero, $0 < k_0 \leq 1$.
ALFAMF	[-]	Fraction $\alpha_{inf} = k_\infty/k_0$ of the reduction factor at creep values approaching infinity, $0 \leq \alpha_{inf} \leq 1$.
BETAMF	[-]	Non-dimensional parameter β that governs how quickly k changes with increasing slip area, $\beta \geq 0$.

The ratio of the slip to adhesion areas is described by the parameter ε that’s computed in the program on the basis of prescribed creepages. It’s effect is switched off by using $\alpha_{inf} = 1$. The original Fastsim algorithm is then recovered by using slope reduction factor $k_0 = 1$.

The simplified theory is not realistic for computing the normal problem. For this the full half-space approach is used (option B = 0). In Hertzian problems, elliptical or parabolical traction bounds may be used as alternatives, by setting B = 2 or 3. Of these options the parabolical traction bound is advised. The elliptical traction bound and the half-space solution cannot predict the shapes of adhesion and slip areas well in cases with large spin, where there should be slipping near the leading edge of the contact area.

When using the simplified theory, the material parameters of Section 5.1.1 must still be specified. These are used in the normal problem when B = 0 and in the calculation of subsurface stresses, which are both based on the full half-space approach.

5.1.4 Interfacial layer of contaminants

Option M = 4 concerns the contact between two homogeneous, elastic bodies as described in Section 5.1.1, separated by a so-called third-body layer or interfacial layer formed of metal oxides (wear debris), sand and clay, etc. This model is shown schematically in Figure 5.2.

The elasto-plastic rheological model proposed by Hou et al. [8] is used to relate the displacement difference $\mathbf{u}_t^{(3),top} - \mathbf{u}_t^{(3),bot} = -\mathbf{u}_t^{(3)}$ in the layer to the shear stress $\boldsymbol{\tau} = \mathbf{p}_t^{(1)}$:

$$\begin{cases} \mathbf{u}_t^{(3)} = \frac{\mathbf{p}_t^{(1)} h^{(3)}}{G^{(3)}} & \text{elastic regime, } \|\mathbf{p}_t^{(1)}\| \leq \tau_c^{(3)} \\ \mathbf{u}_t^{(3)} = \left(\frac{\tau_c^{(3)}}{G^{(3)}} + \frac{\|\mathbf{p}_t^{(1)}\| - \tau_c^{(3)}}{k^{(3)}} \right) h^{(3)} \frac{\mathbf{p}_t^{(1)}}{\|\mathbf{p}_t^{(1)}\|} & \text{plastic regime, } \|\mathbf{p}_t^{(1)}\| > \tau_c^{(3)} \end{cases} \quad (5.7)$$

Note the sign convention used for $\mathbf{u}_t^{(3)}$, which makes the total displacement difference

$$\mathbf{u}_t = \mathbf{u}_t^{(1)} - \mathbf{u}_t^{(2)} + \mathbf{u}_t^{(3)}. \quad (5.8)$$

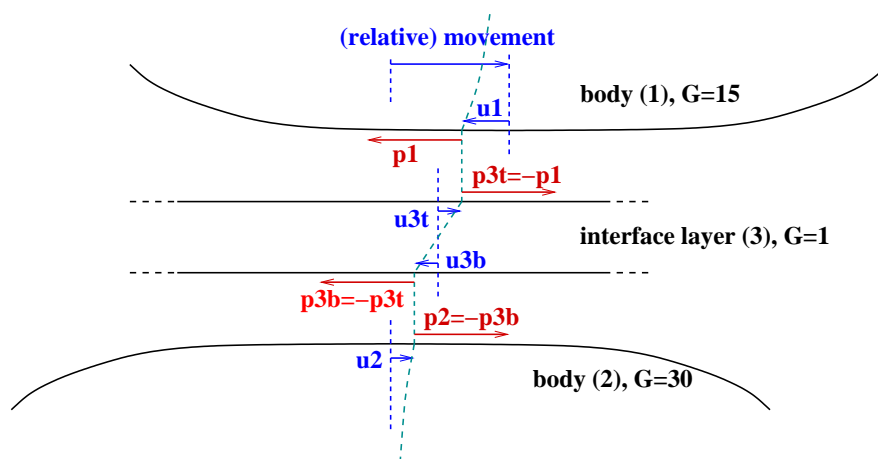


Figure 5.2: Illustration of tractions $\mathbf{p}_t^{(a)}$ acting on bodies $a = 1..3$ and displacements $\mathbf{u}_t^{(a)}$ in the bodies as a result of (gross) relative movement.

The strength of the interfacial layer is thus described by four parameters:

GG3	$[N/mm^2]$	Shear elastic modulus of the interface layer, $G^{(3)} > 0$.
LAYTHK	$[mm]$	Thickness $h^{(3)}$ of the interface layer.
TAUCRT	$[N/mm^2]$	Critical shear stress $\tau_c^{(3)}$ above which plasticity effects occur in the interface layer.
GGPLST	$[N/mm^2]$	Shear plastic modulus $k^{(3)}$ of the interface layer.

Note: the plastic regime is not implemented yet, currently TAUCRT and GGPLST are ignored.

5.1.5 Numerically calculated influence coefficients

A different type of “material model” is one in which the influence function $\mathbf{A}(\mathbf{x}, \mathbf{x}')$ of equation (2.5) is provided in the form of a table via an auxiliary input-file. This is part of the extensions of CONTACT for conformal contact situations [49]. The idea is illustrated in Figure 5.3, which shows finite element meshes used for computing the response of wheel and rail when the contact moves to the rail gauge corner. A similar strategy may be used in other scenarios, for instance for the computation of contact problems involving layered materials.

This option is activated by selecting $\mathbf{C}_3 = 9$ (page 32). An additional line of input is then required:

CFNAME	$[-]$	Filename for the input-file with numerically computed influence coefficients. The structure of this file is documented separately.
--------	-------	--

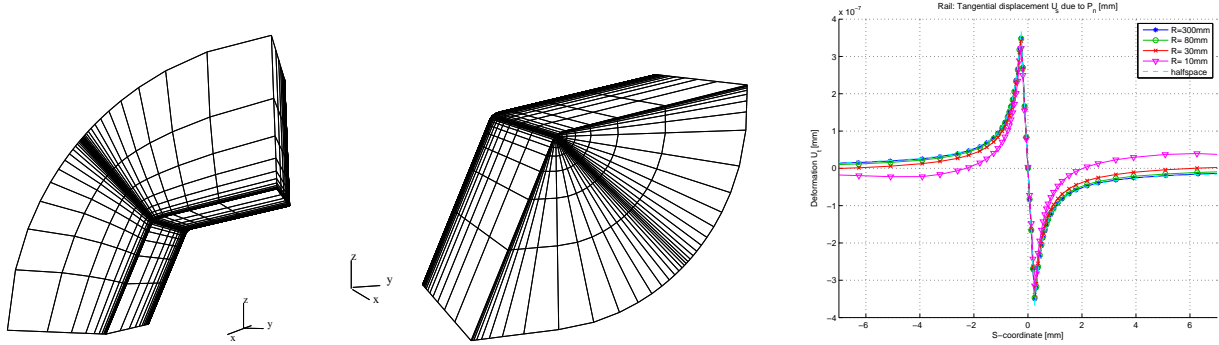


Figure 5.3: *Left/middle: finite element meshes used for numerical calculation of influence coefficients. Right: tangential response of the rail to a localized normal load for different radii of curvature, illustrating deviations from the half-space approach.*

Numerically calculated influence coefficients can be combined with material models $\mathbf{M} = 0$ (Section 5.1.1) and $\mathbf{M} = 4$ (Section 5.1.4). On the one hand, these material models will be used as “best guess” when subsurface stresses are requested. On the other hand this allows for using the interfacial layer model in conformal contact situations too.

5.2 Friction description

The basic friction law that is used by the program is dry, Coulomb friction with two constant friction coefficients, for points in the slip and adhesion areas. This option is obtained when $L = 0$.

$$\mu(\mathbf{x}, t) = \begin{cases} \mu_{stat} & \text{where } s_a = V \|\mathbf{s}_t(\mathbf{x}, t)\| = 0 \text{ (adhesion)}^2, \\ \mu_{kin} & \text{where } s_a > 0 \text{ (slip area)}. \end{cases} \quad (5.9)$$

FKIN [–] Kinetic coefficient of friction $\mu_{kin} > 0$.

FSTAT [–] Static coefficient of friction $\mu_{stat} \geq \mu_{kin}$.

Other friction laws, particularly concerning slip-velocity dependent friction, have been implemented too. These are illustrated in Figure 5.4. Results are presented in the examples in Sections 6.4 and 6.12.

When $L = 2$, a linear decrease of the friction coefficient is used, until a certain minimum is reached:

$$\mu_s(s_a) = \mu_{kin} + \mu_{lin1} \cdot \max(0, 1 - s_a/2s_{h1}) + \mu_{lin2} \cdot \max(0, 1 - s_a/2s_{h2}) \quad (5.10)$$

Here $s_a(\mathbf{x}, t)$ is the magnitude of the absolute slip velocity (≥ 0) at position \mathbf{x} .² μ_{lin1} , μ_{lin2} , s_{h1} and s_{h2} are the coefficients of the friction law. Two terms are provided for flexibility, and

²In the calculation of shifts ($T = 1$) the absolute slip velocity s_a is computed as $S_a/\delta t$, with S_a the slipped distance [mm] and $\delta t = 1$ [s].

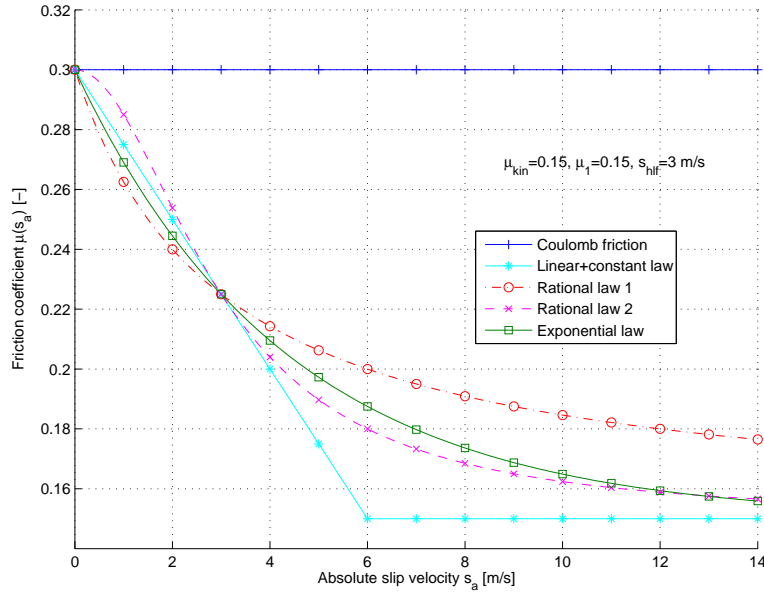


Figure 5.4: Dependence of friction coefficient μ on absolute slip velocity s_a for the friction laws of equations (5.10)–(5.12).

allow two different time-scales to be incorporated. μ_{lin1} and μ_{lin2} are the maximum sizes of the linear terms (at $s_a = 0$, $\mu_{stat} = \mu_{kin} + \mu_{lin1} + \mu_{lin2}$). Setting one of these to zero disables a term. s_{h1} and s_{h2} are the slip velocities at which the initial contributions are halved.

FKIN	[–]	Limit value $\mu_{kin} > 0$ for the coefficient of friction for large slip velocities, used when L = 2 – 4.
SABSH1	[mm/s]	Absolute slip velocity $s_{h1} > 0$ for which the size of a term is halved compared to $s_a = 0$. Used when L = 2 – 4.
SABSH2	[mm/s]	Absolute slip velocity $s_{h2} > 0$ for which the size of a term is halved compared to $s_a = 0$. Used when L = 2 – 4.
FLIN1	[–]	Coefficient μ_{lin1} of (5.10), used when L = 2.
FLIN2	[–]	Coefficient μ_{lin2} of (5.10), used when L = 2.

When L = 3, a decrease of the friction coefficient is used described by a so-called rational formula:

$$\mu_s(s_a) = \mu_{kin} + \frac{\mu_{rat1}}{1 + s_a/s_{h1}} + \frac{\mu_{rat2}}{1 + (s_a/s_{h2})^2} \quad (5.11)$$

The two terms can again be configured independently. The parameters μ_{rat1} and μ_{rat2} control their sizes, and s_{h1} and s_{h2} the rate of decay.

FRAT1	[–]	Coefficient μ_{rat1} of (5.11), used when L = 3.
FRAT2	[–]	Coefficient μ_{rat2} of (5.11), used when L = 3.

Finally, when $L = 4$, an exponential decrease of the friction coefficient is used:

$$\mu_s(s_a) = \mu_{kin} + \mu_{exp1} \exp(-\log(2) \cdot s_a/s_{h1}) + \mu_{exp2} \exp(-\log(2) \cdot s_a/s_{h2}) \quad (5.12)$$

The factor $\log(2)$ makes that s_{h1} and s_{h2} are again the absolute slip velocities in which the size of the terms is halved compared to $s_a = 0$.

FEXP1	[−]	Coefficient μ_{exp1} of (5.12), used when $L = 4$.
FEXP2	[−]	Coefficient μ_{exp2} of (5.12), used when $L = 4$.

Note that equation (5.12) incorporates Polach’s exponential formula [25],

$$\mu_s(s_a) = \mu_{stat} ((1 - A)e^{-Bs_a} + A). \quad (5.13)$$

For given A and B , one may simply set $\mu_{kin} = A\mu_{stat}$, $\mu_{exp1} = (1 - A)\mu_{stat}$, $s_{h1} = \log(2)/B$ and use $\mu_{exp2} = 0$. Note that B should be given with unit $[s/mm]$, $1000\times$ smaller than the reference values provided in [25], with unit $[s/m]$.

The friction laws (5.10)–(5.13) lead to unwanted effects in transient calculations when they are based on the instantaneous slip velocity s_a at time t . This is illustrated in the example in Section 6.12 and analysed further in [48]. The point is that the friction coefficient changes abruptly at the transition from the adhesion to the slip area, see the line “ $d_c = 0$ ” in Figure 5.5.

In order to circumvent these unwanted effects, the coefficient μ_s is not applied directly but via a relaxation process. The actual friction coefficient $\mu(\mathbf{x}, t)$ tends towards the target value $\mu_s(s_a(\mathbf{x}, t))$, but also has a memory for the previous values of μ and s_a :

$$\dot{\mu}(\mathbf{x}, t) = -\frac{\max(s_a(\mathbf{x}, t), s_0)}{d_c} (\mu(\mathbf{x}, t) - \mu_s(s_a(\mathbf{x}, t))) \quad (5.14)$$

This form implies that the transient behavior consists of an exponential decay towards the steady state value μ_s . d_c is the characteristic sliding distance over which the adaptation occurs (typically in the order of μm). Its effect is illustrated in Figure 5.5, showing the actual friction coefficient $\mu(\mathbf{x}, t)$ for a steady rolling cylinder. The coefficient s_0 is a small velocity ($\mathcal{O}(mm/s)$) that allows the friction coefficient to change in the adhesion area (where $s_a = 0$), if it comes from a different value than the static coefficient $\mu_{stat} = \mu_s(0)$.

MEMDST	[mm]	Characteristic distance d_c for the friction memory effect (≥ 0). An instantaneous friction law is obtained by setting $d_c = 0$.
MEM_S0	[mm/s]	Minimum velocity s_0 for the friction memory effect (≥ 0).

In cases where $L = 1$, the friction law and coefficients of the previous case are maintained.

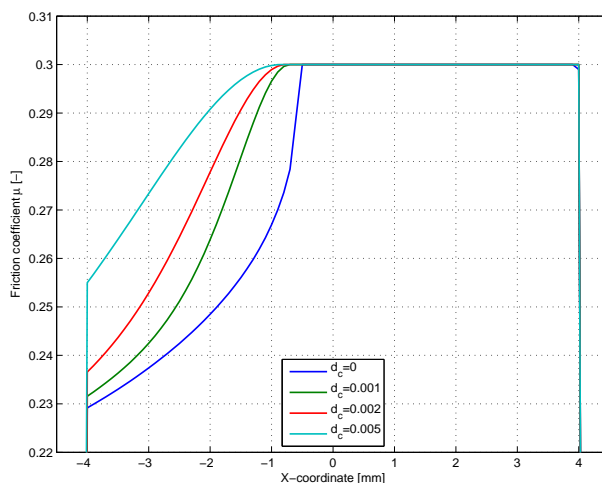


Figure 5.5: Typical effect of friction memory (equation (5.14)) on the friction coefficient μ : gradual instead of abrupt change of $\mu(\mathbf{x}, t)$.

5.3 Potential contact area and discretisation

The potential contact area is a rectangular area aligned with the x - and y -axes, $(x, y) \in [x_l, x_h] \times [y_l, y_h]$. It is divided into $m_x \times m_y$ rectangular elements of size $\delta x \times \delta y$, see Figure 5.6.

MX, MY $[-]$ The number of discretisation elements in x - and y -directions respectively, number of columns and rows of the discretisation grid.

NPOT $[-]$ Total number of discretisation elements, $NPOT = MX \cdot MY$.

The elements are numbered using two-dimensional indices $(ix, iy) \in \{1 \cdots m_x\} \times \{1 \cdots m_y\}$, as well as with a one-dimensional index

$$I = ix + (iy - 1) \cdot m_x. \tag{5.15}$$

The centers of the elements are given by:

$$\mathbf{x}_I = \left[x_l + (ix - \frac{1}{2}) \cdot \delta x, y_l + (iy - \frac{1}{2}) \cdot \delta y \right]^T \tag{5.16}$$

Note: the potential contact area should be somewhat larger than the true contact area:

1. The solver SteadyGS for the steady state rolling problem requires one “exterior” element at the trailing edge of the contact area. If there is an interior element in the first grid column ($ix = 1$), SteadyGS will not be used, the slower and less robust ConvexGS will be used instead.

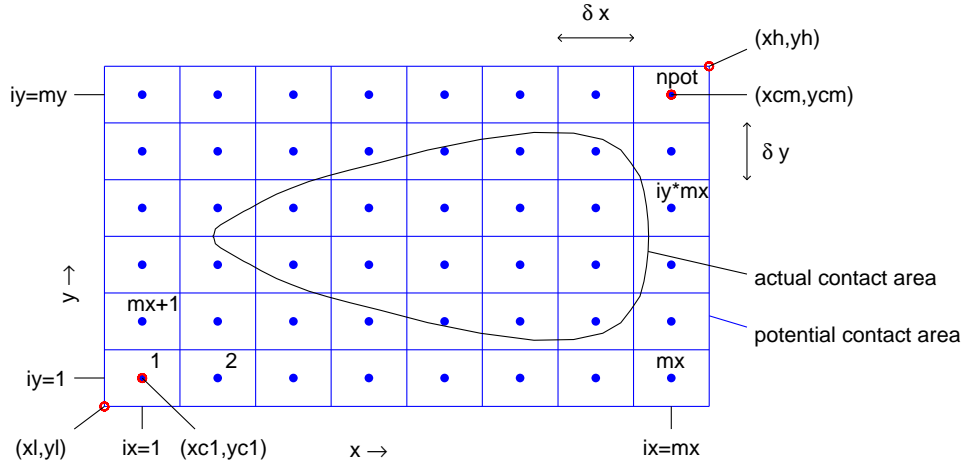


Figure 5.6: *The potential contact area, its discretisation, and the numbering of the elements.*

2. To properly estimate the displacement difference \mathbf{u}_t at the leading edge requires two “exterior” elements there too (last two grid columns, $mx - 1$ and mx). In grid rows where these elements are not available, the so called leading edge correction will be switched off.

It is recommended to use two additional grid rows and columns around the actual contact area at all sides.

The location and discretisation of the potential contact area can be specified in a number of ways.

IPOTCN — Integer flag. Negative values: Hertzian options, positive values: direct specification of potential contact area by the user.

5.3.1 Hertzian geometries

The variables in this section describe the prescription of the geometry of the bodies when the problem is Hertzian. The potential contact area is then derived from the Hertzian solution.

AA, BB [mm] The semi-axes a, b of the contact ellipse.
 A1, B1 [mm⁻¹] The curvatures A, B in x - and y -direction.

The curvatures are related to the effective radii of curvature of the two bodies by

$$A1 = \frac{1}{2R_{x,eff}^{(1)}} + \frac{1}{2R_{x,eff}^{(2)}}, \quad B1 = \frac{1}{2R_y^{(1)}} + \frac{1}{2R_y^{(2)}}. \quad (5.17)$$

In wheel-rail contact, the effective rolling radius for the wheel is $R_{x,eff} = R_w(y)/\cos(\delta)$ [45]. That is, the contact angle δ makes the surface look flatter in rolling direction. This is

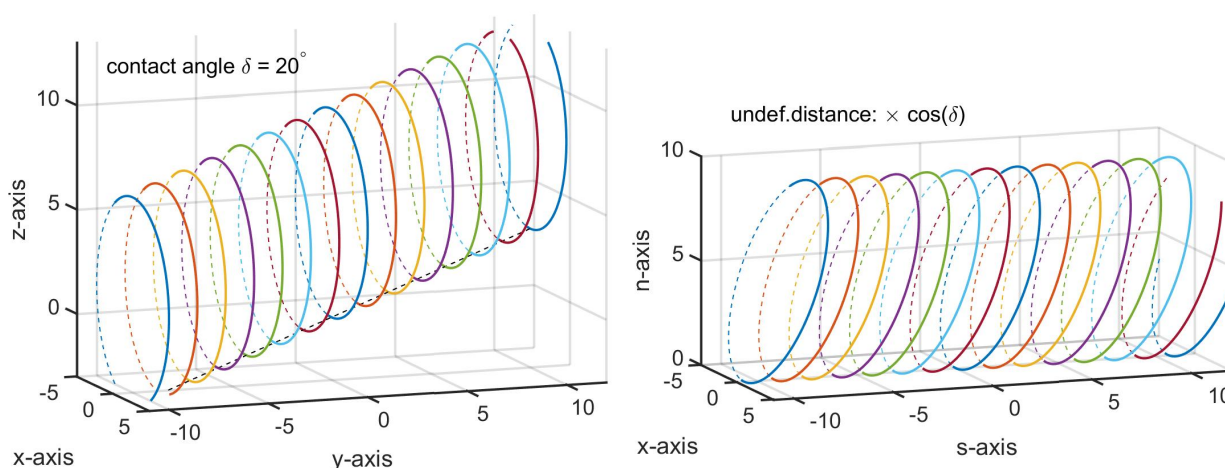


Figure 5.7: *Left: vertical sections of a wheel in global $Oxyz$ -coordinates, circles of almost the same radius $R_w(y)$. Right: rotated to contact local $Oxs-n$ -coordinates, all circles touching at $n = 0$. The effective radius becomes $R_{x,eff} = R_w(y) / \cos(\delta)$ [45].*

illustrated in Figure 5.7 using constant wheel radius R_w . Since the actual radius R_w is much bigger than shown in the figure, its variation with y is a second order phenomenon.

These values may be prescribed in the input in different combinations:

- IPOTCN = -1: curvatures A1, B1 specified;
- IPOTCN = -2: curvature A1 and aspect ratio AA/BB specified;
- IPOTCN = -3: semi-axes AA, BB specified.

Other variables used in the Hertzian problem are the material constants (Section 5.1) and either the approach PEN or the normal force FN, see Section 5.5.

When you have specified sufficient parameters, the other parameters can be computed from the Hertzian solution. This is done by the program. It is then possible to take as the potential contact area the rectangle $[-AA, AA] \times [-BB, BB]$.

SCALE [-] Scale parameter for the potential contact area.

The potential contact area becomes the rectangle $[-AA \cdot s, AA \cdot s] \times [-BB \cdot s, BB \cdot s]$ with $s = \text{SCALE}$. Choose **SCALE** = 1.1 for a potential contact area that is 10% larger than the actual Hertzian contact area.

In order to add two grid rows and columns around the actual contact area to the potential contact area, you may use **SCALE** = $\text{MX} / (\text{MX} - 4)$. Here (MX,MY) is the size of the potential contact area, (MX - 4,MY - 4) the box encompassing the actual contact area.

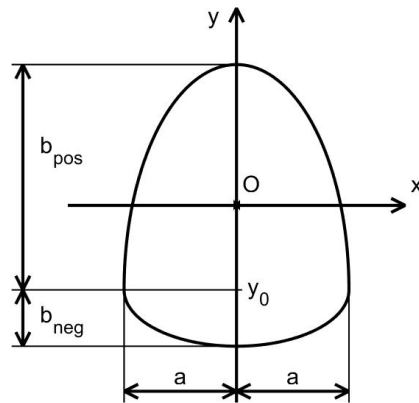


Figure 5.8: *Definitions for the double half-elliptical contact region of the SDEC approach.*

In non-quasiidentity, the true contact can fall outside the Hertzian ellipse. Therefore a larger potential contact area should be used. This is achieved by choosing **SCALE** > 1 in those cases. Using the **SCALE** parameter, you can also circumvent having to re-compute the discretisation later on. For instance when you want to compute a sequence of cases where the normal load is gradually increased, you can already in the first case achieve a potential contact area that is sufficiently large for the final case. For instance setting **SCALE** = 1.1 allows for a normal load that is $1.1^3 = 1.331$ as large as the initial load. Do not choose **SCALE** < 1 , because then the true contact area will fall partly outside the potential contact area, i.e. not all tractions will be taken into account.

5.3.2 SDEC: simple double half-elliptical contact area

The variables in this section allow to create a contact area that consists of two half-ellipses with different semi-axes according to the SDEC approach [24]. The dimensions regarding the two half ellipses are illustrated in Figure 5.8. Note that the origin of the contact coordinates is chosen differently than in [24], as discussed in [44].

