Vtech CMCC

Contact | Mechanics | Computing | Consulting

P.O. Box 27097
3003 LB Rotterdam
The Netherlands

info@vtech-cmcc.nl

# User guide for CONTACT, Rolling and sliding contact with friction

*Technical Report* 20-01, version 'v23.2'

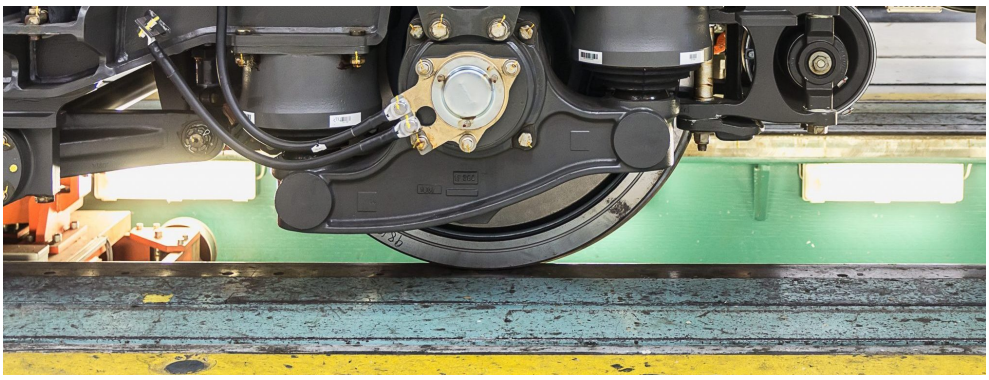
*Date*

November 10, 2023

*Author(s)*

Dr.ir. E.A.H. Vollebregt





# Contents

# 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 contacts. For instance to investigate the behavior of the wheel/rail creep forces and the subsequent wear and damage of railway wheels and rails [29, 65].

This program implements the famous theories for rolling contact by Prof. J.J. Kalker of Delft University of Technology. These were presented first in [14] and are described in full detail in [17]. Extended introductions are given in [19] and [60]. Extensions by Vollebregt concern the automated analysis of wheel/rail contacts [57, 56, 59, 62], the effects of third body layers [51, 66] and falling friction [63], together referred to as 'Extended CONTACT', and speedup by dedicated iterative solvers, e.g. [50, 74].

## 1.1 Prerequisites

To understand this report and work with our program, a basic understanding of the deformation of solid objects is needed. 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 Wikipedia[1] and in text books such as [4] and [31].

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 [9]. The dynamics of deformable objects are surveyed among others in [5]. Reference texts on multi-body dynamics are provided by [38, 39]. The former of these is more general and pays attention to flexible bodies, the latter is targeted more on railway applications.

[1] `en.wikipedia.org/wiki/solid_mechanics`

## 1.2 Purpose of CONTACT

Contact mechanics concerns the interaction of deformable bodies. A key aspect is that the contact area depends on the deformation of the bodies, that depends on the contact area and contact stresses. The determination of the contact area thus becomes 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 parts of the boundary that are fixed and known beforehand.

Different types of contact problems may be identified. For instance frictionless and frictional problems, stationary and instationary, and concentrated or diversified. Concentrated contacts are those where the contact area is small compared to the typical dimensions of the bodies, and where the contacting bodies are 'smooth edged' near the contact zone.

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 [16, 72]. 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 plotting routines provided,
- 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, Python, Fortran and C.

The latter two options are relevant for wear calculations and for Vehicle System Dynamics (VSD) simulation codes. Add-ons to GENSYS, NUCARS, SIMPACK Rail [68] and Universal Mechanism[2] have been realized and are provided on a commercial basis.

[2] `www.gensys.se`, `www.aar.com/nucars`, `www.simpack.com`, `www.universalmechanism.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 [14, 44]. 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 [17]. 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 [45] and the extension to viscoelastic contact problems (steady state rolling) [70, 71]. At that time, the program was used for creating a table for the Hertzian creep-force law [18] and for the study of rough elastic contacts [20]. 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 (Kalker's) CONTACT'93 [61].

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 [46]. 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 (VORtech's) CONTACT v9.1 [47].

VORtech extended the range of applicability of CONTACT since 2009 by incorporation of velocity dependent friction laws [51, 63], effects of roughness and contamination ('third body layer') [51, 65, 66], by making extensions for solving conformal contact problems [55, 56, 64], and via new enhanced solvers on the basis of Conjugate Gradients and FFTs [50, 74].

The software was transferred to the company Vtech CMCC in 2020, that now continues this development and distribution. Research is being done to extend CONTACT for rails and wheels with variable profile cross-section, i.e. switches and crossings [62] and wheel out-of-roundness. When such new functionality is achieved, this will be included in future versions of this document.

## 1.4 Structure of this report

The remainder of this report is structured as follows.

- The operation of the program is described in Chapter 2. 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 program's execution.
- Chapter 3 describes the input and output quantities for wheel/rail contact analysis.
- Chapter 4 describes the additional input and output quantities for basic contacts, such as material and friction data.

- The usage of the stand-alone program is demonstrated further through the examples in Chapter 5. Chapter 6 describes the Matlab scripts for visualization of the results.
- The use of the CONTACT library is documented in Chapter 7.
- Appendix A specifies the files that are used.
- Finally, Appendix B presents a quick overview of the computational model used. For further details and background information the reader is referred to [17, 19, 56, 65] and [60].

## 1.5 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 [11] and Kalker [17] 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 [36], whereas [65] presents a review on the creep forces.
- 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 2, 3 and 4.
2. Run the program for the examples that are provided and go through their description in Chapter 5 of this guide.
3. Put guide cards at Sections 2.3 and A.2 of this guide; this reference information is consulted frequently.

# Chapter 2

# Operation of the program

The basic structure of the CONTACT stand-alone program is illustrated schematically in Figure 2.1.

## 2.1 Starting the program

### 2.1.1 Terminal-based, textual interface

The program CONTACT is a terminal-based program, with textual input from an 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 2.1.2).

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

```
Mode of operation of this program.

IMODE=1: license management,
IMODE=2: start from input file <EXPERIM>.inp,
IMODE=3: check input file <EXPERIM>.inp,
```

### 2.1.2 Windows-based interface, the CONTACT GUI

The CONTACT GUI is a small auxiliary program by which CONTACT's command line may be avoided. The main purpose of the GUI is to assist in selecting an input file and starting the calculation.