AA	[mm]	Semi-axis a of the two half ellipses in x -direction.
BNEG, BPOS	[mm]	The semi-axes b_{neg}, b_{pos} of the two half contact ellipses in y -direction.

This uses option **IPOTCN** = -4 for the potential contact area specification. The pressure distribution is then prescribed using option **B** = 4, using a Hertzian-like formula in each strip

$y = \text{const}$:

$$b = \frac{b_{pos} + b_{neg}}{2}, \quad \psi = \frac{b_{pos} - b_{neg}}{2b}, \quad y_0 = -b\psi, \quad (5.18)$$

$$y - y_0 > 0 : x_{l,pos} = a \sqrt{1 - \frac{(y - y_0)^2}{b^2(1 + \psi)^2}}, \quad (5.19)$$

$$y - y_0 < 0 : x_{l,neg} = a \sqrt{1 - \frac{(y - y_0)^2}{b^2(1 - \psi)^2}}, \quad (5.20)$$

$$p_{max} = \frac{3}{2} \frac{F_n}{\pi ab}, \quad p_n(x, y) = \frac{p_{max}}{a} \sqrt{x_l^2(y) - x^2}. \quad (5.21)$$

The precise form of the undeformed distance is then constructed from the pressure profile.

Other variables used in the SDEC approach are the material constants (Section 5.1), the normal force F_N , see Section 5.5, and the **SCALE** parameter of the Hertzian approach (Section 5.3.1

5.3.3 Direct specification of the potential contact area

The parameters related to the potential contact area and its discretisation are (refer to Figure 5.6):

DX, DY	$[mm]$	Sides $\delta x, \delta y$ of each element.
XL, YL	$[mm]$	Coordinates (x_l, y_l) of the lower left corner of the potential contact area.
XH, YH	$[mm]$	Coordinates (x_h, y_h) of the upper right corner of the potential contact area.
$XC1, YC1$	$[mm]$	Coordinates of the center of element $(1, 1)$ of the potential contact area.
XCM, YCM	$[mm]$	Coordinates of the center of element (mx, my) of the potential contact area.

Different options are available, as indicated by variable **IPOTCM**.

- **IPOTCM** = 1: coordinates **XL, YL** specified together with stepsize **DX, DY**;
- **IPOTCM** = 2: coordinates **XL, YL** specified together with coordinates **XH, YH**;
- **IPOTCM** = 3: coordinates **XC1, YC1** specified together with stepsize **DX, DY**;
- **IPOTCM** = 4: coordinates **XC1, YC1** specified plus **XCM, YCM**. (Note: not allowed when **MX** = 1 or **MY** = 1.)

5.4 Non-Hertzian geometry specification

In Hertzian problems the geometry is specified by the loading parameters and curvatures, see Section 5.3.1. In non-Hertzian cases the distance $h(\mathbf{x})$ between the undeformed surfaces of the two bodies must be specified. For this different parametrisations are available.

IBASE – Integer flag. Form of undeformed distance.

- IBASE = 1: undeformed distance is quadratic in (x, y) .
- IBASE = 2: the bodies are circular in x , the profile in y -direction is specified.
- IBASE = 3: the profile is quadratic plus the difference of two sines in x -direction.
- IBASE = 9: a general profile is used, the undeformed distance is specified explicitly for all elements.

When IBASE = 1 the following formula is used:

$$h(x, y) = b_1 x^2 + b_2 x y + b_3 y^2 + b_4 x + b_5 y + b_6 \quad (5.22)$$

B(1:3) $[mm^{-1}]$ Coefficients of quadratic terms.

B(4:5) $[-]$ Coefficients of linear terms.

B(6) $[mm]$ Coefficient of constant term.

Option IBASE = 2 is intended for non-Hertzian rolling of a body of revolution. The axle of the body is parallel to the y -axis at x -coordinate x_m . The effective radius of curvature in x -direction is R_m . The profile “ $p(y)$ ” in y -direction is arbitrary. It is specified at a regular spacing, which may be different from the grid sizes used in the potential contact area. Linear inter- or extrapolation is used to get the profile at other y -coordinates. The formula used for the undeformed distance is

$$h(x, y) = p(y) + \frac{(x - x_m)^2}{2 R_m} \quad (5.23)$$

NN – Number of profile points in y -direction.

XM $[mm]$ x -coordinate of the axis of the body of revolution.

RM $[mm]$ Effective radius of curvature of the surface in rolling direction.

Y1 $[mm]$ Lowest y -coordinate y_1 for which the profile is specified.

DY1 $[mm]$ Increment of y -coordinates between successive points y_k and y_{k-1} .

B(1:NN) $[mm]$ Profile heights $p(y_k)$ above $z = 0$ for the sample points y_k , $k = 1 \dots NN$.

Note that the specification of the profile here has nothing to do with the choice of the potential contact area. More specifically: the number of heights NN is independent of the number of rows MY . The profile will be determined by interpolation between the successive heights.

Note further that the effective rolling radius can be different from the vertical height of the axle above the plane. The undeformed distance is measured in normal direction. For a wheel with radius R and a contact angle δ , the effective radius of curvature is $R/\cos\delta$, see Section 5.3.1 (Figure 5.7) and [15, eq. (1.61b)].

The formula used with $IBASE = 3$ is the following.

$$h(x, y) = b_1 \sin(b_2(x - b_3)) - b_4 \sin(b_5(x - b_6)) + x^2/b_7 + y^2/b_8 \quad (5.24)$$

This implies that b_7 and b_8 are radii of curvature ($[mm^{-1}]$), b_1 and b_4 are vertical distances ($[mm]$), b_3 and b_6 are horizontal positions where the sines are zero ($[mm]$), and b_2 and b_5 are frequencies ($[rad/mm]$).

When $IBASE = 9$, the values $h(\mathbf{x}_I)$ are specified for all elements I .

H(I) $[mm]$ Undeformed distance at center of element I .

Refer to (5.15) and (5.16) for the numbering and coordinates of the elements.

The specification of the undeformed distance may be completed by the application of a so-called *planform*. Elements that lie outside the planform will get a very large undeformed distance so that they cannot enter the contact area.

IPLAN – Integer flag. Type of planform.

- IPLAN = 1: Unrestricted planform, all elements of the potential contact area may enter the actual contact area.
- IPLAN = 2: Quadratic planform. Elements are excluded from the contact area when a quadratic function $pl(x, y)$ is ≥ 0 . The function pl is specified by six parameters analogously to $h(x, y)$ in (5.22).
- IPLAN = 3: Union of two rectangles, specified by eight parameters.

$$[x_l^{(1)}, x_h^{(1)}] \times [y_l^{(1)}, y_h^{(1)}] \cup [x_l^{(2)}, x_h^{(2)}] \times [y_l^{(2)}, y_h^{(2)}] \quad (5.25)$$

The parameters are specified in the order in which they occur in this formula.

5.5 Kinematic variables

The normal problem is specified either using the approach or the total normal force, depending on the N_3 -digit (page 35).

PEN	[<i>mm</i>]	Approach δ_n of the two bodies. Constant offset to the profile specification $h(x, y)$: the total undeformed distance is $h(x, y) - \delta_n$. If the minimum of $h(x, y)$ is 0, δ_n is the maximum penetration of the undeformed surfaces.
FN	[<i>N</i>]	Total normal force in the contact area.

5.5.1 Tangential quantities in case of a shift ($T = 1$)

In the calculation of a shift ($T = 1$), the following variables are used to characterize the relative motion of the two bodies:

CKSI, CETA	[<i>mm</i>]	Rigid shift, i.e. the displacement of the two bodies' axles in x - and y -directions respectively in a time step δt .
CPHI	[<i>rad</i>]	Rotation shift, angular displacement of the two bodies around the z -axis in a time step δt .

When $E_3 = 9$ an additional term is added:

EXRHS(I, :)	[<i>mm</i>]	Extra term in the rigid shift of the elements, specified explicitly by the x - and y -components for each element (order: x, y for element 1, x, y for element 2, ..., e.g. using one line per element.)
-------------	---------------	--

The rigid shift of all elements is computed with

$$\mathbf{W}_{It} = [\text{CKSI} - \text{CPHI} \cdot y_I + \text{EXRHS}_{Ix}, \text{CETA} + \text{CPHI} \cdot x_I + \text{EXRHS}_{Iy}]^T \quad (5.26)$$

The true time step size δt is unknown to the program. It is set to 1 *s*, and the corresponding variables are set to $\text{CHI} = 0^\circ$, $\text{DQ} = 1 \text{ mm}$ and $\text{VELOC} = 1 \text{ mm/s}$. This way, a shift of 3 *mm* corresponds to an absolute slip velocity of 3 *mm/s* and a relative slip velocity of 3.

5.5.2 Tangential quantities in case of rolling ($T = 2 - 3$)

In rolling problems ($T = 2, 3$) the following variables are used.

CHI	[rad]	Rolling direction χ . This may be given in degrees using the notation 180d , which is converted to [rad] and displayed as such in the output-file. The value should be near 0 or π (180°), i.e. rolling in positive or negative x -direction. It is not used in shifts.
DQ	[mm]	Rolling distance traversed per timestep, i.e. $\delta q = V \cdot \delta t$. This should preferably be of the order of the grid size DX [55].
VELOC	[mm/s]	The rolling velocity V . This affects viscoelastic material behaviour (M = 1), velocity dependent friction laws (L = 2 – 4), and the frictional power dissipation FRIC .
CKSI, CETA	[–]	Creepages in x - and y -directions respectively, rigid slip velocities relative to the rolling speed V .
CPHI	[rad/mm]	Spin creepage, angular velocity of the two bodies around the z -axis relative to the rolling speed V .

When **E** = 9 an additional term is added which is particularly relevant for conformal contact situations:

EXRHS(I, :)	[–]	Extra term in the rigid slip of the elements, specified explicitly by the x - and y -components for each element (e.g. using one line of input per element.)
-------------	-----	--

In this case the rigid shift is computed from the function

$$\mathbf{W}_{It} = \text{DQ} \cdot \begin{bmatrix} \text{CKSI} - \text{CPHI} \cdot (y_I + sn \cdot \delta q/6) + \text{EXRHS}_{Ix} \\ \text{CETA} + \text{CPHI} \cdot (x_I + cs \cdot \delta q/6) + \text{EXRHS}_{Iy} \end{bmatrix} \quad (5.27)$$

The additional terms $(sn, cs) \cdot \delta q/6$ compensate for the rolling distance traversed in a time step, along rolling direction **CHI**.

The total forces in x - and y -directions are defined as follows.

FX, FY	[–]	Total tangential forces in x - and y -directions, divided by the static traction bound FSTAT · FN .
--------	-----	---

These can be both output (**F** = 0, 1) and input (**F** = 1, 2) of the program. Note that a problem with creepages prescribed is easier to solve than a problem with total forces prescribed. An additional (Newton-Raphson) iteration process is used when total forces are prescribed, to determine the appropriate values for the creepages. This process is not fully reliable when the total forces are close to the maximum values that can be attained (full sliding solution).

5.6 Iteration constants

The different iteration processes that are used are described in Section 2.8. These processes are terminated when the required accuracy is reached, or when a prescribed maximum number of iterations is exceeded.

MAXOUT	–	Maximal number of iterations for the outer-loop, the Panagiotopoulos process. In quasi-identical cases (difference parameter $AK = 0$) Johnson’s process is used with $MAXOUT = 1$. Otherwise a value of 20 is usually sufficient.
MAXIN	–	Maximal number of iterations for the active set algorithms NORM and TANG. Usually a small value such as 20 suffices, a larger value may be needed when using a slip velocity dependent friction law ($L = 2 - 4$), which does not have a separate iteration counter.
MAXNR	–	Maximal number of iterations for the Newton-Raphson procedures that are used when total forces are prescribed. A value of 10 or 20 is usually ok.
MAXGS	–	Maximal number of iterations in iterative procedures NormCG, TangCG, ConvexGS and SteadyGS. This may be set to a high value such as 500 or 999.

When one of the iteration constants MAXOUT, MAXIN or MAXNR is reached, it is assumed that the process does not converge for the problem at hand. This results in an error stop, because the next case cannot always be computed properly. Reaching MAXGS in the iterative solvers does not immediately result in an error stop. When the error has been reduced during the process we continue, if it grows the program is stopped.

EPS	[–]	Requested relative accuracy of output quantities. This constant is used in many places in stop-criteria.
-----	-----	--

The typical stop-criterion used is

$$\|\mathbf{p}^k - \mathbf{p}^{k-1}\|_{rms} < EPS \cdot \|\mathbf{p}^k\|_{rms} \quad (k: \text{iteration counter}). \quad (5.28)$$

Suitable modifications are made for the case that $\mathbf{p}^k \equiv 0$. Note that this criterion does not guarantee small errors in the final results. Errors of other iteration procedures (Panagiotopoulos, Newton-Raphson) are compounded. The criterions that are used there can be found in the flow trace (see description of the W-digit in Section 3.3).

Two warnings that are related to the iteration accuracy EPS are the following:

NORM: WARNING. There are **** elements with small deformed distance and

**** elements with small pressure.

TANG: WARNING. There are **** elements with small slip and **** elements with tractions close to the traction bound.

These warnings indicate that the element divisions between C and E (NORM, interior and exterior elements) and between H and S (TANG, adhesion and slip areas) may be affected by inaccuracies. Particularly the former one should not be ignored. It indicates that

- elements in the contact area with normal pressure $p_n < \epsilon$ might actually belong to the exterior, and
- exterior elements with small deformed distance might better be interior elements.

The estimates of the element divisions become more reliable if the iteration process is continued and the approximation error is reduced, i.e. when a smaller tolerance EPS is used. Similar considerations hold for the latter warning regarding the slip and adhesion areas. In that case the accuracy assessment is on the pessimistic side; the results are usually more reliable than the warning suggests.

OMEGAH	[−]	Relaxation parameter ω_h for ConvexGS and SteadyGS for elements in the adhesion area.
OMEGAS	[−]	Relaxation parameter ω_s for ConvexGS and SteadyGS for elements in the slip area.
INISLP	[−]	Flag for initial estimate for slip velocity. Used in slip velocity dependent friction laws ($L = 2 - 4$), else INISLP = 0 can be used.
OMGSLP	[−]	Relaxation parameter ω_{slp} for the slip velocity. Used in the iteration procedure for slip velocity dependent friction laws ($L = 2 - 4$), ignored otherwise.

Relaxation is an internal feature of the Gauss-Seidel based solvers ConvexGS and SteadyGS, see [35] and [36]. The program contains suitable default values for these parameters that are used in the default case when $G = 0$. Fine-tuning is possible by setting the G -digit to 2 or 3.

Our experience is that TangCG should be used for shifts and transient rolling ($T = 1, 2$) and SteadyGS for steady state rolling ($T = 3$). This is the default choice when $G = 0$. Deviation from the defaults is needed only in specific circumstances.

- Large problems may benefit from reducing ω_h, ω_s ;
- Small step sizes $c = \delta q / \delta x \ll 1$ may need reducing ω_h, ω_s in order to work well [36].

When using velocity-dependent friction, there may exist multiple valid solutions to a case [5, 48]. The initial estimate for the slip-velocity then determines which one of the possible solutions is found. Three different strategies are provided. When $\text{INISLP} < 0$, the slip velocity is approximated from below, such that the “low-slip” solution is found. When $\text{INISLP} > 0$, the slip velocity is approximated from above and the “high-slip” solution is obtained. Finally the choice $\text{INISLP} = 0$ is in between and uses the slip velocity of the previous case as initial estimate.

5.7 Global output quantities

Global output quantities are written to the output file `<experim>.out` when the 0-digit is set to values 1 – 4.

The global part of the output firstly contains a description of the problem that is solved ($0 \geq 2$), its control digits and the primary input values that are used (except discretisation and geometry). Then several aggregate output quantities are displayed: the total forces or creepages as determined by the program, the torsional moment, elastic energy and frictional power. Finally sensitivities computed by the Newton-Raphson processes may be displayed, statistics about the element division and iterations and a picture of the contact area.

5.7.1 Output in rolling problems (T = 2 – 3)

The approach PEN, creepages CKSI, CETA and total forces FN, FX and FY were already described in Section 5.5. These may be input or output depending on the N- and F-digits.

MZ	[N mm]	Torsional moment M_z around the z -axis of the local coordinate system.
ELEN	[J]	Elastic energy.
FRIC	[W]	Frictional power dissipation by the surface tractions.

The formulae used for these quantities are:

$$\text{MZ} = \delta x \delta y \sum_I (p_{Iy} x_I - p_{Ix} y_I), \quad (5.29)$$

$$\text{ELEN} = \frac{\delta x \delta y}{2 \cdot 1000} \sum_{i \in \{x,y,n\}} \sum_I p_{Ii} u_{Ii}, \quad \text{FRIC} = \frac{V}{1000} \cdot \delta x \delta y \sum_{\alpha \in \{x,y\}} \sum_I p_{I\alpha} s_{I\alpha} \quad (5.30)$$

Note that $s_{I\alpha}$ is the relative slip velocity, hence the multiplication with V . Note: the elastic energy is not computed when $\text{B} = 2$ or 3 is used, since the normal displacements u_{In} are then not available. A zero is displayed in the output instead.

Note: the frictional power computed by CONTACT suffers from a substantial discretisation error. In certain cases the following macroscopic quantity may be used as an alternative:

$$\text{FRIC} = V \cdot 10^{-3} \cdot (\text{CKSI} \cdot F_x + \text{CETA} \cdot F_y + \text{CPHI} \cdot M_z), \quad (5.31)$$

with F_x, F_y here being the total tangential forces in $[N]$. This alternative is valid only in steady state rolling, when the two bodies are elastically similar (quasi-identical, $K = 0$ in equation (5.2)), and when the creepages are constant throughout the contact patch ($\mathbf{E}_3 = 0$). This is illustrated for instance in [22].

The sensitivities are calculated when the Newton-Raphson loop is used for prescribed total forces. This is currently only the case when $\mathbf{F} = 1$ or $\mathbf{F} = 2$. When a sensitivity has not been computed, a zero is displayed in the output.

DFX/DKSI	$[-]$	Sensitivity of relative tangential force FX to a change of creepage CKSI (Section 5.5).
----------	-------	---

The sensitivities of FX to CETA and of FY to CKSI and CETA are likewise defined.

5.7.2 Output in case of shifts ($\mathbf{T} = 1$)

In case of a tangential shift, there are some differences compared to the output of rolling problems described above.

FRIC	$[J]$	Frictional work of the surface tractions.
DFX/DKSI	$[1/mm]$	Sensitivity of relative tangential force FX to a change of rigid shift CKSI.

In this case the true velocity V and time step δt are unknown to the program. The shift distance \mathbf{S}_{It} in the step is used instead of the velocity:

$$\text{FRIC} = \delta x \delta y \sum_{\alpha} \sum_I p_{I\alpha} S_{I\alpha}. \quad (5.32)$$

5.8 Main solution arrays

The detailed output of quantities inside the contact area for $\mathbf{0} = 4$ is rather crude, but may be beneficial for inspection of the results for small problems.

IGS(I)	–	Element division of current time instant, stated per element I (0 =Exterior, 1 =Adhesion, 2 =Slip). Refer to (5.15) and (5.16) for the numbering and coordinates of the elements.
PS(I, 1)	$[N/mm^2]$	The normal pressures p_{In} of the current time step in element I , acting on body 1, the upper body.
PS(I, 2:3)	$[N/mm^2]$	The tangential tractions \mathbf{p}_{It} per element I , load per unit area acting on body 1.
PV(I, 1:3)	$[N/mm^2]$	The tractions \mathbf{p}'_{It} of the previous time instant.
US(I, 1:3)	$[mm]$	Displacement difference \mathbf{u}_I per element I in normal and tangential directions. Note: the normal displacement difference is not computed when $B = 2$ or 3 ; in that case zeros are displayed in the output instead.

The quantities that are printed in the output are:

X, Y	$[mm]$	coordinates \mathbf{x}_I of the center of each element I , cf. (5.16).
H-PEN	$[mm]$	True undeformed distance $h(\mathbf{x}_I) - \delta_n$, taking into account the approach δ_n .
PN	$[N/mm^2]$	Normal pressure $PS(I, 1) = p_{In}$.
TRCBND	$[N/mm^2]$	Traction bound $\mu_I \cdot p_{In}$ in the slip area.
ABS(PT)	$[N/mm^2]$	Magnitude of tangential tractions $\ \mathbf{p}_{It}\ $.
ARG(PT; -S)	$[deg]$	Direction of tangential tractions $\arg(\mathbf{p}_{It})$.

In rolling contact problems ($T = 2 - 3$) the relative slip velocity is displayed:

ABS(S)	$[-]$	Magnitude of the relative slip velocity $\ \mathbf{s}_{It}\ $.
RIG.SLIP	$[-]$	x - and y -components of the relative rigid slip velocity \mathbf{w}_{It} .

In the computation of a shift ($T = 1$), the shift distance is displayed instead:

ABS(S)	$[mm]$	Magnitude of the shift distance $\ \mathbf{S}_{It}\ $.
RIG.SHFT	$[mm]$	x - and y -components of the rigid shift distance \mathbf{W}_{It} .

5.9 Subsurface stresses

The calculation of subsurface stresses is activated by the control digit **S** (Section 3.3). This computes the subsurface displacements $u_i(x, y, z)$, displacement gradients $\partial u_i / \partial x_j$ and interior stresses $\sigma_{ij}(x, y, z)$, i.e. the stress tensor $\boldsymbol{\sigma}$. From this it derives stress invariants like the mean

hydrostatic stress σ_{hyd} , the equivalent tensile stress σ_{vm} of the von Mises criterion, and the principal stresses $\sigma_1, \sigma_2, \sigma_3$ and maximum shear stress $\sigma_{tresca} = \sigma_1 - \sigma_3$ used in the Tresca criterion.

Values can be computed for a large grid of points independent of the surface contact area. Output can be requested for the maximum values in this grid and for the values in each grid point separately.

5.9.1 Control digits for the subsurface stress calculation

The input of the subsurface points in `<experim>.inp` starts with the control digits A_s and O_s , governing the level of output of subsurface stresses to the Matlab-file `<experim>.<case>.subs` and the to the out-file.

A_s - MATFIL, subsurface stress : governs the use of the Matlab-file `<experim>.<case>-.subs`, in cases where the subsurface stress calculation is used ($S \geq 1$):

0 - the `subs`-file is not created;

1 - the displacements and stress invariants in subsurface stress points are written to a Matlab-file `<experim>.<case>.subs`;

2 - additionally, all components of the stress tensor are written to the `subs`-file.

O_s - OUTPUT, subsurface stress : governs the extent of the output to the output-file `<experim>.out`:

0 - no results are printed to the output-file (values *are* computed, and stored in internal memory for use in other calculations);

1 - minimum output is printed, just the maximum values of primary stress invariants;

2 - the maximum values are printed for additional invariants, e.g. the principal stresses;

3 - not used;

4 - the detailed results for the sub-surface (deformations, stresses) are printed as well, in addition to 2.

These control digits are read from the input-file when $S = 2$ or $S = 3$. The same value is used for all blocks of subsurface points.

5.9.2 Input for the subsurface stress calculation

The locations $[x, y, z]^T$ for the subsurface stress calculation are specified in “blocks” of $NX \cdot NY \cdot NZ$ points, for coordinates `X(1:NX)`, `Y(1:NY)` and `Z(1:NZ)`. Multiple blocks may be used

consecutively. Each block starts with a control digit `ISUBS`. In a way this is comparable to the `MODULE` number: calculations continue until `ISUBS = 0`.

In principle, the subsurface stresses can be calculated in any point of the contacting (elastic) half-spaces, independent of the potential contact area and its discretization. However, the computed stresses are affected adversely by the piecewise constant approximation that is used ($C_3 = 2$, page 32) [20, 54]. This can be resolved using bilinear loading elements, setting $C_3 = 3$, at the expense of longer computations. A practical alternative is to sample the subsurface stresses at the element centers only [54]. This is the recommended approach for most applications.

The use of element centers is facilitated by the input options `ISUBS = 1–7`. An option to compute subsurface stresses at any location (non-centers) is provided by `ISUBS = 9`.

The input of one block describes the following values:

<code>NX,NY,NZ</code>	–	Number of x -, y - and z -coordinates used in a block of subsurface points. The block consists of $NX \cdot NY \cdot NZ$ points.
<code>X(NX),Y(NY)</code>	$[mm]$	Coordinate specification for a block of subsurface points.
<code>Z(NZ)</code>		Note: in the actual calculations and output, the points are sorted in ascending order.

The coordinates per block can be specified in different ways.

<code>ISUBS</code>	–	Integer flag. Input option for the coordinate specification for a block.
--------------------	---	--

- `ISUBS = 1`: at the centers of all discretization elements of the potential contact area, using constant spacing `DZ` in vertical direction.

<code>NZ</code>	–	Number of layers n_z in vertical direction.
<code>ZL</code>	mm	z -value z_l of the lowest layer in vertical direction.
<code>DZ</code>	mm	Step size δz between layers in vertical direction.

The values $z_{i_z}, i_z = 1, \dots, n_z$ are formed as $z_{i_z} = z_l + (i_z - 1)\delta z$.

- `ISUBS = 2`: at the centers of a regular selection of discretization elements, using constant spacing `DZ` in vertical direction.

<code>IXL</code>	–	Starting column number i_x^l for the selection.
<code>IXINC</code>	–	Column number increment i_x^{inc} for the selection.
<code>IXH</code>	–	Ending column number i_x^h for the selection.

The range of column numbers is formed as $i_x = i_x^l + k \cdot i_x^{inc}$ for all values of k such that $i_x^l \leq i_x \leq i_x^h$. Column numbers outside the potential contact area (less than 1 or higher than MX) are ignored. The number of points n_x is determined accordingly.

Similar values IYL, IYINC and IYH are used for the selection of rows of the potential contact area.

The values for z_{i_z} are specified in the same way as for ISUBS = 1.

- ISUBS = 3: at the centers of an irregular selection of discretization elements, using constant spacing DZ in vertical direction.

NX	–	Number of column numbers used in x -direction.
IX(j)	–	List of column numbers i_x^j used in x -direction.

Similar values NY and IY(j) are used for the selection of rows of the potential contact area.

The values for z_{i_z} are specified in the same way as for ISUBS = 1.

- ISUBS = 5: at the centers of all discretization elements of the potential contact area, at explicitly specified Z-positions.

NZ	–	Number of layers in vertical direction.
Z(j)	–	List of vertical positions z_j for all layers.

- ISUBS = 6: at the centers of a regular selection of discretization elements, specified in the same way as with ISUBS = 2, at explicitly specified Z-positions as with ISUBS = 5.
- ISUBS = 7: at the centers of an irregular selection of discretization elements, cf. ISUBS = 3, at explicitly specified Z-positions as with ISUBS = 5.
- ISUBS = 9: at explicitly specified X-, Y- and Z-positions.

See NX, NY, NZ and X(NX), Y(NY) and Z(NZ) described above.

5.9.3 Output of subsurface stress calculation

The output of the subsurface stress calculation is written to the file `<experim>.<ncase>.subs` (Section A.5), which can be imported directly in the Matlab program using the script `load_strs.m` (Section 7.3). Maximum values per block are written to the file `<experim>.out` according to the 0_s -digit, and for $0_s \geq 4$ the complete results are written to the `out`-file as well.

Within subprogram STRESS a different numbering of coordinate directions is used than in the remainder of the program. Here the x -axis is the first coordinate direction, $y = 2$ and $z = 3$. In this subprogram positive values are used for *tensile* rather than compressive stress.

The theory related to this calculation is described in [14], which is reproduced in Appendix C of [15].

- XTABL(k, 3)** [mm] Expanded list of coordinates of points where subsurface stresses are calculated. Each point \mathbf{k} occupies one line, one additional line is used for each block. Note: the points are sorted in ascending order.
- DISPL(k, i)** [-/mm] Displacements in direction i in subsurface point \mathbf{k} : $\mathbf{u}^{(a)}(\mathbf{x}_k) = [u_x^{(a)}, u_y^{(a)}, u_z^{(a)}]^T$. Here a stands for the body number, with body 1 the upper body with $z \geq 0$.

At each point \mathbf{x} in the subsurface there's a stress tensor $\boldsymbol{\sigma}(\mathbf{x})$:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}. \quad (5.33)$$

SIGMA(k, i, j) [N/mm²] Stresses $\sigma_{ij}(\mathbf{x}_k)$ at subsurface point \mathbf{k} .

The components σ_{ij} aren't always the most interesting quantities for the stress state in a point \mathbf{x} . It is often more relevant to consider additional values that are derived from the tensor $\boldsymbol{\sigma}$. The first one is the mean (hydrostatic) stress σ_{hyd} , one-third of the first stress invariant I_1 :

$$I_1 = \text{trace}(\boldsymbol{\tau}) = \sigma_{xx} + \sigma_{yy} + \sigma_{zz}, \quad \sigma_{hyd} = \frac{1}{3}I_1. \quad (5.34)$$

This describes the change of volume of the material by the stresses acting at \mathbf{x} .

SIGHYD(k) [N/mm²] Mean hydrostatic stress σ_{hyd} .

Subtracting the volumetric stress tensor from $\boldsymbol{\sigma}$ yields the stress deviator tensor \mathbf{s} , that describes the distortion of the material:

$$\mathbf{s} = \boldsymbol{\sigma} - \sigma_{hyd} \mathbf{I}, \quad \text{with } \mathbf{I} \text{ the } 3 \times 3 \text{ identity.} \quad (5.35)$$

The second invariant J_2 of this tensor is used to compute the equivalent (tensile) stress σ_{vm} in the von Mises criterion.

SIGVM(k) [N/mm²] The “equivalent (tensile) stress” or “von Mises stress” σ_{vm} of the von Mises criterion.

These values are computed as

$$J_2 = \frac{1}{2} \sum_{ij} (s_{ij})^2, \quad \sigma_{vm} = \sqrt{3J_2}. \quad (5.36)$$

Using these, the von Mises criterion for the onset of plastic yield may be expressed as

$$J_2 \geq k^2, \quad (5.37)$$

with k the yield strength of the material in simple shear, or as

$$\sigma_{vm} \geq \sigma_{yield}, \quad (5.38)$$

with σ_{yield} the yield strength in simple tension. Note that older versions of CONTACT did not include the factor $1/2$ in J_2 nor the factor 3 in σ_{vm} .

Finally CONTACT provides the principal stresses and the maximum shear stress used in the Tresca failure criterion.

SIGJ(**k**, **j**) $[N/mm^2]$ Principal stresses $\sigma_j, j = 1-3$ at subsurface point **k**. These are the eigenvalues of the stress tensor, ordered such that $\sigma_1 \geq \sigma_2 \geq \sigma_3$.

SIGTR(**k**) $[N/mm^2]$ Maximum shear stress $\sigma_{tresca} = \sigma_1 - \sigma_3$.

The Tresca criterion for the onset of plastic yield is

$$\sigma_{tresca} \geq 2k \text{ or } \sigma_{tresca} \geq \sigma_{yield}. \quad (5.39)$$

Chapter 6

Examples

In this chapter we present a number of examples for the program. These examples correspond to input- and reference-output files that are provided with the program in the `examples` directory.

6.1 The Cattaneo shift problem

A sphere is pressed onto a plane and then shifted tangentially. This problem is entered in two steps in order to illustrate some input options of the CONTACT program.

The first case concerns the frictionless normal problem only. A Hertzian input-option is used: `IPOTCN < 0`. The radius of the sphere is 50 mm such that the curvatures at the contact point are $A = B = 0.01\text{ mm}^{-1}$ (see equation (5.17)). The sphere and plane are assumed to consist of the same (soft) polyethylene material, with $G = 200\text{ N/mm}^2$ and $\nu = 0.42$.

We use the Hertz theory to determine the normal force F_n that creates a contact area with radius 1 mm . For a circular contact area this says that

$$a = b = \left(\frac{3}{2} F_n \frac{1 - \nu^2}{E} \frac{1}{A + B} \right)^{1/3}. \quad (6.1)$$

This gives $F_n = 9.1954\text{ N}$ and for the maximum pressure $p_{n,max} = 3F_n/2\pi ab = 4.3905\text{ N/mm}^2$. In the example these data are entered via `AA` and `BB`, it is possible to use `A1` and `B1` just as well. The contact area is discretised with 15×15 elements, two additional elements are used on all sides by setting `SCALE = 19/15 = 1.267`.

3 MODULE

200100	P-B-T-N-F-S	PVTIME, BOUND , TANG , NORM , FORCE , STRESS
022020	L-D-C-M-Z-E	FRCLAW, DISCONS, INFLCF, MATER , RZNORM, EXRHS
000331	G-I-A-O-W-R	GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW , RETURN

```

100 100 30 1 0.0001 MAXGS , MAXIN , MAXNR , MAXOUT, EPS
9.1954 0.000 0.000 0.000 FN, CKSI, CETA CPHI
0.400 0.400 FSTAT, FKIN
0.420 0.420 200. 200. POISS 1,2, GG 1,2
-3 IPOTCN
19 19 1.000 1.000 1.26667 MX,MY,AA,BB,SCALE

```

When CONTACT is run, information on the Hertz solution is printed to the output-file cattaneo.out:

```

HERTZIAN GEOMETRY WITH NORMAL FORCE PRESCRIBED
SEMIAXES PRESCRIBED, AA,BB: 1.000 1.000
THE CURVATURES A1,B1 ARE: 0.1000E-01 0.1000E-01
EFFECTIVE RAD.CURV RHO, SEMI-AXIS CP 100.0 1.000
POTENTIAL CONTACT, SCALE: -1.267 -1.267 1.267
DISCRETISATION MX,MY, DX,DY: 19 19 0.1333 0.1333

```

The “effective radius of curvature” is $\rho = 2/(A + B)$, the effective semi-axis $c = \sqrt{ab}$.

The second case in the example input-file concerns the tangential shift problem. In this case the geometry is entered using a non-Hertzian approach. Option IBASE = 1 means that a quadratic profile is used. The coefficients b_1 and b_3 that are entered here correspond to the curvatures A and B :

```

% Second case: using non-Hertzian geometry-description, and including the
% tangential shift problem.

```

```

3 MODULE
201120 P-B-T-N-F-S PVTIME, BOUND , TANG , NORM , FORCE , STRESS
022020 L-D-C-M-Z-E FRCLAW, DISCNS, INFLCF, MATER , RZNORM, EXRHS
101441 G-I-A-O-W-R GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW , RETURN
9.1954 -0.8750 0.000 0.000 FN, FX, FY, CPHI
0.400 0.400 FSTAT, FKIN
0.420 0.420 200. 200. POISS 1,2, GG 1,2
1 IPOTCN
19 19 -1.26667 -1.26667 .13333 .13333 MX,MY,XL,YL,DX,DY
1 1 IBASE, IPLAN
% QUADRATIC UNDEFORMED DISTANCE
0.0100 0.000 0.0100 0.000 0.000 0.000 B(I), I=1, 6
% UNRESTRICTED PLANFORM

```

The relevant control digits (Section 3.3) are

- the P-digit, which describes that the sphere was free from tangential tractions initially;

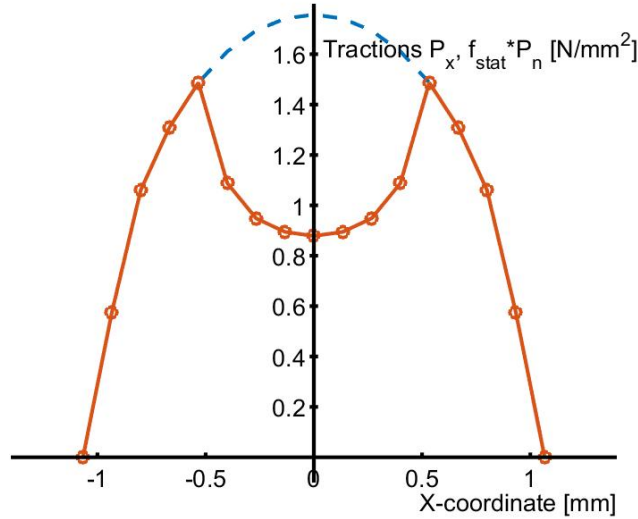


Figure 6.1: Results for the Cattaneo shift problem. Distribution of tangential traction p_x along the center-line $y = 0$.

- the T-digit, which states that the problem concerns a shift;
- the N- and F-digits, that state that total forces are being prescribed.

A tangential force is applied to the sphere of $(1 - \theta^3) \mu_{stat} F_n$, with $\theta = 1/2$. According to Cattaneo this yields a circular adhesion area with radius $(1 - \theta)$ concentric with the contact area [15, sec. 5.2.1.1]. The corresponding shift of the sphere with respect to the plane is given by

$$W_x = (1 - \theta^2) \frac{3\mu_{stat} F_n}{2\pi G} \left(\frac{\pi}{2} - \nu \frac{\pi}{4} \right) \quad (6.2)$$

In our case the sphere is the upper body, body 1. The tangential force F_x that is entered in the input-file concerns the total load in the contact interface, which oppose the force by which the sphere is shifted. Further note that the force is entered relative to the maximum $\mu_{stat} F_n$, hence $F_x = -7/8$. The shift is $W_x = 0.00817 \text{ mm}$, and the theoretical tangential traction is

$$\begin{aligned} p_x &= -\mu_{stat} p_{n,max} \left(\sqrt{1 - r^2} - \frac{1}{2} \sqrt{1 - 4r^2} \right), & \text{when } 0 \leq r \leq \frac{1}{2} \\ &= -\mu_{stat} p_{n,max} \sqrt{1 - r^2} & \text{when } \frac{1}{2} \leq r \leq 1 \\ &= 0 & \text{when } r \geq 1. \end{aligned} \quad (6.3)$$

This Cattaneo solution is not altogether correct; there is a traction component p_y orthogonal to x -axis that is ignored.

After you’ve run CONTACT for this problem, you find the shift in the output-file:

FN	FX	FY	MZ	ELAST.EN.	FRIC.WORK
9.195	-3.218	0.000	0.000	8.525E-05	-2.750E-06

FN/G	SHIFT X	SHIFT Y	APPROACH
4.598E-02	8.155E-03	0.000	1.998E-02

The results may be inspected further with the Matlab package, see Figure 6.1. The commands for producing this figure are (see Section 7.2):

```
s = loadcase('cattaneo',2); % the tangential problem concerns the
opt2 = plot2d;                % second case in the actual input-file
opt2.yslc = 0.0;
opt2.facpt = -1.0;
plot2d(s, opt2);
```

This shows the negative of the tangential tractions, $-p_x$, along the centerline through the contact area $y = 0$. The dashed line is the traction bound $\mu_{stat} p_n$, with maximum $\mu_{stat} p_{n,max} = 1.756 N/mm^2$.

6.2 The 2D Carter/Fromm problem

The second example concerns the steady rolling of an infinite cylinder over an elastic half-space (equivalently: two infinite cylinders with doubled radius) with the same elastic constants. This is a 2D “plane strain” problem. Such a problem is input to CONTACT using a “truncated 3D problem”: by using a single row of elements with a very large δy . Note that the difference between a very large strip $[-a, a] \times [-\delta y/2, \delta y/2]$ and an infinite strip $[-a, a] \times [-\infty, \infty]$ is negligible at $y = 0$, except for the displacements \mathbf{u} that are ill-defined in 2D problems [9].

We consider a steel cylinder with a radius of $R^{(1)} = 500 mm$ and a steel half-space ($R^{(2)} = \infty$, combined radius $R = R^{(1)}$), with elastic constants $G^{(a)} = 82000 N/mm^2$, $\nu^{(a)} = 0.28$. According to the 2D Hertz solution, the normal force required per unit of width to achieve a contact area $[-a, a]$, with $a = 1 mm$ is

$$F_n = \frac{\pi a^2 G}{4 R (1 - \nu)} = 178.90 N/mm \quad (6.4)$$

This is entered in CONTACT multiplied by the element width δy as `FN = 35780`. The undeformed distance is specified using option `IBASE = 1`: a quadratic function in x and y (page 59). In this case there is no dependence on y , such that the corresponding b_i are set to 0.

The theoretical relations between the size of the adhesion area, the total tangential force and the longitudinal creepage are:

$$\xi = -\mu_{stat} \frac{a - a'}{R}, \quad F_x = (1 - (a'/a)^2) \mu_{stat} F_n. \quad (6.5)$$

For adhesion in 60% of the contact area ($a' = 0.6$) this yields $\xi = -0.024\%$ and `FX = 0.64`.

The input used for CONTACT is as follows.

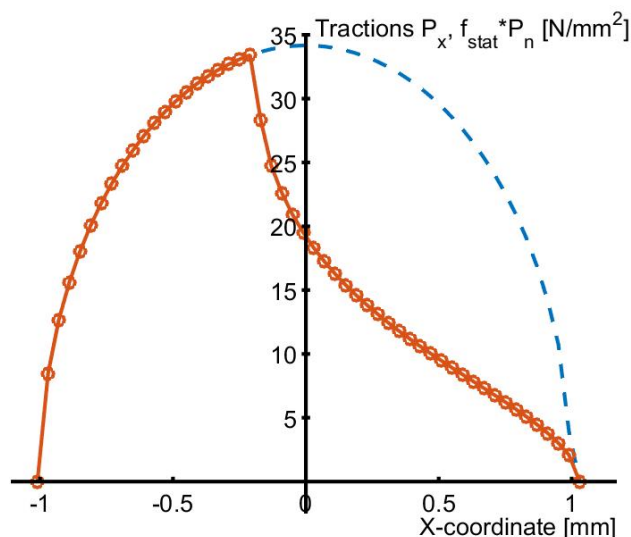


Figure 6.2: Tangential tractions p_x for the 2D Carter/Fromm problem.

```

3  MODULE
203100      P-B-T-N-F-S      PVTIME, BOUND , TANG , NORM , FORCE , STRESS
022020      L-D-C-M-Z-E      FRCLAW, DISCNS, INFLCF, MATER , RZNORM, EXRHS
001441      G-I-A-O-W-R      GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW , RETURN
  999  100  30  1  1e-5      MAXGS , MAXIN , MAXNR , MAXOUT, EPS
35780.     -0.00024  0.000  0.000      FN, CKSI, CETA, CPHI
  0.300    0.300
  0.000    0.040  30000.      FSTAT, FKIN
  0.280    0.280  82000.  82000.      CHI, DQ, VELOC
  3      POISS 1,2,  GG 1,2
  55  1  -1.0500  0.000  0.0400  200.0  IPOTCN
  1  1      IBASE, IPLAN
% QUADRATIC UNDEFORMED DISTANCE
  0.001  0.0  0.0  0.0  0.0  0.0      B1,B2,B3,B4,B5,B6
% UNRESTRICTED PLANFORM

```

Note that the grid is specified in a different way than in the previous example ($IPOTCN = 3$), and that a time step size $\delta q = \delta x$ is used. The resulting traction distribution is shown in Figure 6.2.

6.3 The 2D Bental-Johnson test-case

Additional phenomena come into play when the two bodies are made of dissimilar materials. Analytical and numerical solutions were provided for this by Bental and Johnson [1] for the

contact of aluminum and steel rollers.

An aluminum roller ($G^{(1)} = 23\,000\text{ N/mm}^2$, $\nu^{(1)} = 0.31$) with radius $R^{(1)} = 50\text{ mm}$ is pressed onto a steel roller ($G^{(2)} = 82\,000\text{ N/mm}^2$, $\nu^{(2)} = 0.28$) of the same size. The normal load is defined using equation (6.4) such that the Hertzian half contact width becomes $a_h = 0.1\text{ mm}$. Note that the effective radius is $R = 25\text{ mm}$: as far as the contact is concerned, two rollers of $R^{(1)} = R^{(2)} = 50$ look like a single roller on a plane with $R = 25\text{ mm}$. In the input this shows up via $b_1 = 1/2R = 0.02\text{ mm}^{-1}$. Further, the combined material parameters G , ν and K are required, that are defined according to [15, eq.(1.44)]:

$$\frac{1}{G} = \frac{1}{2} \left(\frac{1}{G^{(1)}} + \frac{1}{G^{(2)}} \right), \quad \frac{\nu}{G} = \frac{1}{2} \left(\frac{\nu^{(1)}}{G^{(1)}} + \frac{\nu^{(2)}}{G^{(2)}} \right), \quad \frac{K}{G} = \frac{1}{4} \left(\frac{1 - 2\nu^{(1)}}{G^{(1)}} - \frac{1 - 2\nu^{(2)}}{G^{(2)}} \right). \quad (6.6)$$

This gives $G = 35\,924$, $\nu = 0.3034$, $K = 0.100$, which in turn gives $F_n = 16.20\text{ N/mm}$.

The first scenario concerns “free rolling”. One roller is driven by an external torque such that it rotates at a constant speed. The other is rolling freely, with negligible friction in its bearings. It is accelerated or decelerated by the contact force, until it approaches a steady state with $F_x = 0$. In a case with equal materials this would give the same circumferential velocities for both rollers, leading to creepage $\xi = 0$. With dissimilar materials the situation is different.

Normal pressures p_n cause tangential displacements $u_x^{(a)}$, $u_y^{(a)}$ to occur in both bodies, $a = 1, 2$. These displacements are bigger for the aluminum than for the steel roller. The sign of $u_x^{(a)}$ is such that particles are drawn towards the center of the contact area. This means that the particles of the aluminum roller traverse a smaller distance when passing through the contact area: the particles of the steel roller move by $\approx 2a$ and those of the aluminum roller by $\approx 2a - 2u$. This introduces a non-zero creepage at free rolling.

The different tangential displacements in the two bodies imply a tendency of the surfaces to slip with respect to each other. If rolling is to the left (counter-clockwise rotation for the upper roller), the upper surface tends to the left with respect to the lower one. This tendency is resisted by tangential tractions $p_x > 0$ (in 2D scenarios, $p_y \equiv 0$). Note that normal pressures invoke tangential tractions. Vice versa, tangential tractions affect the pressures p_n as well. Finally, free rolling is obtained by introducing a creepage $\xi > 0$, that balances the upper surface’s tendency to the left. This is investigated using four cases:

- 1, 2. Minimal friction, $\mu = 0.0001$, such that there’s practically no influence from p_x on p_n ;
3. Maximal friction, $\mu = 10.0$, such that the influence from p_x on p_n is maximal;
4. Typical friction, $\mu = 0.05$, for aluminum on steel contact.

The input used for CONTACT looks as follows:

3 MODULE

203100	P-B-T-N-F-S	PVTIME, BOUND , TANG , NORM , FORCE , STRESS
022020	L-D-C-M-Z-E	FRCLAW, DISCNS, INFLCF, MATER , RZNORM, EXRHS
000011	G-I-A-O-W-R	GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW , RETURN
500 100 30 1	1e-6	MAXGS , MAXIN , MAXNR , MAXOUT, EPS
16201.9	0.00052 0.000 0.000	FN, CKSI, CETA, CPHI
0.0001	0.0001	FSTAT, FKIN
180.0d	0.001 1000.	CHI, DQ, VELOC
0.310	0.280 23000. 82000.	POISS 1,2, GG 1,2
1		IPOTCN
216 1	-0.108 -500. 0.001 1000.	MX,MY,XL,YL,DX,DY
1 1		IBASE, IPLAN
0.020	0.0 0.0 0.0 0.0 0.0 0.0	B1,B2,B3,B4,B5,B6

A few points are worth emphasizing:

- This first case uses a prescribed creepage, $F = 0, \xi = 0.00052$, instead of specifying the force $F_x = 0$. This is because the Newton-Raphson procedure has great difficulty with scenarios where full sliding occurs at practically all creepages. This first case is used with minimal output. Then the second case uses its results as initial estimate, by setting $I = 1$. With this improved initial estimate the Newton-Raphson process works well.
- In the first two cases we use $\text{MAXOUT} = 1$ even though there's material dissimilarity. This means that Johnson's process is used instead of the Panagiotopoulos process, whereby the influence of tangential tractions on normal pressures is ignored. This is safe in this case because the influence is negligible.
- Rolling is to the left, $\chi = 180^\circ$, such that particles enter the contact area at $x = -0.1$ and leave at $x = 0.1 \text{ mm}$. This honours the convention used by Bentall and Johnson such that the pictures are compared more easily.

The theoretical result by Bentall and Johnson says that the creepage at free rolling is

$$\xi = 0.457 \frac{\kappa a}{R}, \quad \text{with } \kappa = 2\beta = \frac{2K}{1-\nu} = 0.2877. \quad (6.7)$$

Here β is Dundur's constant [9, p.110] and K is Kalker's difference parameter. The formula gives $\xi = 0.0005259$ whereas CONTACT gives $\xi = 0.0005260$ at $\mu = 0.0001$. At $\mu = 0.05$ the creepage reduced to 0.000429, whereas at $\mu = 10.0$ it reduced further to $\xi = 0.000359$. Note that the sign of the creepage depends on the convention used for the rolling direction and on numbering of the rollers (through the signs of κ and K). This is summarized as that the softer roller tends to roll at higher angular velocity than the stiffer one.

The maximum pressure increases from 103.14 to 103.81 N/mm^2 (+0.65%) due to the influence of p_x on p_n . This doesn't show up in the contact area that consists of 200 elements in all four cases. A case with a larger influence is shown later in Section 6.10.

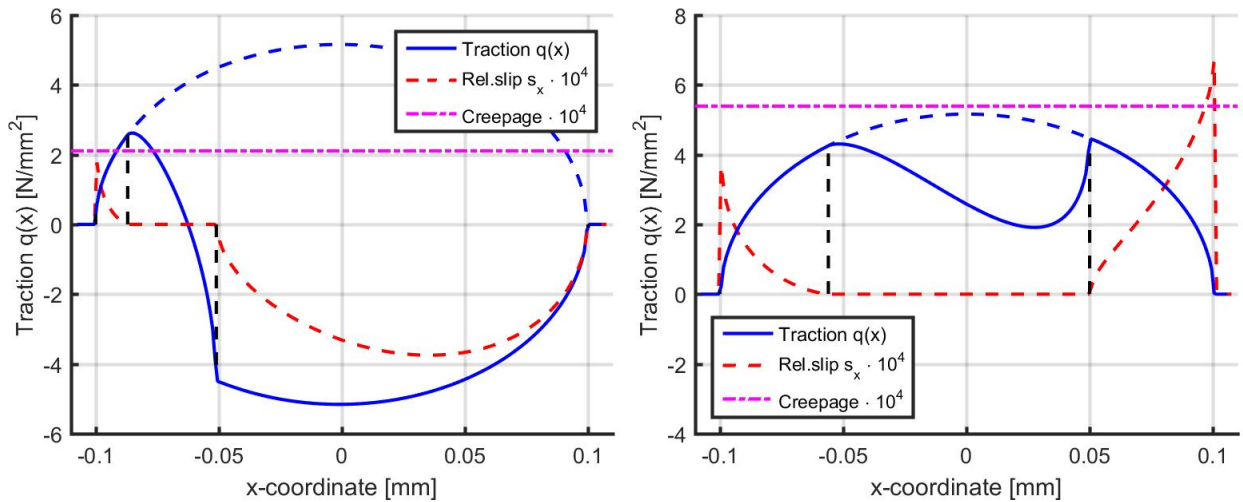


Figure 6.3: Traction $q(x) = p_x^{(2)}$ on the bottom roller in the Bentall-Johnson test-case for two scenarios of tractive rolling, cf. [1, Fig.9]. Left: $F_x = 0.75$ on upper roller, right: $F_x = -0.75$. In both cases $\mu = 0.05$.

Two more cases in the input-file concern tractive rolling at relative tangential forces $F_x = 0.75$ and -0.75 . In the first of these the aluminum roller (1) is braked by the contact force exerted by the steel roller (2), in the latter case the aluminum roller is accelerated. The results of these cases are shown in Figure 6.3. These results show excellent agreement with the corresponding results presented in [1, Fig.9].

6.4 Calculation of creep force curves

A main outcome of CONTACT for railway applications is the relation between the creepages and creep force. These are the subject of example `tractcurv.inp`, which concerns the creep force for the Siemens locomotive Europrinter 127 001. Measurements of the creep force were presented by Engel et al. for a study of traction control strategies [6]. The measurements concern straight running on tangent track, i.e. with pure longitudinal creep. The measurements are shown in Figure 6.4 together with the computed results.

The creep force curve is computed in CONTACT with a series of cases, one for each creep value. In the example 30 cases are used per creep force curve, with small steps for ξ at first and larger spacing after saturation has occurred. The input for the first case can be understood on the basis of the examples presented above:

3 MODULE

203100	P-B-T-N-F-S	PVTIME, BOUND , TANG , NORM , FORCE , STRESS
022020	L-D-C-M-Z-E	FRCLAW, DISCNS, INFLCF, MATER , RZNORM, EXRHS
000341	G-I-A-O-W-R	GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW , RETURN

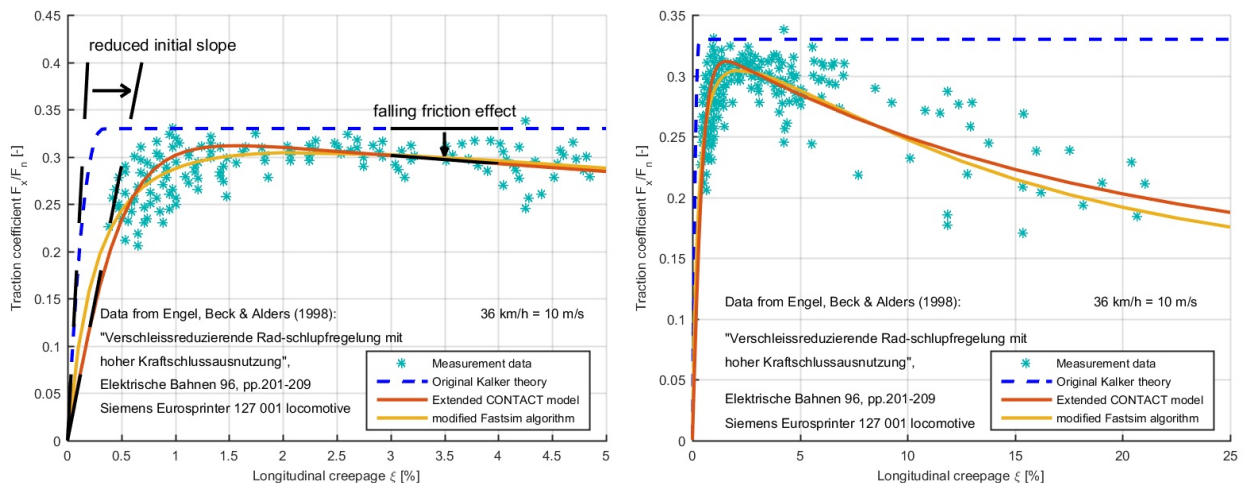


Figure 6.4: Measured and computed creep forces for the Siemens locomotive Europrinter 127001 for pure longitudinal creepage.

```

80 200 5 1 1e-6 MAXGS , MAXIN , MAXNR , MAXOUT, EPS
106700. 0.00001 0.000 0.000 FN, CKSI, CETA, CPHI
0.330 0.330 FSTAT, FKIN
0.000 0.100 10000. CHI, DQ, VELOC
0.280 0.280 82000. 82000. POISS 1,2, GG 1,2
-2 IPOTCN
44 44 0.0008 0.500 1.100 MX,MY,A1,AOB,SCALE
    
```

Noteworthy points are as follows:

- The wheel radius is $R = 625 \text{ mm}$, such that the curvature in rolling direction is $A = 0.0008 \text{ mm}^{-1}$.
- The Hertzian input option $\text{IPOTCN} = -2$ is used for prescribing A and a/b . The value $a/b = 0.5$ is typical for a wheelset at central position on the rails.
- For this locomotive the vertical load is 106.7 kN per wheel, and the velocity used in the experiments is $V = 10 \text{ m/s}$.
- The first creep force curve concerns Kalker’s original model with Coulomb dry friction: $L = 0, \mu_{stat} = \mu_{kin} = 0.33$.

The following 29 cases re-use most of the inputs by proper setting of the control digits (L, D, C, Z and G):

```

3 MODULE
203100 P-B-T-N-F-S PVTIME, BOUND , TANG , NORM , FORCE , STRESS
    
```

```

100000      L-D-C-M-Z-E   FRCLAW, DISCNS, INFLCF, MATER , RZNORM, EXRHS
100341      G-I-A-O-W-R   GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW , RETURN
106700.     0.0004      0.000      0.000      FN, CKSI, CETA, CPHI

```

In the 31st case the calculation of a second creep force curve is started. This curve concerns extensions of CONTACT for incorporating the “reduced initial slope” and “falling friction” effects, see Figure 6.4 (left). These effects are incorporated via the interfacial layer of material model $M = 4$ (Section 5.1.4) and the velocity dependent friction of $L = 4$ (Section 5.2). The corresponding inputs are:

```

% Exponential falling friction (L=4):
  0.1400    0.1900    1250.    0.000    0.000  FKIN, FEXP1,SABSH1,FE2,SH2
  0.003     1.000
...
% Elastic bodies with elasto-plastic interface layer (M=4):
  0.280     0.280     82000.  82000.    POISS 1,2,  GG 1,2
  8200.     1.250     0.000   0.000    GG3, LAYTHK, TAUCRT, GGPLST

```

Next, cases 61 to 90 employ the original Fastsim algorithm with parabolic traction bound ($M = 3, B = 3$). Finally cases 91 to 120 use “modified Fastsim” with parameters $k_0 = 0.54, \alpha_{inf} = 0.02, \beta = 0.64$ as specified in [33]. The friction parameters are derived with the help of equation (5.13) with $\mu_{stat} = 0.36, A = 0.38, B = 0.7$.

```

% Exponential falling friction (L=4), static 0.36(!):
  0.1368    0.2232    990.    0.000    0.000  FKIN, FEXP1,SABSH1,FE2,SH2
...
% Slope reduction for modified Fastsim algorithm (M=3):
  0.540     0.020     0.620    KO_MF,ALFAMF,BETAMF

```

After the simulation, the traction forces F_x are presented in the output-file `tractcurv.out`:

```

FN          FX          FY          MZ          ELAST.EN.   FRIC.POWER
1.067E+05  -248.6          0.000        0.000        3.265       -9.007E-05
FN/G       FX/FSTAT/FN    FY/FSTAT/FN  APPROACH
1.301      -7.060E-03     0.000        7.651E-02

```

They are imported into Matlab using the script `parse_out3.m` and plotted with the following commands:

```

[creep, force] = parse_out3('tractcurv.out');
cksi = reshape( creep(:,2), 30, 4 );
fx    = reshape( force(:,2), 30, 4 );
fstat = [ 0.33, 0.33, 0.33, 0.36 ];
plot(cksi, -fx*diag(fstat), '-o');

```

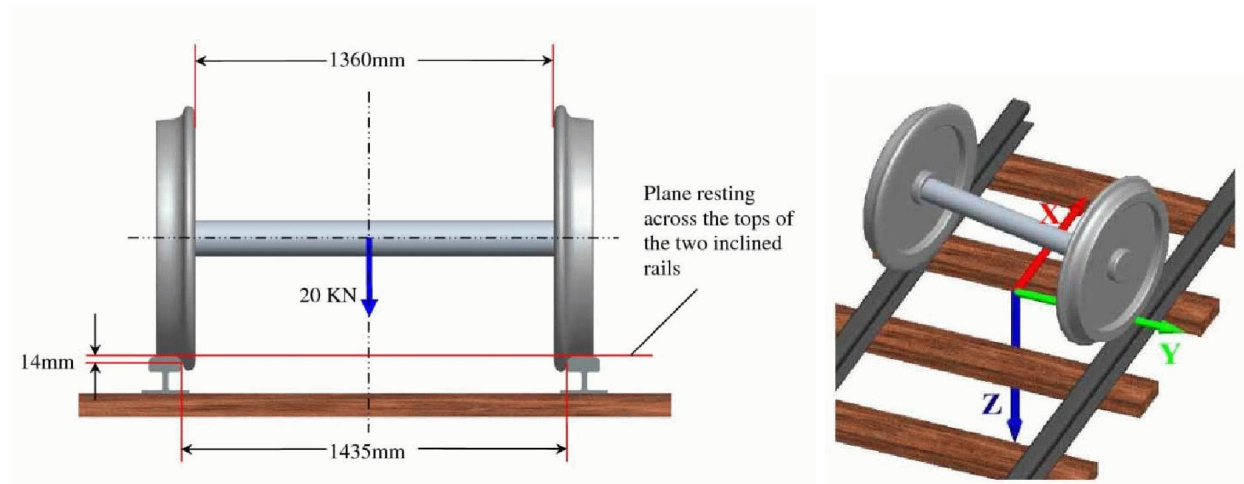


Figure 6.5: *Illustration of the simple wheelset model used in Manchester benchmark simulation Case A [30].*

Note that the conventions used in CONTACT make that a positive creepage ξ comes with a negative force F_x . For a coordinate system with z pointing upwards, the upper body 1 is the wheel. A positive creepage ξ then means that particles of the wheel move slower through the contact area than the particles of the rail. Adhering together of the particles then requires negative displacements $u_x^{(1)}$ (and positive $u_x^{(2)}$), which again requires negative tractions $p_x^{(1)}$ acting on the wheel surface. Since creep force curves are typically plotted with positive force for positive creep we use $-fx$ in the Matlab plot-command.

6.5 The Manchester wheel-rail benchmark

An important application area for CONTACT concerns the detailed study of wheel-rail problems. In order to illustrate the use of CONTACT for realistic wheel and rail profiles, we consider a case from the Manchester contact benchmark. This benchmark is proposed in [30] and is presented together with initial results in [31]. Data for the profiles are provided¹ thanks to dr. Shackleton of the Institute of Railway Research of Huddersfield University.

The aim of “Case A” of the benchmark is to compare predictions from different contact models for clearly defined contact conditions. To this end a single wheelset is considered as illustrated in Figure 6.5. Real wheel (new S1002 wheels) and rail profiles (new UIC60 rails at 1:40 inclination) are used, with prescribed lateral displacement, yaw angle, vertical load, velocity, and coefficient of friction.

The example uses two input-files, `mbench_a22_left` and `mbench_a22_right`, for the left and right wheels of the wheelset respectively. Each file defines 21 cases corresponding to bench-

¹www.kalkersoftware.org/downloads/manch-benchmark.zip

mark case A-2.2 with steady rolling and including yaw.

The example uses `module 1` for wheel/rail contact analysis (Chapter 4), concerning a wheelset of which the position and velocity states are fully prescribed. This isn’t entirely so in the benchmark problem, that relies on the dynamic equations to complete the problem specification. For instance, it isn’t defined how the vertical load is imposed: at the center of mass or at the bearings, distributed equally or in an asymmetrical way. This is dealt with by making some ad-hoc assumptions; once the states are better described, the resulting problem can be solved in a similar way.

First we solved the static problem of the benchmark, with vertical force divided equally over the two wheels. This uses option $N_1 = 1$ with $FZ = 10\text{ kN}$. The primary unknown in this case is the roll angle ϕ_{ws} : if this is chosen poorly, then different values for the wheelset z_{ws} are obtained for left and right wheels. A basic iteration was used, starting from $\phi_{ws} = 0$, solving the left and right contacts, and adjusting ϕ_{ws} to account for the difference, in order to find the appropriate values.

After this the full benchmark problem was solved, including rolling at a speed $V = 2\text{ m/s}$. Here, the difficulty is to choose the wheelset angular velocity ω_{ws} in an appropriate way. In the end, at steady rolling, there should be no resulting moment in rolling direction, i.e. $r^{lft} F_x^{lft} + r^{rgt} F_x^{rgt} = 0$. This is simplified in the example, tuning ω_{ws} to get $F_x^{lft} + F_x^{rgt}$ close to zero. The procedure we used consisted of sampling F_x for a range of possible values, using interpolation to get the final result. Note that the roll angle ϕ_{ws} wasn’t adjusted with respect to the frictionless case, such that some discrepancies arise between the two wheels.

The resulting input for a case is then entered as follows:

```
% 13: Lateral displacement 6.0 mm, yaw angle 14.4 mrad

1 MODULE
0203100 C-P-B-T-N-F-S CONFIG, PVTIME, BOUND, TANG, NORM, FORCE, STRESS
122033 L-D-C-M-Z-E FRCLAW, DISCNS, INFLCF, MATER , ZTRACK, EWHEEL
101221 G-I-A-O-W-R GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW , RETURN
0.280 0.280 82000. 82000. POISS 1,2, GG 1,2
0.200 0.200 1.000 1.000 DX, DS, DQREL, SAFETY
1435.0 14.0 0.000 GAUGWD, GAUGHT, CANT
'MBench_UIC60_v3.prr' 0 RFNAME, MIRRORY
0.0 0.0 0.000 0.0 0.0 0.000 DY, DZ, ROLL, VY, VZ, VROLL
1360.0 -70.0 460.0 FBDIST, FBPOS, NOMRAD
'MBench_S1002_v3.prw' 0 WFNAME, MIRRORY
0.0 6.0 10000. -0.0006340 0.0144 0.0 S, Y, FZ, ROLL, YAW, PITCH
2000. 0.0 0.0 0.0 0.0 -4.329592 VS, VY, VZ, VROLL, VYAW, VPITCH
```

One interesting point regarding this input is the absence of cant. This is included in the profile itself, following the original benchmark specification, that describes the rail in the canted position.

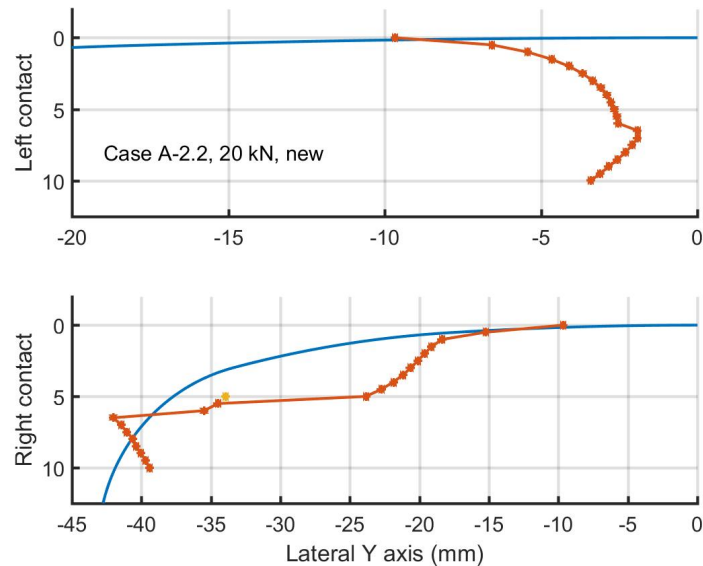


Figure 6.6: Results for Manchester benchmark example: contact locations at left and right wheels, as function of the lateral wheelset displacement.

One interesting result for this example concerns the contact locations that are found at the different wheelset positions. These are shown in Figure 6.6, for the left and right rails in the top and bottom graphs respectively. The coordinates used on the horizontal axis are with respect to the rail origin, chosen as the highest point in the canted rail profile. Note that the track center is on the left for both rails, that is, left-handed coordinates are used for the left rail. Note further that two contact patches are found on the right rail if the wheelset is displaced to $y_{ws} = 5 \text{ mm}$. The results are generally in line with those presented in [31], showing that the yaw angle has little effect on the lateral contact position.

The results are illustrated further in Figure 6.7, for a lateral displacement of 6 mm , displaying the contact patches at the rail surface. These pictures are created as follows:

```
l13 = loadcase('mbench_case_a22_left',13);
pr = read_profile('Mbench_UIC60_v3.prr');
opt = plot3d; opt.rw_surfc = 'pr';
plot3d(l13, opt, pr);
set(gca,'ydir','reverse')
```

Note how the y -axis is reversed for the left-handed wheel.

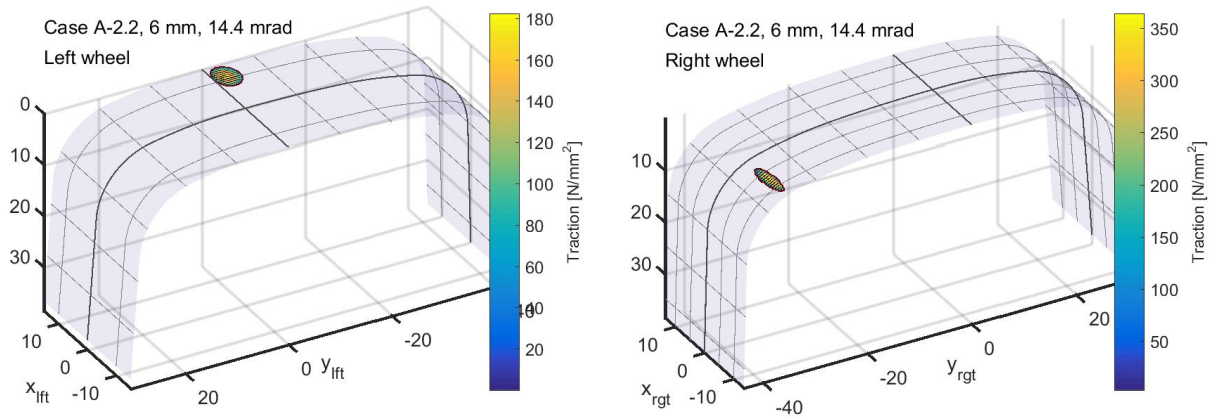


Figure 6.7: Results for Manchester benchmark example: wheelset with lateral displacement $y_{ws} = 6 \text{ mm}$, yawed at $\psi_{ws} = 14.4 \text{ mrad}$.

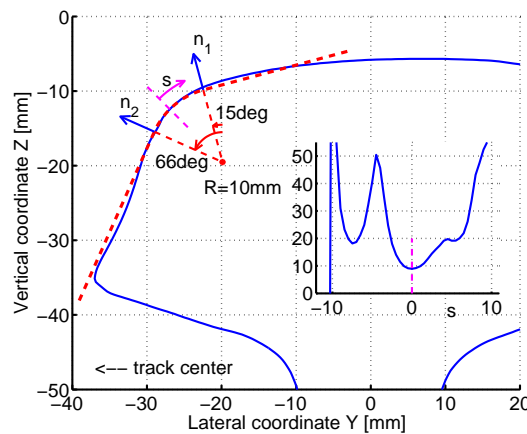


Figure 6.8: Measured worn profile with conformal contact situation at the rail gauge corner.

6.6 Conformal contact

In wheel-rail contact analysis, it is typically assumed that the contact is “concentrated”, i.e. the contact area is assumed small with respect to the dimensions of the contacting bodies as a whole, such that the contact area almost lies in a plane. This assumption is clearly violated in the case of contact between the wheel flange root and rail gauge corner. This is illustrated in Figure 6.8 via a measured worn rail profile. At the rail gauge corner the normal direction changes orientation by 41° over a distance of 7 mm , because the radius of curvature goes down to less than 10 mm . This leads to *conformal contact* situations between the flange root and rail gauge corner, where the contact area is curved.

The example in this section is taken from the paper [45], and focuses on the different aspects related to solving conformal contact problems. Three different aspects are taken into account:

undeformed distance The computation of the undeformed distance between wheel and rail is changed [49]. Approximating a circular arc by a quadratic function is no longer appropriate, and the changing normal direction is taken into account.

varying spin creepage The rigid slip originating from spin cannot be represented by $\mathbf{w} = [\phi y, -\phi x]^T$ anymore (equation (5.27)), but must take the varying normal direction into account [19].

influence coefficients The response of the true bodies to surface loads deviates from the response of the elastic half-space. The true response can be computed using FEM and introduced via numerically computed influence coefficients ([49], see Section 5.1.5, Figure 5.3).

We consider a vehicle during steady curving. The wheelset is positioned such that the outer wheel makes contact to the rail at the rail gauge corner at a position where the contact angle is about 45° (position $s = 0$ in Figure 6.8). This defines the spin creepage $\phi = -0.001537 \text{ rad/mm}$ for a planar contact analysis. The longitudinal and lateral creepages ξ and η can take any values in principle, depending on the angle of attack (yaw angle ψ) and the rolling radius difference between left and right wheels of the wheelset. These depend among others on the radius of curvature and the steering ability of the vehicle. The values that are selected are $\psi = -14 \text{ mrad}$ ($\eta = -0.99 \%$) and $\xi = -0.4 \%$.

The curvature of the contact patch is defined through the transverse radius $R_{yr} = 10.0 \text{ mm}$ for the rail profile. A strongly conformal situation is constructed using radius of curvature $R_{yw} = -10.2 \text{ mm}$ for the wheel, which may occur in the flange root of a worn wheel with S1002 profile. A typical value of $F_n = 100 \text{ kN}$ is used for the wheel normal load. The corresponding approach $\delta_n = 0.09089 \text{ mm}$ is derived using the Hertzian theory, and is then held fixed in the different cases such that the total force becomes variable again.

This single scenario is modeled in five different cases in input-file `conformal.inp`.

1. In the first case, a Hertzian approximation is used and all effects of conformality are ignored. The profile is entered via the quadratic function (5.22), with the coefficients computed using (5.17).
2. In the second case, the true undeformed distance is computed in Matlab and entered into CONTACT via option `IBASE = 9`. The approach δ_n is used already in Matlab such that $\delta_n = 0$ is entered in the input-file.

The third, fourth and fifth cases re-use the grid and undeformed distance of the second case, by setting `D = 0` and `Z3 = 0`.

3. In the third case, numerically computed influence coefficients are used. This is entered via `C3 = 9` and `CFNAME = 'inflcf_r10_mx51.txt'`. The file is prepared in advance, using finite element calculations for meshes such as the ones displayed in Figure 5.3 [49]. There is a close correspondence between the CONTACT grid and the input-file: these

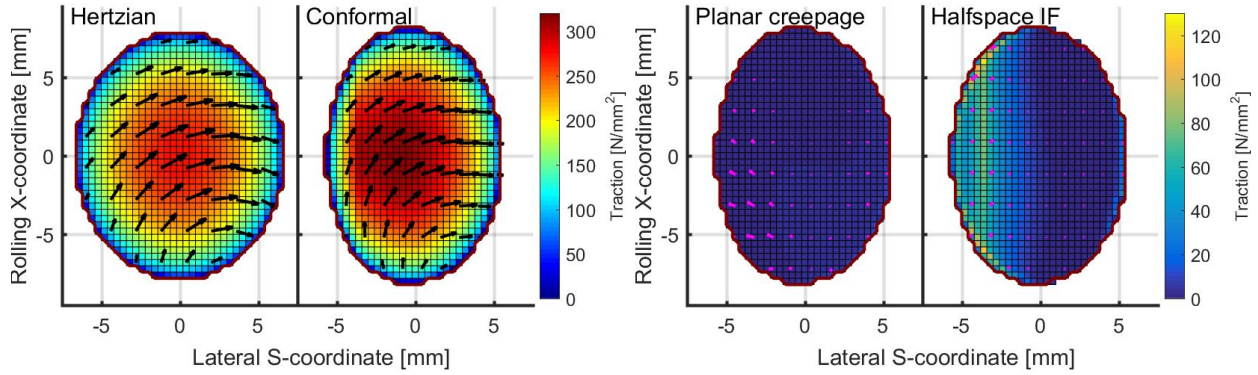


Figure 6.9: Comparison of results for different approximations to a conformal contact problem. Left: results for case 1, Hertzian approach, and case 5, full conformal approach. Right: contribution of planar \rightarrow varying creepage (case 3 \rightarrow 5) and of halfspace \rightarrow numerical influence functions (4 \rightarrow 5).

must both use the same grid spacings $\delta x, \delta y$ and also the number of elements must be the same.

4. In the fourth case, the effect of conformality on the creepages is properly taken into account. The rigid slip is evaluated in Matlab for all points of the potential contact area and printed to a file, and is incorporated in the CONTACT input-file via option $E_3 = 9$. Since the full rigid slip is computed in Matlab we set $CKSI = CETA = CPHI = 0$. Further we use the half-space elasticity again, setting $C_3 = 2$ instead of $C_3 = 9$.
5. Finally, the fifth case incorporates all three effects simultaneously. The proper undeformed distance that was introduced in case 2, the numerically computed influence coefficients of case 3, and the varying creepages of the fourth case. Since the rigid slip is the same as in the previous case it can be reused, using $E_3 = 1$, instead of repeating all the values again.

The results are illustrated in Figure 6.9. It can be seen that the undeformed distance calculation has a considerable effect on the size and width of the contact patch. The Hertzian approach overestimates the width and contact area by 10–15% and underestimates the pressures and tangential traction correspondingly. Varying creepage and numerical influence coefficients have less pronounced effects on the total forces, but do change the distribution of stresses over the contact patch.

The pictures of Figure 6.9 are created using `opt.field='ptabs+vec'`. In order to get two plots in one figure we use Matlab's `subplot` command and set `opt.addplot=1`. The colors for magnitude of tangential tractions are controlled using `opt.zrange=[0 320]` and `[0 130]` respectively. In the figure on the right we used `opt.veccolor='m'` and `opt.vecscale=0.005` ($[mm]$ per $[N/mm^2]$).

Note that the colors in Figure 6.9, right, indicate the difference in magnitude, $\|\mathbf{p}^i\| - \|\mathbf{p}^j\|$,

rather than the magnitude of the difference, $\|\mathbf{p}^i - \mathbf{p}^j\|$. The blue color for “planar creepage” thus indicates that the traction magnitude did not change much in this scenario, whereas the arrows indicate a significant change of direction. This says that the pressure distribution remained nearly the same between cases $3 \rightarrow 5$, and underwent bigger changes between cases $4 \rightarrow 5$.

6.7 The use of the FASTSIM algorithm

The input-file `fastsim.inp` illustrates the use of the FASTSIM approach using the CONTACT program. A single Hertzian case with aspect ratio $a/b = 2$ is solved three times using different solution methods.

1. The first case uses the original CONTACT algorithm, by selecting $B = 0$ and $M = 0$.
2. The second case uses the FASTSIM algorithm in the recommended way, i.e. with parabolic traction bound. This is achieved by setting $B = 3$ and $M = 3$.
3. The third case uses the FASTSIM algorithm together with an elliptical traction bound: $B = 0$ or 2 , $M = 3$.

% second case: FASTSIM algorithm, parabolical traction bound (M=3, B=3)

```

3 MODULE
233100      P-B-T-N-F-S      PVTIME, BOUND , TANG , NORM , FORCE , STRESS
122320      L-D-C-M-Z-E      FRCLAW, DISCNS, INFLCF, MATER , RZNORM, EXRHS
100341      G-I-A-O-W-R      GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW , RETURN
 82000.     0.000      -0.000625  0.000625      FN, CKSI, CETA, CPHI
 0.000      0.400      30000.     CHI, DQ, VELOC
 0.280      0.280      82000.     82000.     POISS 1,2, GG 1,2
 1.000      1.000      1.000      KO_MF, ALFAMF, BETAMF
-3
44  44      8.000      4.000      1.100      MX,MY,AA,BB,SCALE

```

Figure 6.10 illustrates the tangential tractions that are obtained. It can be seen that the parabolical traction bound gives results that compare better to those of the half-space approach. The total forces compare quite well in this case: $F_y = -0.61$ for CONTACT, and -0.64 and -0.59 for FASTSIM with parabolic and elliptical traction bound respectively. Note that these results are affected by the discretisation that is used, and that no general conclusions can be drawn on basis of a single case. For more information on the accuracy of the FASTSIM approach (and USETAB, Polach’s method) when compared to CONTACT see [46]. For a comparison of parabolical and elliptical traction bounds and one versus three flexibilities see [51].

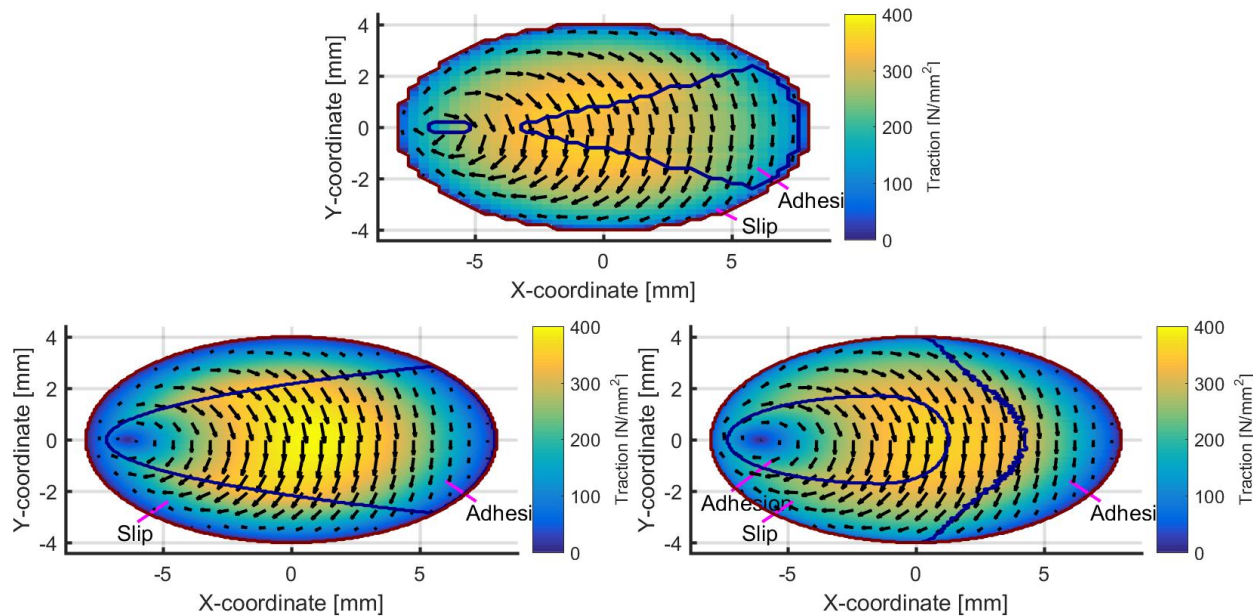


Figure 6.10: *Tangential tractions in a steady state rolling problem with large spin. Top: results for the CONTACT (half-space) algorithm on a 40×40 grid. Bottom: results for the FASTSIM algorithm with parabolical ($B=3$, left) and elliptical traction bound ($B=2$, right), computed on 200×200 grids.*

The pictures are created with the Matlab program `plot3d` (Section 7.2), in which the following settings are used:

```
s = loadcase('fastsim'); opt = plot3d;
opt.field='ptabs+vec'; opt.numvecx=22; opt.numvecy=12;
opt.external=NaN; opt.zrange=[0 400];
plot3d(s, opt); shading flat; axis equal;
```

The blue to yellow colors are obtained using the colormap `'parula'`. This is the default in Matlab version R2014b and higher. In older versions of Matlab, the colormap `'jet'` is used as the default instead.

6.8 Instationary problems: from Cattaneo to Carter

The example `"catt_to_cart.inp"` shows the computation of transient phenomena. This example concerns the situation described in [15, paragraph 5.2.2.5].

The test-case concerns the traction distribution arising in a wheel which is at rest at first and then starts accelerating. The geometry, material constants and normal load are chosen such that a circular contact area of radius 3.5 mm is obtained. After the initial Cattaneo shift the

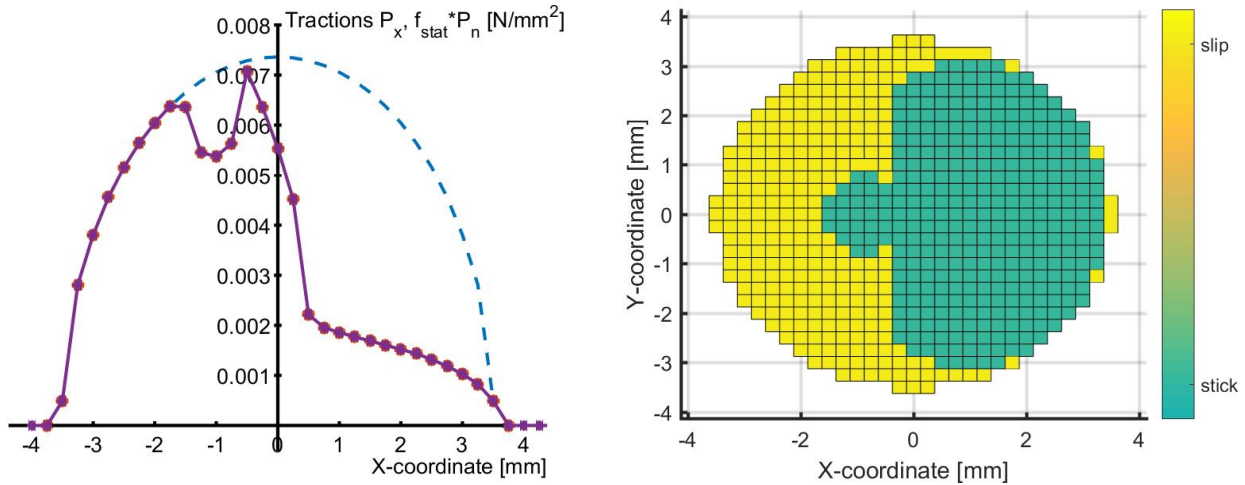


Figure 6.11: *Left: tangential tractions p_x at centerline $y = 0$ for the transient rolling problem “from Cattaneo to Carter” after “3 units” of time. Right: corresponding element division.*

spheres roll with a constant longitudinal force $F_x = -0.657$ prescribed without lateral and spin creepage. The results are written to `mat`-files after 4, 8, 12, 20 and 28 steps of rolling, corresponding to a rolling distance of 1, 2, 3, 5 and 7 *mm*, called “1–7 units of rolling” in [15].

Results of this example are presented in Figure 6.11. The graph on the left shows the tractions p_x along the centerline $y = 0$ after 3 units of time. The graph on the right shows the corresponding areas of slip and adhesion. These results are qualitatively different from those in [15], because of the higher resolution that can be used nowadays.

One specific aspect of the input-file `catt_to_cart.inp` is that it solves the same problem twice. The first sequence of 57 cases uses the moving coordinate system of option `T = 2`. In these cases the geometry is the same in all steps:

```

1 1
0.002963 0.000 0.002963 0.000 0.000 0.000
IBASE, IPLAN
B(I), I=1, 6
    
```

The next 57 cases solve the same problem using the world-fixed coordinate system of option `T = 1`. In this case the quadratic undeformed distance must be shifted to the right over a distance of δq in each step:

$$h(x, y) = b_1 \cdot (x - k \cdot \delta q)^2 + b_3 \cdot y^2 \tag{6.8}$$

With $\delta q = 0.25 \text{ mm}$ this yields in step $k = 4$ the following geometry:

```

1 1
0.002963 0.000 0.002963 -0.0059260 0.000 0.0029630
IBASE, IPLAN
B(I), I=1, 6
    
```

A larger grid is used (90×33 instead of 33×33 elements) in order to accommodate for all time steps with a single grid. And in the output the value CKSI is multiplied by $DQ = 0.25 \text{ mm}$.

The easiest way to create the corresponding input-file is via a small Matlab script that writes out the problem data per case.

The two approaches give practically the same results. This is shown by the red ($T = 2$) and blue lines ($T = 1$) in Figure 6.11 (left). To create this figure is a bit intricate. Loading and plotting the data for the first sequence is similar as before (explained further in Section 7.2):

```
r3=loadcase('catt_to_cart',13); % "r3" = rolling, 3 units == case 13
s3=loadcase('catt_to_cart',70); % "s3" = shift, 3 units == case 70
opt=plot2d; opt.yslc=0; opt.facpt=-1;
plot2d(r3,opt);
```

Adding the data for the second sequence requires a specific Matlab command:

```
plot(s3.x(1:end-12), -s3.px(13:end,17), 'b-*'); % row 17 == centerline y=0.
```

The reason for this is that this problem “s3” for the shift uses a different (larger) grid than problem “r3” (rolling). Using these specific indices for arrays x and px we shift the data 12 grid distances to the left.

A nice feature of the element division in Figure 6.11 (right) is that the exterior area is not shown coloured in the picture. This is achieved using the Matlab commands:

```
opt=plot3d; opt.field='eldiv'; opt.exterval=NaN; plot3d(r3,opt);
```

6.9 The calculation of subsurface stresses

The calculation of subsurface stresses is illustrated in the example `subsurf.inp`. This starts by defining the contact problem as usual, the main difference being that the S -digit is used. In the first case in the input-file $S = 3$, and new subsurface points are entered. Two blocks of subsurface points are defined using input option $ISUBS = 9$ (see also Section 5.9.2):

```
% subsurface points:
  2  1      MATFIL, OUTPUT
% first block of subsurface points:
  9      ISUBS
  1  1  15  NX, NY, NZ
% points x:
  0.0
% points y:
```

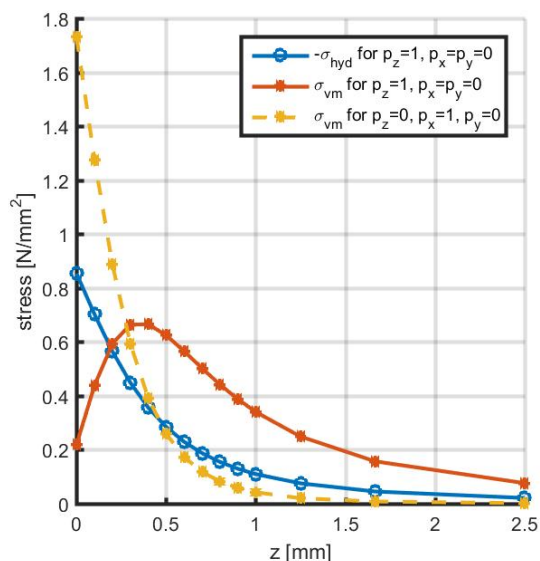


Figure 6.12: *Subsurface stresses in the half-space ($z \geq 0$), due to unit loadings on a square with unit sides in normal and tangential directions.*

```

0.0
% points z:
0.0  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9
1.0  1.25 1.667 2.5  5.0

% second block of subsurface points:
9          ISUBS
21  21  15  % NX, NY, NZ
...
0          ISUBS
    
```

When this case is computed, the surface tractions are solved first and subsurface stresses are evaluated immediately thereafter. Some aggregate results are printed to the output-file, whereas the main detailed results are put in a table for Matlab. This latter output is stored in the file `subsurf.0001.subs`; the first part of the name is the experiment-name used, and the middle part “.0001” is the case-number. The extension `.subs` is used for files containing subsurface results. This is prepared in such a way that it can be imported in Matlab at once.

The second case in the example has the S-digit set to 1. As such it re-uses the subsurface points defined in the first case of the input-file.

The contact problem in this example concerns the experiment described by Kalker in [15, paragraph 5.2.2.4]. It concerns the subsurface field resulting from a unit load in a square element with size $1 \times 1 \text{ mm}$. The first case concerns a load in normal direction, the second case concerns a normal plus tangential load. The subsurface stresses resulting from this

experiment are displayed in Figure 6.12. Note that Kalker’s σ_{ii} is $3\sigma_{hyd}$.

The Matlab commands used for producing this figure are (see Section 7.3):

```
% load results for cases 1 and 2, each using two blocks 'a' and 'b'
[s1a, s1b] = loadstrs('subsurf', 1);
[s2a, s2b] = loadstrs('subsurf', 2);
dif = diffstrs(s2a, s1a);
plot(s1a.z, -squeeze(s1a.sighyd(1,1,:)), '-o');
plot(s1a.z, squeeze(s1a.sigvm(1,1,:)), '-*');
plot(dif.z, squeeze(dif.sigvm(1,1,:)), 'r--*')
```

Here “diffstrs” is used in order to obtain the stresses due to the tangential traction alone. For more information type “help diffstrs” at the Matlab prompt.

The second block of points in the subsurface input specifies a 3D grid of points. This allow plots to be made such as shown in Figure 6.13. This plot is created with the Matlab commands:

```
[s2a, s2b] = loadstrs('subsurf', 2);
opt = plotstrs; opt.yslc = 0;
opt.typplot = 'contourf';
opt.cntrlvl = [0:0.02:0.08, 0.12:0.04:0.40];
plotstrs(s2b, opt);
set(gca, 'clim', [0 0.40]);
h=findobj(gcf, 'type', 'colorbar');
set(h, 'ylim', [0 0.40], 'ytick', opt.cntrlvl);
```

6.10 Dissimilar materials: Spence compression

The example `spence35.inp` illustrates the interaction between normal and tangential tractions when dissimilar materials are used.

We consider an elastic sphere with Poisson’s ratio ν . It is pressed onto a rigid half-space in the presence of dry friction with coefficient μ_{stat} . There is rotational symmetry; the contact area C and adhesion area H are circular, and the ratio of the contact radius and the adhesion radius is a constant depending only on μ_{stat} and ν .

The problem was solved semi-analytically by Spence [32]. It was used as a test-problem a.o. by Kalker [15, paragraph 5.3.1] and Willner [56]. Solutions for the example are shown in Figure 6.14.

The example uses a sphere radius $R^{(1)} = 121.5\text{ mm}$, modulus of rigidity $G^{(1)} = 0.5\text{ N/mm}^2$, Poisson’s ratio $\nu^{(1)} = 0$ and coefficient of friction $\mu_{stat} = 0.2986$. The combined material parameters (Section 5.1.1) are $G = 1, \nu = 0$ and $K = 0.5$, which is the maximal difference

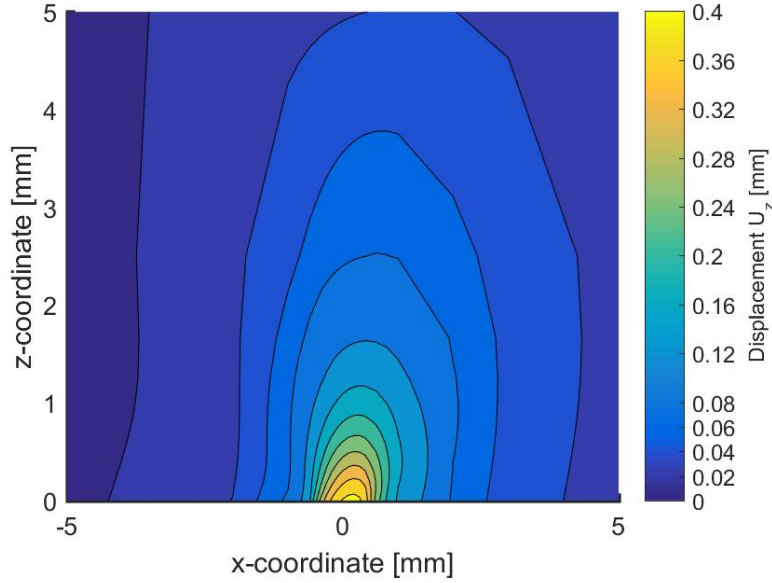


Figure 6.13: Normal displacements u_z at $y = 0$ inside body 1 ($z \geq 0$), due to unit loads $p_n = 1$ and $p_x = 1$ on a square with unit sides.

that can be obtained. With these parameters, the radius of the adhesion area is half that of the radius of the contact area [32].

To understand the results of Figure 6.14 we consider the half-space solution by Boussinesq [2]. This solution states that a localized compressive normal load P acting on body (1) invokes tangential displacements $\mathbf{u}_t^{(1)}$. At the surface $z = 0$, these tangential displacements are directed inwards, towards the location where P is applied.

In the examples of the previous sections we used identical elastic constants for bodies (1) and (2). In those cases the tangential displacements were identical for the two bodies, thus not creating any displacement difference \mathbf{u}_t (cf. equation (2.3)). In the current example body (2) is rigid, $\mathbf{u}^{(2)} = \mathbf{0}$, and consequently a normal load P *does* invoke a tangential displacement difference between opposing surface particles of bodies (1) and (2). This displacement difference is counteracted by frictional surface tractions $\mathbf{p}_t^{(1)}$ acting on body (1), directed radially outwards w.r.t. where the load P is applied.

Finally, the tangential tractions also invoke normal displacements $u_n^{(1)}$. These add to the profile of the sphere, such that the the normal pressure distribution is also modified. This is shown by the traction bound in Figure 6.14 (right). At the center $r = 0$, the normal pressure is 1.31 times higher than for the Hertzian pressure distribution, whereas the pressures are reduced at the outer side of the contact area.

In the results presented here, a grid of 91×91 elements of size $\delta x = \delta y = 0.1 \text{ mm}$ is used. The load is gradually increased in 69 steps as $F_n = (k/90)^3$ for $k = 2 - 70$. These cases are connected to each other via the definition of the (micro) slip velocity (equation (2.15)).

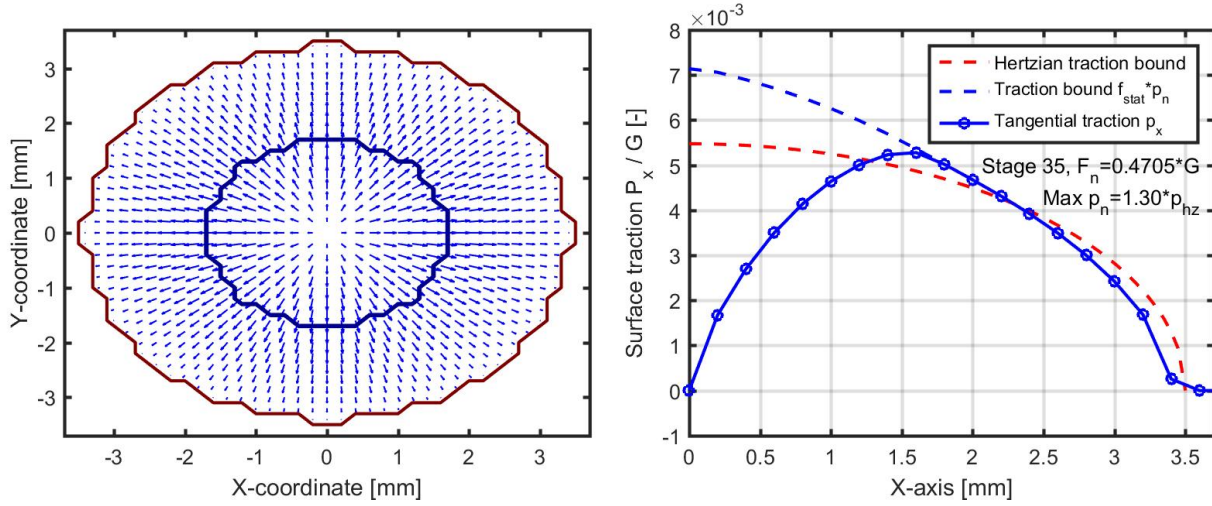


Figure 6.14: *Spence compression. Left: element divisions and direction of tangential tractions. Right: tangential tractions as function of radial position r .*

The relevance of this is clearly illustrated by considering that the load is imposed at once, and comparing the results. Figure 6.15 shows the results of this scenario and for simulations where the load is applied in four or seven steps.

By definition, surface particles in the adhesion area are locked in to the corresponding particles of the other body. In the Spence problem, this means that their tangential position does not change. In CONTACT this is computed via the *shift*, the time-increment $\mathbf{u}_t - \mathbf{u}'_t$ (note that $\mathbf{W}_t = \mathbf{0}$). Now if the sphere was not yet loaded at the previous time-instance t' , its tangential displacements \mathbf{u}'_t were zero too, and equation (2.15) gives $\mathbf{u}_t = \mathbf{0}$ in the adhesion area at the new time instance. This is reflected in the horizontal part of the light-blue line in Figure 6.15 (right) for $r < 0.9$. The shift $\mathbf{u}_t - \mathbf{u}'_t$ is over-estimated, which increases the tangential tractions \mathbf{p}_t (Figure 6.15, left) and reduces the size of the adhesion area.

Similar effects are visible when using four or seven steps. The results quickly tend to the correct values, but significant oscillations are visible. These are due to adding multiple discretisation elements to the adhesion area per time step. Note that for the grid discretisation that is used, the adhesion area has a radius of about 17 elements in the final stage.

6.11 Steady rolling of two viscoelastic cylinders

The next example employs the viscoelastic material model in CONTACT for two identical cylinders with parallel axes in rolling contact [53]. Both cylinders have a radius $R^{(a)} = 100 \text{ mm}$ (combined radius $R = 50 \text{ mm}$) and consist of polymer PA6, a viscoelastic material that is characterized by a standard linear solid model with an initial Young’s modulus $E_g = 3200 \text{ MPa}$, Poisson’s ratio $\nu = 0.4$, spring ratio $FG = f = 1$ (final Young’s modulus $E_r =$

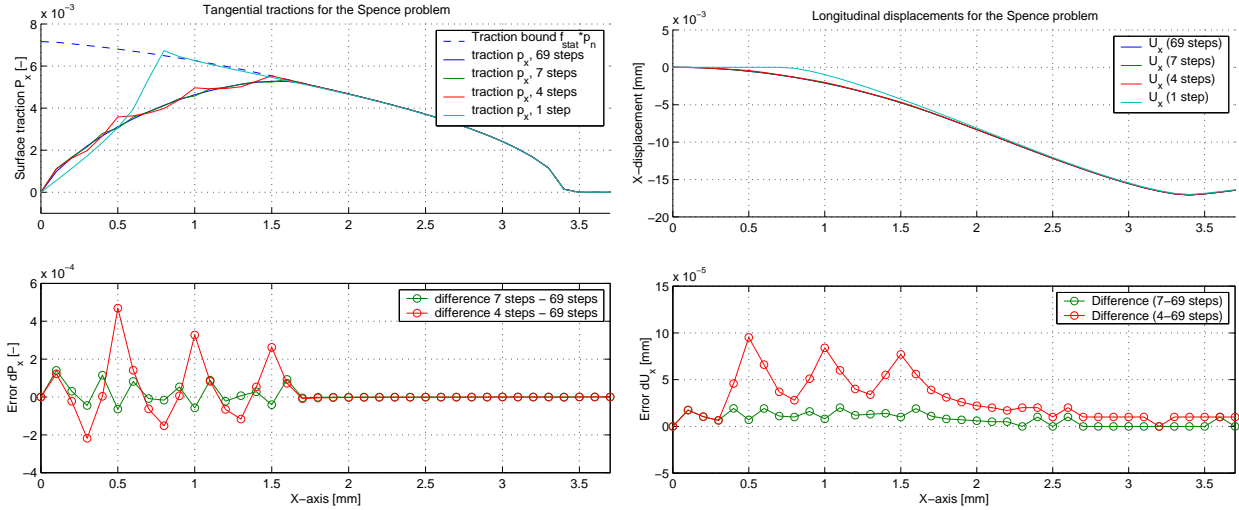


Figure 6.15: *Spence compression using too few stages in applying the load. Left: tangential tractions, right: tangential surface displacements as function of radial position r .*

1600 MPa) and creep relaxation time $TC = \tau_C$ varying between three different test-cases. The corresponding initial modulus of rigidity that is input to CONTACT is (cf. last row of Table 5.1)

$$GG = G_g = E_g/2(1 + \nu) = 1143 \text{ MPa}. \quad (6.9)$$

The cylinders are pressed together with a normal force per unit length of $F_n = 600 \text{ N/mm}$, and the coefficient of friction is $\mu = 0.3$. The relative traction force is taken as $F_x/\mu F_n = -0.6$.

In case of very slow relaxation (relaxation time $\tau_C \rightarrow \infty$), the viscoelastic effect is largely unnoticed. The viscoelastic problem reduces to an elastic problem with initial modulus of rigidity $G_\infty = G_g$. At the other extreme, $\tau_C \downarrow 0$, the material responds almost instantaneously, elastically with $G_0 = G_r = E_r/2(1 + \nu)$ (cf. equations (5.3) and (5.4)). The corresponding shear modulus is obtained from $G_r = G_g/(1 + f)$, which gives $G_r = G_g/2 = 571.5 \text{ MPa}$. In Figure 5.1 (left), this is interpreted as that the two springs have equal strength, such that the total stiffness is halved.

According to the 2D Hertz solution (e.g. equation (6.4)), the half-width of the contact area a_h and the maximum normal pressure p_0 are given as

$$a_h = \left(\frac{4RF_n(1 - \nu)}{\pi G} \right)^{1/2}, \quad p_0 = \left(\frac{F_n}{\pi R} \frac{G}{1 - \nu} \right)^{1/2}. \quad (6.10)$$

For $G_g = G_\infty$ (stiff, glassy) and $G_r = G_0$ (soft, rubbery), equations (6.10) give $a_h = 4.48 \text{ mm}$, $p_0 = 85.3 \text{ MPa}$ and $a_h = 6.33 \text{ mm}$, $p_0 = 60.3 \text{ MPa}$, respectively.

The example input file `visc_cylindr.inp` contains four viscoelastic cases with varying τ_C (0, 0.009, 0.045 and 2.0 s), and two elastic cases with varying G (571.5 and 1143 N/mm²). The viscoelastic cases are defined conform the following input:

```

3  MODULE
203120      P-B-T-N-F-S  PVTIME, BOUND , TANG  , NORM  , FORCE  , STRESS
022120      L-D-C-M-Z-E  FRCLAW, DISCNS, INFLCF, MATER  , RZNORM, EXRHS
001341      G-I-A-O-W-R  GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW  , RETURN
  100   100   30      1      1e-5   MAXGS , MAXIN  , MAXNR  , MAXOUT, EPS
600000.   -0.600   0.000   0.000          FN, FX, FY, CPHI
  0.300   0.300          FSTAT, FKIN
  0.000   0.160   1000.          CHI, DQ, VELOC
  0.400   0.400   1143.   1143.          POISS 1,2,  GG 1,2
  1.0     1.0     0.009   0.009          FG 1,2, TC 1,2
  3
99  1   -7.920   0.000   0.160   1000.0  MX,MY,XC1,YC1,DX,DY
  1     1
%  QUADRATIC UNDEFORMED DISTANCE
0.010   0.000   0.000   0.000   0.000   0.000   B(I), I=1, 6

```

Things to note here are:

- The M (MATER) control digit is set to 1, i.e. for viscoelastic materials.
- The normal force per unit length F_n is translated to FN by multiplying with δy .
- The potential contact area is defined similarly as in the Carter/Fromm example, Section 6.2. In the y -direction, the contact region consists of one element with a large size of $\delta y = 1000 \text{ mm}$.
- The undeformed distance between the cylinders equals $h(x, y) = R - \sqrt{R^2 - x^2}$. For $x \ll R$ this is approximated as $h \approx x^2/2R$, and entered as a quadratic profile with IBASE = 1.

The results for the test-cases are shown in Figure 6.16. The lowest pressures and shear tractions are found if the viscoelastic effects take place immediately, relaxation distance $V\tau_C = 0$. This is the softest material behaviour, resulting in the widest contact area with semi-width $a_h = 6.34 \text{ mm}$, which is also obtained in the elastic case using $G = G_0$. For increasing τ_C the contact area shrinks, first at the trailing edge and later also at the leading edge of the contact area. The tractions increase correspondingly, until ultimately the traction profile equals that of the elastic case with $G = G_\infty$. For intermediate values of τ_C , an asymmetric traction profile is found, which is a typical feature of viscoelastic contact. At the inlet additional pressure is needed in order to overcome the creep relaxation and avoid interpenetration of the two surfaces. At the outlet, it takes a while for the material to relax back to its original form such that less pressure is needed there.

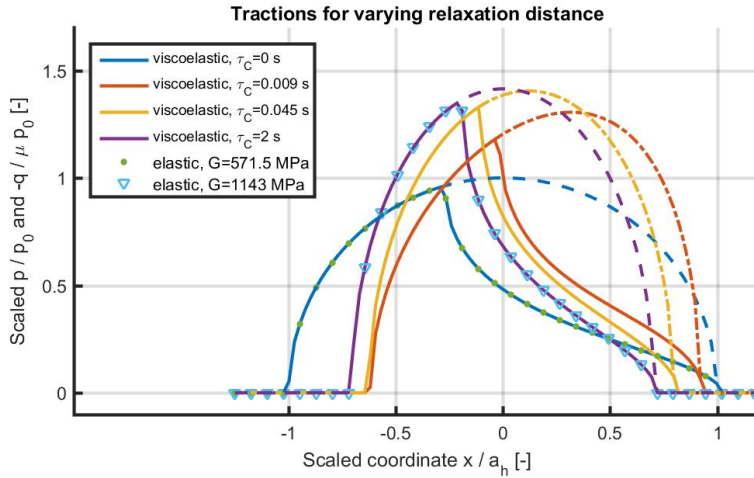


Figure 6.16: Results for the rolling contact of two viscoelastic cylinders. Distribution of normal and tangential traction $\mu_{stat} p_n(x)$ and $p_x(x)$ for different relaxation distances $V\tau$. The axes are normalized with $a_h = 6.34 \text{ mm}$ and $p_0 = 60.3 \text{ MPa}$.

6.12 Transient rolling with velocity dependent friction

Besides Coulomb’s law of dry friction, several extended friction laws are provided as well as described in Section 5.2. Particularly for experimenting with “falling friction”, that is, friction formulations where the friction coefficient μ decreases with increasing local slip velocity $\|\mathbf{s}_t\|$. The results of this are demonstrated in example `veldep_fric`, which corresponds to the test case that is used in the paper [48].

The example concerns the 2D rolling contact of a cylinder on a plane. The geometry and normal pressure are set such that a contact patch size of 8 mm is obtained. Three different friction laws are provided in the input-file:

```

422020          L-D-C-M-Z-E  FRCLAW, DISCNS, INFLCF, MATER , RZNORM, EXRHS
...
% Linear+const slip-velocity dependence, L=2:
% 0.180  0.120  100.  0.0  0.0  FKIN, FLIN1, SABSH1, FL2, SH2
% Rational slip-velocity dependence, L=3:
% 0.180  0.060  100.  0.060  0.3162  FKIN, FRAT1, SABSH1, FR2, SH2
% Exponential slip-velocity dependence, L=4:
  0.180  0.120  100.  0.0  0.0  FKIN, FEXP1, SABSH1, FE2, SH2
  0.000  1.000  MEMDST, MEM_S0

```

The last one of these is used by setting $L = 4$, and corresponds to equation (5.12).

There appear to be multiple solutions to the discrete system. This is demonstrated by calculating three different cases consecutively:

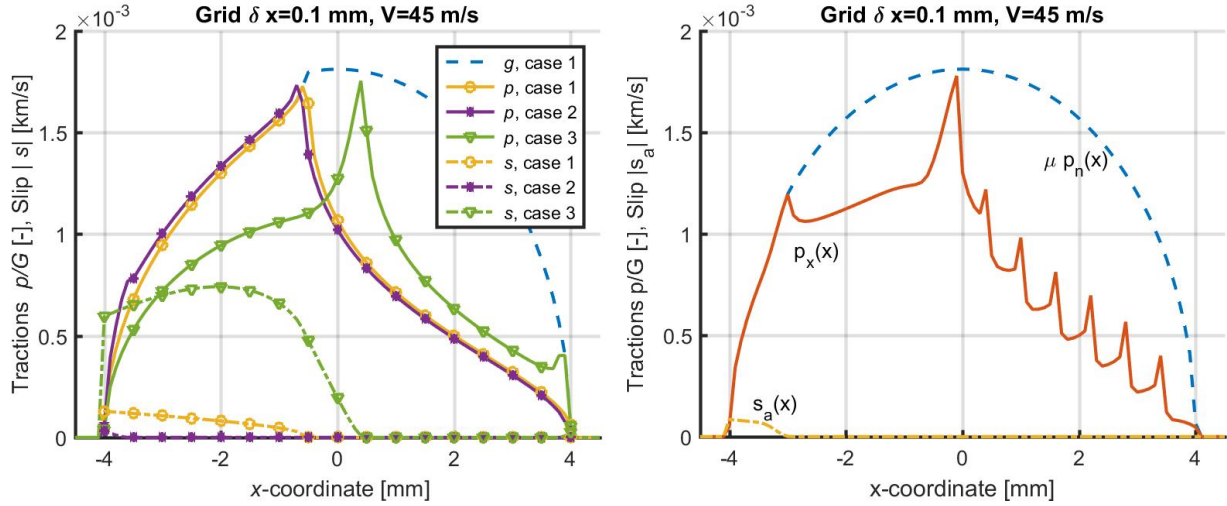


Figure 6.17: Results obtained with slip-velocity dependent friction law $L = 4$. Left: demonstration of multiple solutions to the same case, depending on initial estimate to the slip. Right: traction distribution after 49 time steps, with peculiar peaks travelling from right to left through the contact area.

- “case 1”: solve for a steady state solution ($p = p(x), u = u(x), \partial u/\partial t = 0$) for one creepage ξ ;
- “case 2”: starting from the steady state for creepage ξ , perform a single time step for a slightly perturbed creepage $\tilde{\xi}$, with initial estimate $s^0 = 0$;
- “case 3”: starting from the steady state for creepage ξ , perform a single time step for a slightly perturbed creepage $\tilde{\xi}$, with initial estimate $s^0 = V \text{sgn}(\xi)$;

The “true” creepage in case 1 is set at $\xi = -0.1\%$. The perturbation in cases 2 and 3 consists of decreasing the creepage to $\tilde{\xi} = -0.09999\%$.

Figure 6.17 (left) shows the results of this experiment, using grid step size $\delta x = 0.1$ mm and rolling velocity $V = 45$ m/s. These results show that two different solutions are possible for exactly the same contact problem. If s is approximated from below (“case 2”), several elements are moved to the adhesion area. If s is approximated from above (“case 3”), a much higher slip is found. And if the creepage is not perturbed then a third solution is possible: the steady state solution that is in between the two extremes.

Note that Figure 6.17 shows the absolute slip s_a , in units of km/s , in order to scale the values to the same range as those of p/G . The largest slip in the steady solution is approximately -0.13 m/s (-0.3%).

The qualitative behaviour of the numerical results is illustrated further in Figure 6.17 (right). This concerns the computed traction distribution after calculation of 49 time steps (4.9 mm rolling distance). In each time step the initial estimate $s^0 = -V$ (high slip) is used.

The result for the first time step is already shown in Figure 6.17 (left). In the second time step the slip velocity drops, most elements are transferred to the adhesion area, and the two peaks in the traction distribution move one grid space to the left. In the next few steps the slip velocity and slip area increase gradually and the peaks travel further to the left. After a while a critical configuration is reached, the slip velocity jumps to a high value, a new peak is found at the leading edge of the contact area, and the process starts to repeat itself.

The background of these peculiar peaks is analysed in [48]. They are explained by a discontinuity in time of the elastic displacements field. Traction builds up until a certain maximum is attained, after which the elastic field jumps to a new state in “zero time”. This behaviour is attributed to the use of a quasi-static approach (ignoring inertia effects) and to the friction law employed, particularly to the use of the instantaneous slip velocity s_a . The peaks can be suppressed by introducing friction memory with a memory length d_c of several μm . The peaks are thus real as far as the numerical model is concerned, but whether they occur in reality is still unknown.

Chapter 7

Matlab plot-programs

Several Matlab scripts are provided for visualisation of the output of the CONTACT program:

1. CONTACT writes its results to tables in the files `<experim>.<ncase>.mat` (surface tractions) and `<experim>.<ncase>.subs` (subsurface stresses);
2. In Matlab you load these results with the scripts `loadcase` and `loadstrs`;
3. The results are plotted with the scripts `plot2d`, `plot3d` and `plotstrs`.

The scripts are meant for basic inspection tasks only; for more sophisticated and stylish pictures you can refine our scripts or use Matlab’s command interface.

An additional script is provided for the overall output quantities:

4. The basic data from the output-file `<experim>.out` (creepages, total forces and overall output quantities) can be loaded into Matlab with the scripts `parse_out1` and `parse_out3`, for modules 1 and 3 respectively;

No specific programs are provided for further processing and visualization of these overall results. You can type `help parse_out3` at the Matlab command prompt for information about this script, see Section 6.4 for an example of its use.

7.1 Prerequisites

Usage of the plot programs requires a license to the (commercial) Matlab package. As an alternative you may work with GNU Octave, a free software that is quite similar to Matlab. Another alternative might be Gnuplot, but this requires a larger development effort for creating new plotting scripts.

The Matlab search path must be adjusted such that our plotting programs can be found. This is done in Matlab with the `addpath` command, e.g.

```
>> addpath('C:\Program Files (x86)\KalkerSoftware\contact_v18.1\matlab');
```

This command may be put in a file "startup.m" in your working directory, in your personal overall startup-file ("My Documents\MATLAB\startup.m") or in a system-wide configuration-file (" \langle MATLAB \rangle \toolbox\local\pathdef.m").

You can check whether the path is set correctly by typing `help contact_v18.1\matlab`. This should show the following output:

Matlab scripts for visualisation of output of the CONTACT program (v18.1).

Loading results into Matlab.

```
loadcase    - load the results for one case for a given experiment name.
diffcase    - compute the difference of results for two cases
loadstrs    - load the results for a subsurface stress calculation.
diffstrs    - compute the difference of results for two cases w.r.t.
              subsurface stress calculations.
```

```
parse_out1  - sample script to read output of the contact patches for
              wheel-rail contact cases (module 1) from an .out-file.
parse_out3  - sample script to read the creepages and forces of generic
              contact cases (module 3) from an .out-file.
```

Visualizing the results.

```
plot2d      - 2D plots of tractions for rows or columns of the contact area
plot3d      - 3D plots of various quantities for the entire contact area.
plotstrs    - plot sub-surface displacements and stresses.
```

Working with wheel/rail profiles.

```
read_profile - generic routine for reading profiles
read_simpack - routine for reading Simpack prr/prw files
read_miniprof - routine for reading Miniprof ban/whl files
modify_profile - routine for making some profile adjustments
write_simpack - routine for writing Simpack prr/prw files
write_miniprof - routine for writing Miniprof ban/whl files
```

7.2 Inspecting the surface stresses

7.2.1 Loading the results into Matlab

The contents of the `.mat`-files (surface tractions) are specified in Section A.4. These are loaded with the script `loadcase`:

```
>> s=loadcase('mbench_a22_left',13)
```

```
s =
```

```
    meta: [1x1 struct]
      mx: 51
      my: 41
      xl: -5.1000
      yl: -4.3000
      dx: 0.2000
      dy: 0.2000
      chi: 0
      dq: 0.2000
x_offset: []
y_offset: []
      x: [1x51 double]
      y: [1x41 double]
t_digit: 3
  mater: [1x1 struct]
  fric: [1x1 struct]
  veloc: 2000
  eldiv: [41x51 double]
    h: [41x51 double]
  mu: [41x51 double]
  pn: [41x51 double]
  px: [41x51 double]
  py: [41x51 double]
  un: [41x51 double]
  ux: [41x51 double]
  uy: [41x51 double]
  srel: [41x51 double]
  shft: [41x51 double]
  trcbnd: [41x51 double]
```

With this command the results for the thirteenth case of the Manchester benchmark example are loaded (Section 6.5). The result is a structure of which the different components can easily be recognized. The members `fric` and `mater` are structs themselves, that contain the data for the friction law and material parameters respectively.

For wheel/rail contact using module 1, a single case may result in multiple contact patches. In such cases, `mat`-files are created for each contact patch separately, with letters ‘a’, ‘b’, etc. appended to the case number.

- By default, `loadcase` will load all the patches and return these in separate structures `s1a`, `s1b`, etc, when called like

```
[s1a, s1b, s1c] = loadcase('mbench_case_1a');
```

This will try to load output for three patches for the first case of experiment `mbench_case_1a`. The output structs will be empty if no corresponding `mat`-file is found.

- It is also possible to retrieve output for a single patch only, by specifying the patch number:

```
s1c = loadcase('mbench_case_1a', 1, 3);
```

The second argument “1” is the case number, the third argument “3” indicates the requested patch number.

7.2.2 Plotting results for the entire (3D) contact area

Several standardised contour- and surface-plots are provided by the script `plot3d`. This script works on the solution structure provided by `loadcase` and further uses an options structure to configure the plot. The options structure is initialized by `plot3d` itself:

```
>> opt = plot3d
opt =
    field: 'ptabs+vec'
  exterval: NaN
  rw_surfc: 'none'
    view: 'default'
   zrange: []
  ixrange: 'auto'
  iyrange: 'auto'
 numvecx: 15
 numvecy: 15
  vecscale: []
  veccolor: []
  vecwidth: []
  addeldiv: []
  eldivcol: [2x3 double]
   addplot: 0
  colormap: 'parula'
```

Note that colormap `'parula'` was introduced in Matlab version R2014b. In older versions of Matlab, the colormap `'jet'` is used as the default.

The main option is the `field` of the solution structure to be plotted. Possible values are (see `"help plot3d"`):

- `'eldiv'`, `'eldiv_spy'`, `'eldiv_contour'`: different ways of presenting the element division. See Figure 6.11 (right);
- `'h'`: the undeformed distance of the two bodies;
- `'pn'`, `'px'`, `'py'`: the normal and tangential tractions acting on body (1), i.e. the rail in module 1;
- `'mu'`: the actual local coefficient of friction;
- `'ptabs'`, `'ptarg'`: the magnitude and direction of the tangential tractions;
- `'ptvec'`: a vector-plot of the tangential tractions;
- `'ptabs+vec'`: show magnitude and direction of tangential tractions in a single plot. See Figures 6.9 and 6.10;
- `'un'`, `'ux'`, `'uy'`: the displacement differences in normal and tangential directions;
- in shifts ($T \leq 1$):
 - `'sx'`, `'sy'`: the components of the local shift (slipped distance);
 - `'shft'`: the magnitude of the local shift;
 - `'shft+vec'`: show magnitude and direction of the local shift in one plot;
- in rolling ($T \geq 2$):
 - `'sx'`, `'sy'`: the components of the relative micro-slip velocity `srel`;
 - `'sabs'`, `'srel'`: the magnitude of the absolute/relative micro-slip velocity;
 - `'sabs+vec'`, `'srel+vec'`: magnitude and direction of the micro-slip in one plot.
- `'fricdens'`: the frictional power density.

Option `rw_surfc` governs how the contact surface is plotted: using `'none'` for the typical flat view, `'prr'` or `'both'` for using rail coordinates, or `'prw'` for using the wheel surface view, see Figure 6.7 for an example. The latter options require that rail and/or wheel profiles or filenames are provided, e.g.

```
plot3d( s1a, opt, 'MBench_UIC60_v4.prr', 'MBench_S1002_v3.prw' );
```

The meaning of the other options is:

exterval	The value to be plotted at points of the exterior area. Particularly useful for plotting fields 'eldiv', 'h' and 'ptabs+vec', using <code>opt.exterval=NaN</code> .
view	The view direction, e.g. [30 30] (rotation about z -axis and elevation above the plane, in degrees). You may use [0 90] to get a 2D color plot or 'rail' for view [90 -90].
zrange	The range of the z -axis to be displayed.
ixrange	The selection of the elements [1,mx] in x -direction to be displayed in the plot.
iyrange	The selection of the elements [1,my] in y -direction to be displayed in the plot.
numvecx	The maximum number of vectors to be displayed in x -direction. If necessary, every 2 nd , 3 rd , etc. grid point is left out of the plot.
numvecy	Same as numvecx for y -direction.
vecscale	Manual scaling factor for vectors on vector-plots. Length of the vector in mm that will be used for a stress of $1 N/mm^2$;
veccolor	Color specification for vectors on vector plots. Default: 'b' for field 'ptvec', 'k' for 'ptabs+vec'.
vecwidth	Line width for vectors on vector plots.
addeldiv	Add contour lines around adhesion area and contact area, à la 'eldiv_contour' (only in 2D plots).
eldivcol	Set colors of lines around adhesion area and contact area for 'eldiv_contour' and 'addeldiv'. Default: dark blue and dark red.
eldivwid	Line width for contours of element division.
addplot	Typically <code>plot3d</code> clears the figure before creating a new plot. Sometimes it is convenient to turn this off and add to an existing plot.
colormap	This changes the colormap for 3D plots. Special values are 'none' and 'black', which create a black-and-white mesh plot instead of a coloured surface.

7.2.3 Plotting results for 2D cross-sections

The script `plot2d` provides a means for plotting the results of 2D calculations (i.e. with `MY=1`) or plotting 2D slices for 3D calculations.

This script is primarily interested in the tangential surface tractions `px/py`, and further plots the traction bound `trcbnd` as well ($= \mu_{stat} p_n$ when using Coulomb friction). It uses an options structure that is initialized again by the script itself:

```
>> opt=plot2d
opt =
    orie: 2
    pxory: 'x'
    xslc: 0
    yslc: 0
    negpn: 1
    facpt: 1
    xlim: []
    ylim: []
    plim: []
    pn_linestyle: '--'
    pt_linestyle: '-o'
```

The meaning of the options is:

- | | |
|--------------|--|
| orie | plot columns of the potential contact area (1, $x = \text{const}$) or rows (2, $y = \text{const}$, default); |
| pxory | plot tangential tractions <code>px</code> ('x', default) or <code>py</code> ('y'); |
| xslc | x -coordinate(s) for a column-wise plot (<code>orie</code> = 1). Can be an array of values for plotting multiple slices at once. Note: the grid column(s) closest to <code>xslc</code> is/are used. Default: 0.0; |
| yslc | y -coordinate(s) for a row-wise plot (<code>orie</code> = 2), see <code>xslc</code> ; |
| negpn | flag for the vertical range to be plotted. <ul style="list-style-type: none"> • 1 = show positive traction bound $\mu_{stat} p_n$; • 0 = show positive and negative traction bounds; • -1 = show negative traction bound $-\mu_{stat} p_n$. |
| facpt | multiplication factor for negating <code>px/py</code> : 1 for plotting the tangential tractions themselves, -1 for plotting their negative <code>-px</code> or <code>-py</code> ; |

xlim	<i>x</i> -range for a row-wise plot (<i>orie</i> = 2). Default: [<i>xl-dx</i> , <i>xh+dx</i>];
ylim	<i>y</i> -range for a column-wise plot (<i>orie</i> = 1). Default: [<i>yl-dy</i> , <i>yh+dy</i>];
plim	vertical range of the plot. The default depends on <i>negpn</i> . For <i>negpn</i> = 1 it is the range [<i>0</i> , <i>pmax</i>] extended a little on both sides;
pn_linestyle	Matlab linestyle for the traction bound;
pt_linestyle	Matlab linestyle for the tangential tractions.

The use of this script is illustrated in the Cattaneo and Carter2D examples: Sections 6.1 and 6.2, Figures 6.1 and 6.2.

7.3 Inspecting subsurface stresses

7.3.1 Loading the results into Matlab

The contents of the `.subs`-files (subsurface stresses) are specified in Section A.5, see further Section 5.9.3 for a description of the physical quantities. These are loaded with the script `loadstrs`:

```
>> sub28 = loadstrs('spence35', 28)
sub28 =
    nx: 45
    ny: 1
    nz: 1
  npoints: 45
    x: [45x1 double]
    y: 0
    z: 1.0000e-06
    ux: [45x1x1 double]
    uy: [45x1x1 double]
    uz: [45x1x1 double]
  sighyd: [45x1x1 double]
  sigvm: [45x1x1 double]
```

Two complications are that there can be multiple grids of points in a single subsurface stress calculation, and that each grid can be three-dimensional. The former is handled by multiple output-arguments for function `loadstrs`:

```
[blk1, blk2, blk3] = loadstrs('subsurf2', 1);
```

The latter means that the output arrays such as `ux`, `uy`, etc. are three-dimensional. These cannot be plotted directly, you select the appropriate slices by providing indices (e.g. `ux(1,1,:)`) and restructure the size of the array using the `squeeze` command.

7.3.2 Plotting the subsurface stresses

The subsurface stresses are plotted with the command `plotstrs`. It works with an options structure like the previous scripts:

```
>> opt=plotstrs
opt =
    field: 'uz'
      dir: 'y'
    xslc: 0
    yslc: 0
    zslc: 0
  addplot: 0
  typplot: 'contourf'
  cntrlvl: 'auto'
    scale: 'linear'
  colormap: 'parula'
```

The quantities that can be plotted are entered in the `field` option:

- `'ux'`, `'uy'`, `'uz'`: elastic displacements in the upper ($z > 0$) or lower body ($z < 0$);
- `'sighyd'` or `'hydro'`: the mean hydrostatic stress σ_{hyd} , see equation (5.34);
- `'sigvm'` or `'mises'`: the von Mises stress σ_{vm} (equation (5.36)).
- In case the full stress tensor is exported ($A = 2$):
 - `'sigxx'`, `'sigxy'`, `'sigxz'`, `'sigyy'`, `'sigyz'`, `'sigzz'`: the components σ_{ij} of the stress tensor `SIGMA` at the subsurface points, see equation (5.33);
 - `'sigma1'`, `'sigma2'`, `'sigma3'`: the principal stresses $\sigma_1, \sigma_2, \sigma_3$ (page 72);
 - `'sigtr'` or `'tresca'`: the maximum shear stress $\sigma_1 - \sigma_3$ of the Tresca criterion (equation (5.39)).

Note that the subsurface stresses are computed for three-dimensional structured grids, called “blocks” in the CONTACT input. The corresponding options are:

dir sets the orientation of the 2D slice to be viewed: `'y'` for a slice parallel to the Oxz -plane (default), `'x'` for a slice parallel to Oyz and `'z'` for a slice with constant depth z , parallel to the Oxy plane;

yslc sets the y -coordinate for an Oxz -plot (`dir='y'`). Default 0.0;

xslc, zslc x - and z -coordinates for `dir='x'` and `dir='z'`.

The other options control the outlook of the plot:

typplot make a color plot (`'surf'`), contour line plot (`'contour'`) or filled contour plot (`'contourf'`);

cntrlvl list of values for which contour lines are desired;

scale use a linear (`'linear'`) or logarithmic (`'log'`) scale for the colors on the plot;

colormap this changes the colormap used, e.g. `'parula'`, `'jet'`, `'hot'`.

A filled contour plot is shown in the subsurface stress example in Figure 6.13 of Section 6.9.

Chapter 8

The CONTACT library

The computational part of CONTACT can be interfaced with other programs and steered from there. This works through the so-called CONTACT library as illustrated schematically in Figure 8.1. The library is a dll (Windows) or so-file (Linux) that provides a well-defined interface. This interface can be accessed from Matlab as well.

8.1 Result elements and contact problems

The library is set up to compute the evolution of multiple contact problems that can be grouped in different ways.

Contact problems are grouped together for “result elements”. CONTACT doesn’t care what these result elements are about. So it is possible to define one result element per wheel (“rail-wheel pair”) that’s associated with multiple contact patches. In module 3, it’s equally possible to define one result element for the tread and another for flange contact. Likewise it’s possible to define one result element for a roller pair using coarse grid resolution and another result element for the same roller pair using fine grid resolution.

Result elements are identified by an integer number `ire` below 1000. The numbering does not have to be consecutive. Each result element can have one or more “contact problems” associated with it, created automatically when using module 1, or defined by the user when using module 3. Contact patches are identified by an integer number `icp` below 10, and these also don’t have to be numbered consecutively.

Most data are configured and stored separately for each contact problem. Therefore practically all subroutines start with the arguments `ire` and `icp`. The code `icp = -1` is used to indicate a task working on all patches of a result element. Note that the memory requirement is proportional to the number of contact problems that are stored simultaneously.

The number of cases computed can be different for each contact problem. The cases for different contact problems can be interlaced in any way as well. Each contact problem is

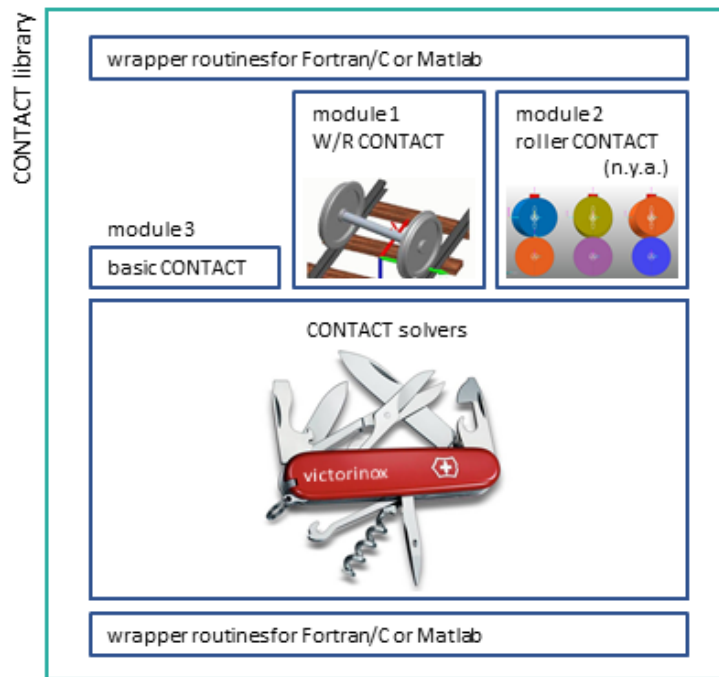


Figure 8.1: *Software architecture for the CONTACT library.*

solved independently, using its own internal data and possibly its own previous state. Two consecutive cases for the same contact problem are governed by the same rules as consecutive cases in the input file. For instance the grid can be changed between steady state cases but must stay the same in transient scenarios.

8.2 Data units and sign conventions

The CONTACT library can work with different data units. This is mainly a cosmetic aspect of the interface. Appropriate scaling is used between the values provided and the values stored internally; internally the same units are used as before. An important part concerns the sign conventions of the inputs (creepage) and the results (traction, force, slip).

CONTACT's unit convention The primary unit convention is the same as the one used throughout this document:

- Body 1 is the upper body, $z > 0$, and body 2 the lower one, $z < 0$.
- Stresses and forces are acting *on* body 1, and relative displacements and velocities are those *of* body 1 *relative to* body 2.

- Lengths are in mm and areas in mm^2 . Forces in N and stresses in N/mm^2 . Angles are in rad in the calculations, but may be printed in $^\circ$ for easier interpretation. Time is in s and velocities are in mm/s .

SI unit convention The main alternative is to use SI units everywhere. This uses the same choices as CONTACT for identifying the bodies and defining the signs of stresses, velocities, displacements and velocities.

- Lengths are in m and areas in m^2 . Forces in N and stresses in N/m^2 . All angles are in rad . Times are in s and velocities in m/s .

SIMPACK's unit convention The SIMPACK program uses SI units but uses a different way of designating the two bodies.

- Body 1 is the rail body, and body 2 is the wheel. These are respectively the upper and lower bodies, with the positive z -axis pointing into the rail.
- Stresses and forces are acting *on* body 2, the wheel. Relative displacements and velocities are those of the rail compared to the wheel.

The unit convention is changed using subroutine `cntc_setflags` by setting flag `CNTC_if_—units` to the appropriate value. This is typically done at the start of the calculation, for all contact problems/result elements that are used.

Wheel and rail profiles must be given in mm regardless of the unit convention that's used.

8.3 Interface routines

The interface of the routines is specified in detail in the Matlab functions. These are shown using Matlab's `help`-feature:

```
>> help matlab_intf
```

```
... provides an overview of the interface routines ...
```

```
>> help cntc_setmaterialproperties
```

```
function [ ] = cntc_setmaterialproperties(ire, icp, g1, nu1, g2, nu2)
```

```
set the material properties for a contact problem
```

`g1, g2` - modulus of rigidity for body 1, 2 [force/area]
`nu1, nu2` - Poisson’s ratio for body 1, 2 [-]

More information for C programmers is provided in the file `contact_addon.h`. For Fortran programmers the interfaces are specified in `contact_addon.ifc`.

8.3.1 Preparations

`cntc_initlibrary` - load library into matlab (Matlab)
`cntc_initializefirst` - open out-file, initialize data-structures
`cntc_initialize` - further initializations per result element
`cntc_setfileunits` - configure logical file units (Fortran)
`cntc_setglobalflags` - configuration of general settings

cntc_initlibrary The first step in Matlab is to load the library, which is done with `cntc_initlibrary`. This returns a struct CNTC with so-called “magic numbers” that are used in the interface.

```
[CNTC, ifcver, ierror] = cntc_initlibrary;
```

The error code will be -12 if no appropriate license could be found.

This function has no analogue in Fortran/C. In that case the magic numbers are provided in `caddon_flags.inc`.

cntc_initializefirst When using the library from Fortran/C, it is needed to properly initialize its data-structures, license, file I/O etc. This is done by subroutine `cntc_initializefirst`. This routine will be called by `cntc_initialize` if it isn’t called explicitly. In Matlab, this routine is called by `cntc_initlibrary`.

cntc_initialize Additional initializations per result element are done by subroutine `cntc_initialize`, for instance setting the module that’s used. The routine invokes the overall initializations of `cntc_initializefirst` in the very first call, if needed. The call to `cntc_initialize` will be made automatically when needed by other routines of the interface. However, in that case the error code and some configuration options are lost.

cntc_setfileunits CONTACT is programmed in Fortran, such that it uses “logical file units” for file I/O. By default CONTACT uses logical units in the range 11–20. If the encompassing program is programmed in Fortran too this may lead to clashes in the logical units used. In that case the unit numbers usable may be configured using `cntc_setfileunits`. This is not needed when using the library from Matlab or C.

Subroutine `cntc_setfileunits` may be called only once and may only be called before `cntc_initializefirst`. This is because `cntc_initializefirst` may open files after which the logical unit numbers cannot be changed anymore.

cntc_setglobalflags Subroutine `cntc_setglobalflags` allows for the configuration of general settings, in particular the amount of print output of the interface itself and the use of parallel computation, which are the same for all result elements.

The flags that can be configured have been given “magic numbers” that are defined in `caddon_flags.inc`. For instance the interface print-output is configured using flag “2000”, which is the value of parameter `CNTC_if_debug` (Fortran/C) or `CNTC.if_debug` (Matlab):

```
numflg    = 1;           % configure idebug = 0 in the CONTACT library
flags(1)  = CNTC.if_debug;
values(1) = 0;
cntc_setglobalflags(numflg, flags, values);
```

8.3.2 Use of parallel computing

CONTACT supports parallel computing in two different ways. In both cases this uses “shared memory” multi-threading using OpenMP:

1. Multiple contact problems may be computed concurrently;
2. the calculations for one contact problem may use parallel computation as well.

The former strategy does nothing unless the encompassing (Fortran/C) program is adapted and parallelized using OpenMP, whereas the latter strategy doesn’t gain much speedup anymore since the introduction of the solvers NormCG, TangCG and SteadyGS. Therefore parallel computation is disabled by default, by setting the number of threads to use to 1 in `cntc_initializefirst`.

Parallel computation is enabled using the flag `CNTC_if_openmp`. The parameter value provided is the number of threads to use, with `-1` asking to use as many threads as there are cpu cores in the machine that is used. A few restrictions should be kept in mind when using the CONTACT library from a parallel program:

- The initializations, setting of global flags and finalizations may not be called from a parallel region.
- Different threads should never work on the same contact problem (`ire`, `icp`) at the same time.

The maximum number of threads that can be used simultaneously may be restricted by the license that is used. This can be inquired using

```
cntc_getmaxnumthreads      - get max.number of concurrently active threads
```

8.3.3 Configuring basic contact problems (module 3)

The configuration of a basic contact problem (module 3) consists of practically all data that are specified for a case in the input-file (Chapter 5). Data that are set once will be remembered until replaced by newer values. Setting data will sometimes change control digits as well on the basis of the values provided. For instance when configuring the normal force, this will automatically switch to $N = 1$.

```
cntc_setflags              - configuration of flags, control digits (3.3)
cntc_setmetadata           - configuring various metadata in CONTACT (-)
cntc_setsolverflags       - configuration of solver parameters (5.6)
```

```
cntc_setmaterialproperties - set material parameters (5.1.1)
cntc_setinterfaciallayer   - set third body layer parameters (5.1.4)
cntc_setfrictionmethod     - set friction parameters (5.2)
cntc_settimestep           - set time step used in shift problems (5.5)
cntc_setreferencevelocity  - set rolling velocity (5.5)
cntc_setrollingstepsize   - set rolling direction and step size (5.5)
```

```
cntc_sethertzcontact      - set Hertzian problem specification (5.3.1)
cntc_setpotcontact        - set pot.contact for non-hertzian cases (5.3)
```

```
cntc_setpenetration       - set approach/penetration (5.5)
cntc_setnormalforce       - set total normal force (5.5)
cntc_setundeformeddistc   - set undeformed distance function (5.4)
```

```
cntc_setcreepages         - set creepages (5.5)
cntc_setextrarigidslip    - set extra term of rigid slip elementwise (5.5)
cntc_settangentialforces  - set total tangential forces (5.5)
```

cntc_setflags Subroutine `cntc_setflags` allows for the detailed configuration of the contact problem, such as the unit convention used and the setting of control digits. The flags that can be configured have been given “magic numbers” that are obtained from `cntc_initlibrary` (Matlab) or from `caddon_flags.inc` (Fortran/C). For instance CONTACT’s 0-digit is configured using flag 1984, which is the value of parameter `CNTC.ic_output` (Fortran/C) or `CNTC.ic_output` (Matlab):

```
numflg    = 1;                % configure 0 = 3 in the CONTACT library
flags(1)  = CNTC.ic_output;
values(1) = 3;
cntc_setflags(ire, icp, numflg, flags, values);
```

When using just a single contact problem ($ire = icp = 1$), this may be abbreviated as:

```
cntc_setflags(1, 1, 1, CNTC.ic_output, 3);
```

A distinction is made between flags that can be set independently per contact problem versus flags that are the same for all contact problems. The latter ones are configured through `cntc_setglobalflags` (paragraph 8.3.1). Subroutine `cntc_setflags` will call `cntc_setglobalflags` for these flags when necessary.

cntc_setmetadata The meta-data provided here are for output-purposes only, i.e. they don't affect the calculation. The data provided here are stored in the `mat`-file for use during postprocessing.

cntc_setsolverflags This subroutine configures the control digit `G` for the iterative solvers, together with the corresponding input parameters as described in Section 5.6. Additionally, this routine allows configuration of the accuracy with which sensitivities are computed.

cntc_setinterfaciallayer This subroutine configures the slope reduction parameters of the modified Fastsim algorithm ($M = 2$ or 3 , Section 5.1.3) and the interfacial layer of $M = 4$ (Section 5.1.4). This routine can further be used to switch back to $M = 0$.

cntc_setfrictionmethod This subroutine configures the friction parameters corresponding to the L-digit (Section 5.2).

cntc_sethertzcontact As explained in Section 5.3, a distinction is made between Hertzian and non-Hertzian geometries. In the former case the user does not have to specify the element sizes `DX`, `DY` nor the extent of the potential contact area. These will be derived from the curvatures and/or semi-axes provided, using options `IPOTCN = -4` to `-1` as described in Section 5.3.1.

cntc_setpotcontact Subroutine `cntc_setpotcontact` provides the non-Hertzian options `IPOTCN = 1` to `4` (Section 5.3). The undeformed distance is consequently described using `cntc_setundeformeddistc`, which provides the various options of `IBASE` as described in Section 5.4. When `IBASE = 9`, the Fortran/C interface requires a one-dimensional array `h(npot)` with `npot=mx*my`, with index `ix` running fastest. In Matlab this data can also be provided as a two-dimensional array `h(my,mx)`.

cntc_settangentialforces The total tangential forces **FX** and **FY** may be specified instead of the creepages **CKSI** and **CETA**. This requires that the **F**-digit is set to 1 or 2 using **cntc_setflags**. The force values entered here are the total forces divided by the static maximum $F_{STAT} \cdot F_N$.

8.3.4 Configuring wheel/rail contact problems (module 1)

The configuration of wheel/rail contact problems (module 1) re-uses several routines defined above for module 3, setting data for all contact patches at once using $icp = -1$. Additional routines are defined for input data that are specific to the wheel/rail contact module.

```

cntc_setrailprofilefname      - set rail profile filename(s) for w/r problems
cntc_setrailprofilevalues    - set rail profile (table) for w/r problems
cntc_setwheelprofilefname    - set wheel profile filename(s) for w/r problems
cntc_setwheelprofilevalues   - set wheel profile (table) for w/r problems
cntc_settrackdimensions      - set the track geometry for a w/r problem
cntc_setwheelsetdimensions   - set wheelset geometry specification
cntc_setwheelsetposition     - set wheelset position specification
cntc_setwheelsetvelocity     - set wheelset velocity specification
cntc_setwheelsetflexibility  - set wheelset flexibility specification

```

If a rail or wheel profile could not be found or could not be processed correctly, the error code -32 will be issued (**CNTC_err_profil** of **caddon.flags.inc**).

For the configuration of the track geometry consult Section 4.2, for the wheelset geometry refer to Section 4.3.

cntc_setpotcontact In wheel/rail contact, subroutine **cntc_setpotcontact** is re-used to set the grid sizes **DX**, **DS** and the **SAFETY** factor.

8.3.5 Solving the contact problem

```

cntc_calculate                - perform actual calculation

```

After the preparations are complete, the actual calculation is started by calling **cntc_calculate**. This will solve the contact problem (module 3) or contact problems (module 1) and store the results in **CONTACT**'s internal memory. Outputs may be written to the **out-file** and one or more **mat-files** will be created if so configured in the control digits.

8.3.6 Global outputs for wheel/rail contact (module 1)

In wheel/rail contact (module 1), a number of values may be retrieved for the overall configuration.

```

cntc_getwheelsetposition    - get the wheelset position parameters
cntc_getwheelsetvelocity    - get the wheelset velocity parameters
cntc_getglobalforces        - get total forces in track and wheelset coords
cntc_getnumcontactpatches   - get the number of separate contact patches
cntc_getcontactlocation     - get the location of one contact patch

```

cntc_getglobalforces This routine returns the total forces and moments in overall horizontal/vertical coordinate directions. The first six values are defined with respect to the track system, the latter six with respect to wheelset coordinates. The total forces can be retrieved for each contact patch separately ($icp > 0$), acting at the contact reference location. The forces can also be summed over the contact patches ($icp = -1$). The location of the total forces is indefinite in this case, and zero moments will be returned. The forces are defined *on* the output-body, by which the sign depends on the unit convention used (Section 8.2).

cntc_getcontactlocation This routine provides the location of the contact reference point of a contact patch icp in terms of the different coordinate systems. Up to 14 output values can be provided, if permitted by the length of the array provided (1–4: $[xyz]cp_tr, deltcp_tr$, 5–9: $[xyzs]cp_r, deltcp_r$, 10–14: $[xyzs]cp_w, deltcp_w$), see page 44 for further information.

8.3.7 Global outputs per basic contact problem (modules 1 & 3)

Output data may be retrieved for each contact patch separately, using the contact local coordinate convention, using the following routines:

```

cntc_getpotcontact          - get parameters of potential contact area (5.3)
cntc_getcontactforces       - get normal/tangential forces (5.5, 5.7)
cntc_getpenetration         - get approach/penetration (5.5)
cntc_getcreepages          - get creepages (5.5)
cntc_getcontactpatchareas  - get size of contact area (-)
cntc_getmaximumpressure    - get maximum pressure (-)
cntc_getmaximumtraction    - get maximum shear stress (-)
cntc_getsensitivities      - get sensitivities of forces wrt. creepages (-)
cntc_getcalculationtime    - get calculation time used (-)

```

cntc_getpotcontact The potential contact area is set automatically by CONTACT in module 1 or when using a Hertzian option. The parameters may then be retrieved according to

the format of `ipotcn = 3` (Section 5.3), using `mx`, `my`, `xc1`, `yc1`, `dx` and `dy`. The coordinates of the element centers are then easily formed as

```
xcentr = xc1 + [0:mx-1] * dx;    ycentr = yc1 + [0:my-1] * dy;
```

The corners of the elements are

```
xcornr = xc1 + ([0:mx]-0.5) * dx;  ycornr = yc1 + ([0:my]-0.5) * dy;
```

cntc_getcontactforces This routine delivers the total forces and moments on a single contact patch `icp > 0`. The forces and moments are defined along the contact local normal and tangential directions. They are aggregated at the contact reference position of the patch, working *on* the output-body. The sign depends on the unit convention used (Section 8.2). Note that the tangential forces provided are the *true forces* (N) instead of the *relative values* to the static maximum value.

cntc_getcreepages The values of `CKSI`, `CETA`, `CPHI` are returned. In rolling problems these are the creepages (Section 5.5.2), in shifts ($T = 1$) they are shift distances (Section 5.5.1). `CKSI` and `CETA` are computed by the program when using $F = 1$ or 2 . `CPHI` is the value that was input before.

cntc_getcontactpatchareas The area of contact, adhesion and slip areas is computed by counting the number of elements in each state and multiplying by $\delta x \cdot \delta y$.

cntc_getmaximumtraction This subroutine determines the maximum tangential traction $\sqrt{p_x^2 + p_y^2}$ over all elements of the contact area.

cntc_getsensitivities Sensitivities are computed when the S2-digit “SENS” is set to 2 or 3. This uses small perturbations of the “input variables”, i.e. the approach δ and creepages ξ, η, ϕ . The resulting equations are solved with maximally `mxsens` iterations up to a relative tolerance `epsens`. Then the sensitivities of the “outputs” (F_n, F_x, F_y, M_z) are computed using a finite difference approach. This results in a sensitivity matrix `sens` of n_{out} rows with n_{in} columns. In Fortran/C this is presented as a 1D array where the columns are presented consecutively.

8.3.8 Detailed outputs of the contact problem

Detailed outputs are presented by the following routines:

```

cntc_getelementdivision - get elementwise adhesion/slip areas (5.8)
cntc_gettractions       - get elementwise tractions (5.8)
cntc_getdisplacements   - get elementwise displacement differences (5.8)
cntc_getmicroslip      - get elementwise micro-slip velocity (5.8)

```

These subroutines provide access to the detailed results for all elements of the contact area.

In the interface for Fortran/C the values are provided as one-dimensional arrays of `npot=mx*my` elements. For instance the displacement differences `US` are provided in three arrays `un(npot)`, `ux(npot)` and `uy(npot)`.

In the Matlab-interface, these arrays are reshaped to two-dimensional arrays of size `(my,mx)`. The value for the discretization element at column `ix` in grid row `iy` is thus obtained as `un(iy,ix)`. The whole array is easily plotted in Matlab using

```
surf(xcornr, ycornr, un); shading flat;
```

This uses `xcornr` and `ycornr` as obtained with `cntc_getpotcontact` (Section 8.3.7).

cntc_getelementdivision Provides array `IGS`. For each element a code 0, 1 or 2 is provided, indicating that the element is in the exterior, adhesion or slip area respectively.

cntc_gettractions Provides array `PS`, via its columns `PN`, `PX` and `PY`. Tractions are defined *on* the output-body as per the unit convention used (Section 8.2).

cntc_getmicroslip Provides array `S`, via its columns `SX` and `SY`. The micro-slip is defined as the relative velocity ($T = 1$: shift distance) *of* the output-body *relative to* the other body.

cntc_getdisplacements Provides array `US`, via its columns `UN`, `UX` and `UY`. The displacement differences are defined as the elastic displacements *for* the output-body *minus* that of the other body.

8.3.9 Finalization

Finally three subroutines are provided for cleaning up after a calculation. Especially `cntc_closelibrary` cleans up all data-structures, closes open files and unloads the library from memory.

```

cntc_resetcalculationtime - reset timers
cntc_finalize             - cleanup for one result element
cntc_closelibrary        - finalize, cleanup and unload the library

```

8.4 Calculation of creep force curves revisited

An example of the usage of the CONTACT library from Matlab is provided in `matlab_tractcurv.m`. This performs the same calculations as the `tractcurv-example` presented in Section 6.4.

The code largely speaks for itself and is explained further through the comments. Four parts may be distinguished:

1. Initialization of the library;
2. Configuring of the fixed part of the contact problem;
3. Looping over all cases to be computed;
4. Plotting the results.

The cases are computed in the same order as in the input-file `tractcurv.inp`, and are even independent of each other. Therefore it suffices to use just one “contact problem” `icp=1` on one “result element” `iwhe=1`. It would be equally valid to use two “contact problems” for the cases with/without interfacial layer, on the same result element or using different result elements.

```
% creepages: a list of values for creating a creep-force curve

cksi = [ 0.00001 0.0004 0.0008 0.0012      0.2250 0.2500 ]; % [-]
ceta = 0; % [-]
cphi = 0; % [rad/mm]

for iksi = 1:length(cksi)

    % set creepages according to next value from cksi
    cntc_setcreepages(iwhe, icp, cksi(iksi), ceta, cphi);

    % compute the contact problem
    ierror = cntc_calculate(iwhe, icp);

    % get forces on upper body (1) (CONTACT unit convention)
    [fn, fx, fy, mz] = cntc_getcontactforces(iwhe, icp);
    fx_list(iksi) = fx;
end
```

The main body of the script just loops over all the creep combinations to be computed. Storing the results in an array `fx_list`, these can be plotted directly against the inputs `cksi`. The results are the same as shown in Figure 6.4.

The usage of the CONTACT library for wheel/rail contact (module 1) is illustrated in the example `matlab_mbench.m`. This computes the same cases as in the Manchester benchmark example of Section 6.5.

Bibliography

- [1] R.H. Bentall and K.L. Johnson. Slip in the rolling contact of two dissimilar elastic rollers. *Int.J. of Mechanical Sciences*, 9:380–404, 1967.
- [2] J. Boussinesq. *Application des Potentiels à l'Étude de l'Équilibre et du Mouvement des Solides Élastiques*. Gauthier-Villars, Paris, 1885.
- [3] A.F. Bower. *Applied Mechanics of Solids*. CRC Press, Taylor and Francis Group, Boca Raton, 2010. See www.solidmechanics.org.
- [4] R.R. Craig Jr. and A.J. Kurdila. *Fundamentals of structural dynamics, 2nd edition*. John Wiley & Sons, Hoboken, New Jersey, 2006.
- [5] B.E. Croft, E.A.H. Vollebregt, and D.J. Thompson. An investigation of velocity-dependent friction in wheel-rail rolling contact. In M. Uchida, T. Maeda, and K. Goto, editors, *Proceedings of the 10th International Workshop on Railway Noise, Nagahama, Japan*, 2010.
- [6] B. Engel, H.P. Beck, and J. Alders. Verschleißreduzierende Rad-schlupfregelung mit hoher Kraftschlussausnutzung. *Elektrische Bahnen*, 96:201–209, 1998.
- [7] R.C. Hibbeler. *Engineering Mechanics - Statics and Dynamics, 13th Edition*. Pearson Prentice Hall, Upper Saddle River, New Jersey, 2012.
- [8] K. Hou, J. Kalousek, and E. Magel. Rheological model of solid layer in rolling contact. *Wear*, 211:134–140, 1997.
- [9] K.L. Johnson. *Contact Mechanics*. Cambridge University Press, Cambridge (UK), 1985.
- [10] J.J. Kalker. *On the rolling contact of two elastic bodies in the presence of dry friction*. PhD thesis, Delft University of Technology, 1967.
- [11] J.J. Kalker. Simplified theory of rolling contact. *Delft Progress Report Series C1*, 1:1–10, 1973.
- [12] J.J. Kalker. The computation of three-dimensional rolling contact with dry friction. *Int. Journ. for Numerical Methods in Engineering*, 14:1293–1307, 1979.

- [13] J.J. Kalker. A fast algorithm for the simplified theory of rolling contact. *Vehicle System Dynamics*, 11:1–13, 1982.
- [14] J.J. Kalker. Numerical calculation of the elastic field in a half-space. *Comm. of Appl.Num.Meth.*, 2:401–410, 1986. Reprinted as Appendix C in [15].
- [15] J.J. Kalker. *Three-Dimensional Elastic Bodies in Rolling Contact*, volume 2 of *Solid Mechanics and its Applications*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1990.
- [16] J.J. Kalker. Book of tables for the Hertzian creep-force law. In I. Zobory, editor, *Proceedings of the 2nd Mini Conference on Contact mechanics and Wear of Wheel/Rail systems*, pages 11–20, Budapest, Hungary, 1996.
- [17] J.J. Kalker. Rolling contact phenomena - linear elasticity. In B. Jacobson and J.J. Kalker, editors, *Rolling Contact Phenomena*, volume 411 of *CISM Courses and Lectures*, pages 1–85. Springer-Verlag, Wien New York, 2000.
- [18] J.J. Kalker, F.M. Dekking, and E.A.H. Vollebregt. Simulation of rough, elastic contacts. *Journal of Applied Mechanics*, 64(2):361–368, 1997.
- [19] Zili Li. *Wheel-rail rolling contact and its application to wear simulation*. PhD thesis, Delft University of Technology, 2002.
- [20] A.E.H. Love. Stress produced in a semi-infinite solid by pressure on part of the boundary. *Philosophical Transactions of the Royal Society of London*, A228:377–420, 1929.
- [21] M. Malvezzi, E. Meli, S. Falomi, and A. Rindi. Determination of wheel-rail contact points with semi-analytic methods. *Multibody System Dynamics*, 20:327–358, 2008.
- [22] R. Munisamy, D.A. Hills, and D. Nowell. Brief note on the tractive rolling of dissimilar elastic cylinders. *Int. Journ. of Mechanical Sciences*, 33(3):225–228, 1991.
- [23] P.M. Naghdi. *P.M. Naghdi’s Notes on Continuum Mechanics*. University of California, Department of Mechanical Engineering, Berkeley, 2001.
- [24] J. Piotrowski, B.B. Liu, and S. Bruni. The Kalker book of tables for non-Hertzian contact of wheel and rail. *Vehicle System Dynamics*, 55:875–901, 2017. DOI: 10.1080/00423114.2017.1291980.
- [25] O. Polach. Creep forces in simulations of traction vehicles running on adhesion limit. *Wear*, 258:992–1000, 2005.
- [26] V.L. Popov. *Contact Mechanics and Friction. Physical Principles and Applications*. Springer-Verlag, Berlin, 2010.
- [27] D. Roylance. Engineering viscoelasticity. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 2001.

- [28] A.A. Shabana. *Dynamics of Multibody Systems – Fourth Edition*. Cambridge University Press, New York, 2013.
- [29] A.A. Shabana, K.E. Zaazaa, and H. Sugiyama. *Railroad Vehicle Dynamics: A Computational Approach*. CRC Press, Boca Raton, 2008.
- [30] P. Shackleton and S.D. Iwnicki. Wheel-rail contact benchmark, version 3.0. Rail Technology Unit, Manchester Metropolitan University, 2006.
- [31] P. Shackleton and S.D. Iwnicki. Comparison of wheel-rail contact codes for railway vehicle simulation: an introduction to the Manchester Contact Benchmark and initial results. *Vehicle System Dynamics*, 46(1-2):129–149, 2008.
- [32] D.A. Spence. The Hertz problem with finite friction. *Journal of Elasticity*, 5:297–319, 1975.
- [33] M. Spiryagin, O. Polach, and C. Cole. Creep force modelling for rail traction vehicles based on the Fastsim algorithm. *Vehicle System Dynamics*, 51:1765–1783, 2013.
- [34] A.S.K.S. Tjoeng and J.J. Kalker. User’s manual for the program DUVOROL in Algol 60 & Fortran for the computation of three-dimensional rolling contact with dry friction. Technical report, Delft University of Technology, Delft, June 1980.
- [35] E.A.H. Vollebregt. A Gauss-Seidel type solver for special convex programs, with application to frictional contact mechanics. *J. of Optimization Theory and Applications*, 87(1):47–67, 1995.
- [36] E.A.H. Vollebregt. Refinement of Kalker’s rolling contact model. In A. Bracciali, editor, *Proceedings of the 8th International Conference on Contact Mechanics and Wear of Rail/Wheel Systems*, pages 149–156, Firenze, Italy, 2009.
- [37] E.A.H. Vollebregt. User guide for CONTACT, J.J. Kalker’s variational contact model. Technical Report TR09-03, v9.1, VORtech, 2009.
- [38] E.A.H. Vollebregt. Improving the speed and accuracy of the frictional rolling contact model “CONTACT”. In B.H.V. Topping, J.M. Adam, F.J. Pallarés, R. Bru, and M.L. Romero, editors, *Proceedings of the 10th International Conference on Computational Structures Technology*, Stirlingshire, United Kingdom, 2010. Civil-Comp Press.
- [39] E.A.H. Vollebregt. The Bound-Constrained Conjugate Gradients method for non-negative matrices. *J. of Optimization Theory and Applications*, 162(3):931–953, 2014. DOI: 10.1007/s10957-013-0499-x.
- [40] E.A.H. Vollebregt. A new solver for the elastic normal contact problem using conjugate gradients, deflation, and an FFT-based preconditioner. *J. of Computational Physics*, 257, Part A:333–351, 2014.

- [41] E.A.H. Vollebregt. Numerical modeling of measured railway creep versus creep-force curves with CONTACT. *Wear*, 314:87–95, 2014.
- [42] E.A.H. Vollebregt. New insights in non-steady rolling contact. In M. Rosenberger, editor, *Proceedings of the 24th International Symposium on Dynamics of Vehicles on Roads and Tracks*, Graz, Austria, 2015. IAVSD.
- [43] E.A.H. Vollebregt. Updates on the rocking phenomenon. In M. Spiryagin, T. Gordon, C. Cole, and T. McSweeney, editors, *Proceedings of the 25th International Symposium on Dynamics of Vehicles on Roads and Tracks*, pages 605–611, Rockhampton, Queensland, Australia, 2017. IAVSD.
- [44] E.A.H. Vollebregt. Comments on “the Kalker book of tables for non-Hertzian contact of wheel and rail”. *Vehicle System Dynamics*, 2018. DOI: 10.1080/00423114.2017.1421767.
- [45] E.A.H. Vollebregt. Conformal contact: Corrections and new results. *Vehicle System Dynamics*, 2018. DOI: 10.1080/00423114.2018.1424917.
- [46] E.A.H. Vollebregt, S.D. Iwnicki, G. Xie, and P. Shackleton. Assessing the accuracy of different simplified frictional rolling contact algorithms. *Vehicle System Dynamics*, 50(1):1–17, 2012. DOI: 10.1080/00423114.2011.552618.
- [47] E.A.H. Vollebregt, J.J. Kalker, and G.Q. Wang. CONTACT’93 users manual. Technical report, Delft University of Technology, 1993.
- [48] E.A.H. Vollebregt and H.M. Schuttelaars. Quasi-static analysis of 2-dimensional rolling contact with slip-velocity dependent friction. *J. of Sound and Vibration*, 331(9):2141–2155, 2012. DOI: 10.1016/j.jsv.2012.01.011.
- [49] E.A.H. Vollebregt and A. Segal. Solving conformal wheel-rail rolling contact problems. *Vehicle System Dynamics*, 52(suppl. 1):455–468, 2014.
- [50] E.A.H. Vollebregt, C. Weidemann, and A. Kienberger. Use of “CONTACT” in multi-body vehicle dynamics and profile wear simulation: Initial results. In S.D. Iwnicki, editor, *Proceedings of the 22nd International Symposium on Dynamics of Vehicles on Roads and Tracks*, Manchester, 2011. IAVSD.
- [51] E.A.H. Vollebregt and P. Wilders. FASTSIM2: a second order accurate frictional rolling contact algorithm. *Comput.Mech.*, 47(1):105–116, 2010. DOI: 10.1007/s00466-010-0536-7.
- [52] G. Wang and J.J. Kalker. Three-dimensional rolling contact of two viscoelastic bodies. In A. Curnier, editor, *Contact Mechanics. Proceedings; International Symposium, October 7 - 9, 1992, Ecole Polytechnique Fédérale de Lausanne*, pages 477–490, Lausanne, 1992. Presses Polytechniques et Universitaires Romandes.

- [53] G. Wang and K. Knothe. Stress analysis for rolling contact between two viscoelastic cylinders. *Journal of Applied Mechanics; Transactions ASME*, 60:310–317, 1993.
- [54] C.D. van der Wekken and E.A.H. Vollebregt. Numerical calculation of the elastic field in a half-space using bilinear elements. *Mathematics and Mechanics of Solids*, 2018. submitted.
- [55] C.D. van der Wekken, E.A.H. Vollebregt, and C. Vuik. Occurrence and removal of wiggles in transient rolling contact simulation. In J. Ambrósio, W. Schielen, and J. Pombo, editors, *Proceedings of EuroMech colloquium 578, Rolling Contact Mechanics for Multibody System Dynamics*, pages 1–11, Lisbon, Portugal, 2017. IDMEC.
- [56] K. Willner. Fully coupled frictional contact using elastic halfspace theory. *ASME Journal of Tribology*, 130:031405–1/8, 2008.
- [57] J. Zhao, E.A.H. Vollebregt, and C.W. Oosterlee. A fast nonlinear conjugate gradient based method for 3D concentrated frictional contact problems. *J. of Computational Physics*, 288:86–100, 2015.

Appendix A

Specification of in- and output-files

A.1 Files used by CONTACT

The files concerned with the program are:

`<experim>.inp` : input file for surface tractions (A.2) and also for calculation of subsurface stresses (A.3);

`<experim>.out` : output file;

`<experim>.<ncase>.mat` : tractions file, for communication with Matlab (A.4);

`<experim>.<ncase>.subs` : subsurface stresses file, for communication with Matlab (A.5).

A.2 Specification of the input file `<experim>.inp`

In this section we specify the input file `<experim>.inp` that is used with input options `imode = 2` and `3` (Section 3.1.1). If the program finds text on a place where it expects digits, or reals instead of integers, it will stop with an error message. When you are searching for the cause of such an error, notice that

- modifying control digits can imply that other variables will be read or not. `T=1` (shift) has no extra input, `T=3` (rolling) requires `CHI`, `DQ` and `VELOC`.

General rules for the input file and its specification are as follows:

- real values can be written in any format such as `0.35` or `3.5E-1`;
- angles may be given in degrees or radians, for instance using `180d` or `3.1416r`. Radians are assumed if no postfix is given;

- logicals must be denoted with T, t, F or f;
- string values (filenames) are enclosed in single quotes and may not contain quotes in the string: 'My Documents\CONTACT\examples';
- comments are indicated by a percent sign “%”. They can start anywhere in a line and end at the end-of-line;
- there can be any number of blanks between the variables, as long as they fit on the line.
- the maximum length of each line is 256 characters;
- all empty lines and lines containing comments only are skipped;
- for a group of variables, such as “CHI DQ VELOC”, the values must be presented in the same order as in the specification, separated by comma’s and/or blanks;
- groups of values and arrays of values (e.g. “B(k)”) may be entered using multiple lines;
- when a group or array of values is read, all *remaining text* on the final input-line is *ignored*. This may be used to add comments without the %-sign;
- if you change the meaning of the input (e.g. $IPOTCN = -1 \rightarrow IPOTCN = -3$), you should adjust the comments as well (MX,MY,A1,B1,SCALE \rightarrow MX,MY,AA,BB,SCALE), in order not to confuse yourself.
- This also applies for the N-digit (PEN or FN for $N = 0$ or 1) and the F-digit (CKSI, CETA for $F = 0$, FX, CETA for $F = 1$ or FX, FY for $F = 2$): make sure that the comment stays up to date.
- some blocks are optional. They are enclosed in []. They are preceded by a condition in the specification;
- some blocks can be repeated. They are enclosed in [] and are preceded by a stop-condition.

Globally, the file consists of a series of module calls with their input. Module 0 (STOP) has no input. The input of the other modules are subdivided in a number of cases. We confine ourselves to the input for one case of wheel/rail contact using module 1 and one case of basic contact with module 3.

A.2.1 Module 1 – wheel/rail contact

The input for one case of wheel/rail contact consists of:

% Control integers, see section 3.3:

```
CPBTNFS      (CONFIG, PVTIME, BOUND , TANG  , NORM  , FORCE  , STRESS)
LDCMZE       (FRCLAW, DISCNS, INFLCF, MATER , ZTRACK, EWHEEL)
GIAOWR       (GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW  , RETURN)
```

% Parameters for the iterative solution algorithms (section 5.6):

```
if G=0,2,3,4: [ MAXGS  MAXIN  MAXNR  MAXOUT  EPS ]
if G=2,3:     [ OMEGAH  OMEGAS  INISLP  OMGSLP   ]
if G=4:       [ INISLP  OMGSLP                ]
```

% Friction description (section 5.2):

```
%      L=0 --> Coulomb friction with static/kinetic coefficients
%      L=2 --> velocity dependent friction with linear/const formula
%      L=3 --> velocity dependent friction with rational formula
%      L=4 --> velocity dependent friction with exponential formula
```

```
if L=0:      [ FSTAT  FKIN                ]
if L=2:      [ FKIN   FLIN1  SABSH1  FLIN2  SABSH2 ]
if L=3:      [ FKIN   FRAT1  SABSH1  FRAT2  SABSH2 ]
if L=4:      [ FKIN   FEXP1  SABSH1  FEXP2  SABSH2 ]
if L=2,3,4:  [ MEMDST MEM_S0                ]
```

```
% Information needed for influence coefficients,
% particularly the material parameters (section 5.1)
% and the rolling direction and step size (section 5.5):
```

```
if INFLCF>=2:
    if INFLCF=9: [ 'CFNAME'                ]
                POISS 1, 2  GG 1, 2
    if MATER =4: [ GG3  LAYTHK  TAUCRT  GGPLST ]
```

% Information needed for the grid discretization (section 4.5):

```
if D=2:      [ DX  DS  DQREL  SAFETY   ]
```

% Information on the track geometry (section 4.2):

```
if Z=1,3,4: [ GAUGWD  GAUGHT  CANT     ]
if Z=3,4:   [ 'RFNAME1'  MIRRORZ1   ]
if Z=4:     [ 'RFNAME2'  MIRRORZ2   ]
```

```

if Z=2,3,4: [ DYRAIL1  DZRAIL1  DROLLR1  VYRAIL1  VZRAIL1  VROLLR1  ]
if Z=4:     [ DYRAIL2  DZRAIL2  DROLLR2  VYRAIL2  VZRAIL2  VROLLR2  ]

```

```
% Information on the wheelset geometry (section 4.3):
```

```

if E=3,4,7,8: [ FBDIST  FBPOS    NOMRADW    ]
if E=3,4,7,8: [ 'WFNAME1' MIRRORZ1 ]
if E=4,8:     [ 'WFNAME2' MIRRORZ2 ]

```

```
% Information on the wheelset state (section 4.3):
```

```

if E>=1, N=0: [ S_WS    Y_WS    Z_WS    ROLL    YAW    PITCH    ]
if E>=1, N=1: [ S_WS    Y_WS    FZ      ROLL    YAW    PITCH    ]
if E>=2, F=0: [ VS_WS   VY_WS   VZ_WS   VROLL   VYAW   VPITCH   ]

```

```
% Information on flexible wheelset deviations (section 4.3):
```

```

if E>=5:      [ DXWHL1  DYWHL1  DZWHL1  DROLLW1  DYAWW1  DPITCHW1
                VXWHL1  VYWHL1  VZWHL1  VROLLW1  VYAWW1  VPITCHW1 ]
if E=6,8:     [ DXWHL2  DYWHL2  DZWHL2  DROLLW2  DYAWW2  DPITCHW2
                VXWHL2  VYWHL2  VZWHL2  VROLLW2  VYAWW2  VPITCHW2 ]

```

```
% Subsurface stress calculation (section 5.9):
```

```
if S=2,3: [ subsurface points (see below) ]
```

This input is illustrated in the `mbench_a22_left` example in the `examples` directory.

A.2.2 Module 3 – basic Hertzian/non-Hertzian contact

For module 3, the input for one case consists of:

```
% Control integers, see section 3.3:
```

```

PBTNFS      (PVTIME, BOUND , TANG  , NORM  , FORCE  , STRESS)
LDCMZE      (FRCLAW, DISCNS, INFLCF, MATER , RZNORM, EXRHS )
GIAOWR      (GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW  , RETURN)

```

```
% Parameters for the iterative solution algorithms (section 5.6):
```

```

if G=0,2,3,4: [ MAXGS   MAXIN   MAXNR   MAXOUT  EPS   ]
if G=2,3:     [ OMEGAH  OMEGAS  INISLP  OMGSLP  ]

```

```

if G=4:      [ INISLP  OMGSLP                      ]

% Kinematics description (section 5.5):
% Note: PEN  is needed when N=0,      FN when N=1
%       CKSI is needed when F=0,      FX when F=1 or 2
%       CETA is needed when F=0 or 1,  FY when F=2

PEN/FN  CKSI/FX  CETA/FY  CPHI

% Friction description (section 5.2):
%   L=0 --> Coulomb friction with static/kinetic coefficients
%   L=2 --> velocity dependent friction with linear/const formula
%   L=3 --> velocity dependent friction with rational formula
%   L=4 --> velocity dependent friction with exponential formula

if L=0:     [ FSTAT  FKIN                          ]
if L=2:     [ FKIN   FLIN1  SABSH1  FLIN2  SABSH2 ]
if L=3:     [ FKIN   FRAT1  SABSH1  FRAT2  SABSH2 ]
if L=4:     [ FKIN   FEXP1  SABSH1  FEXP2  SABSH2 ]
if L=2,3,4: [ MEMDST MEM_S0                        ]

% Information needed for influence coefficients,
% particularly the material parameters (section 5.1)
% and the rolling direction and step size (section 5.5):

if C>=2:
    if T>1: [ CHI  DQ   VELOC                      ]
    if C=9: [ 'CFNAME'                            ]
    POISS 1, 2  GG 1, 2
    if M=1: [ FG 1, 2  VT 1, 2                      ]
    if M=2: [ FLX  KO_MF  ALFAMF  BETAMF           ]
    if M=3: [      KO_MF  ALFAMF  BETAMF           ]
    if M=4: [ GG3  LAYTHK  TAUCRT  GGPLST         ]

% Information needed for the grid discretization (section 5.3).
% Hertzian options when IPOTCN<0 (sections 5.3.1, 5.3.2), direct
% specification of potential contact area when IPOTCN>0 (section 5.3.3).

if D=2: IPOTCN
    if IPOTCN=-1: [ MX  MY  A1  B1  SCALE ]
    if IPOTCN=-2: [ MX  MY  A1  AOB  SCALE ]
    if IPOTCN=-3: [ MX  MY  AA  BB  SCALE ]

```

```

    if IPOTCN=-4: [ MX  MY  AA  BNEG BPOS SCALE ]
    if IPOTCN= 1: [ MX  MY  XL  YL  DX  DY  ]
    if IPOTCN= 2: [ MX  MY  XL  YL  XH  YH  ]
    if IPOTCN= 3: [ MX  MY  XC1 YC1  DX  DY  ]
    if IPOTCN= 4: [ MX  MY  XC1 YC1  XCM YCM ]

% Information for the undeformed distance calculation, when not using
% a Hertzian option above (section 5.4):

if Z>1 & IPOTCN>0: IBASE  IPLAN
    if IBASE= 1: [ B1  B2  B3  B4  B5  B6  ]
    if IBASE= 2: [ NN  XM  RM  Y1  DY1
                  B(k), k=1..NN          ]
    if IBASE= 3: [ B1  B2  B3  B4  B5  B6  B7  B8  ]
    if IBASE= 9: [ H(I), I=1,NPOT          ]
    if IPLAN= 2: [ PL1 PL2 PL3 PL4 PL5 PL6 ]
    if IPLAN= 3: [ XL1 XH1 YL1 YH1 XL2 XH2 YL2 YH2 ]

% Kinematics, extra terms to tangential right hand side (section 5.5):

if E=9: [ EXRHS(I,2), EXRHS(I,3), I=1,NPOT  ]

% Subsurface stress calculation (section 5.9):

if S=2,3: [ subsurface points (see below) ]

```

This input is illustrated further in the file `template.inp` in the `examples` directory.

A.3 Subsurface-stress input in the file `<experim>.inp`

The input of the subsurface points in `<experim>.inp` starts with the control digits A and 0, governing the writing of the Matlab-file `<experim>.<case>.subs` and the level of output to the out-file (Section 5.9.1). This line appears once irrespective of how many blocks of points are used.

```

    if S=2,3: [ MATFIL, OUTPUT ]

```

Next follow one or more blocks of $NX \cdot NY \cdot NZ$ points. The presence of more blocks is signalled by the integer digit `ISUBS`. In a way this is comparable to the `MODULE` number: calculations continue until `ISUBS = 0`. The input of one block consists of the following lines (see section 5.9.2):

```

if S=3: [
  ISUBS
  if ISUBS=1 or 5: no input for IX,IY - all elements selected
  if ISUBS=2 or 6: [ IXL IXINC IXH
                    IYL IYINC IYH      ]
  if ISUBS=3 or 7: [ NX NY
                    IX(i), i=1,..,NX
                    IY(j), j=1,..,NY  ]
  if ISUBS=1,2,3: [ NZ ZL DZ      ]
  if ISUBS=5,6,7: [ NZ
                    Z(iz), iz=1,..,NZ ]
  if ISUBS=9:     [ NX NY NZ
                    X(i), i=1,..,NX
                    Y(j), j=1,..,NY
                    Z(k), k=1,..,NZ  ]
  if ISUBS=0:     last block complete, no more input
]

```

A.4 Specification of the file <experim>.<case>.mat

This file is written for each case with A-digit ≥ 1 and contains the tractions of the case. The case number “<case>” is written in 4 or 6 digits or omitted for case-numbers above one million.

The file is used in the plot-programs loadcase.m, plot2d.m and plot3d.m (Section 7.2).

The tractions can be stored for elements inside the contact area only ($A = 1$) or all elements of the potential contact area ($A = 2$). The first few lines contain several extra variables that are needed for the pictures. Matlab knows only matrices so that the file must consist of lines that all have an equal number of columns (11):

```

line 1: % comment line, describing grid discretisation variables
line 2:  Mx   My   Xl   Yl   Dx   Dy   Chi  Dq  (dum)  (dum)  Fmt
line 3: % comment line, describing the material parameters used
line 4:  Tdigit Mdigit Gg1   Gg2   Poiss1 Poiss2
line 5: % comment line, describing the friction law used
line 6:  Frclaw  (dum)  Veloc  (if L=0:) Fstat  Fkin
line 7: % comment line, describing columns of the table
for all elements in the (potential) contact area:
  i  Igs(i)  H(i)  Mu(i)  Pn(i)  Px(i)  Py(i)  Un(i)  Ux(i)  Uy(i)  SRel(i)
i    = element number. For element (ix,iy), i = ix + mx*(iy-1)
Igs(i) = state, 0=Exter, 1=Adhes, 2=Slip
H(i)   = undeformed distance in element i

```


$\text{Mu}(i)$ = actual friction coefficient in element i
 $\text{Pn}, \text{Px}, \text{Py}$ = normal/tangential tractions in element i
 $\text{Un}, \text{Ux}, \text{Uy}$ = normal/tangential displacement differences in element i
 $\text{SRel}(i)$ = magnitude of the relative slip velocity in element i

Note: in the computation of shifts ($T = 1$) instead of rolling ($T \geq 2$), the shift distance Shft is stored instead of Srel .

A.5 Specification of the file <experim>.<case>.subs

This file is used by the Matlab script `plotstrs.m`. It contains 8 or 14 data columns depending on whether the full stress tensor is exported or not, see the A-digit on page 34. For each block of subsurface points the file contains the following lines:

```

line 1: % comment line, describing variables on second line
line 2:  Nx  Ny  Nz
line 3: % comment line, describing columns of the table
for all Nx.Ny.Nz subsurface points of the block:
    line 3+k:  X   Y   Z   UX   UY   UZ   SIGHYD  SIGVM [ ...
                                   SIGXX  SIGXY  SIGXZ  SIGYY  SIGYZ  SIGZZ ]
X,Y,Z      = coordinates of the point where stresses are calculated
Un,Ux,Uy   = normal/tangential displacements in subsurface point k
SigHyd     = mean hydrostatic stress  $\sigma_{\text{hyd}} = I_1/3$ 
SigVM      = von Mises stress  $\sigma_{\text{vm}} = \sqrt{3*J_2}$ 
Sigxx,...,Sigzz = components of the stress tensor

```

Refer to Sections 5.8 and 5.9 for the description of these output quantities.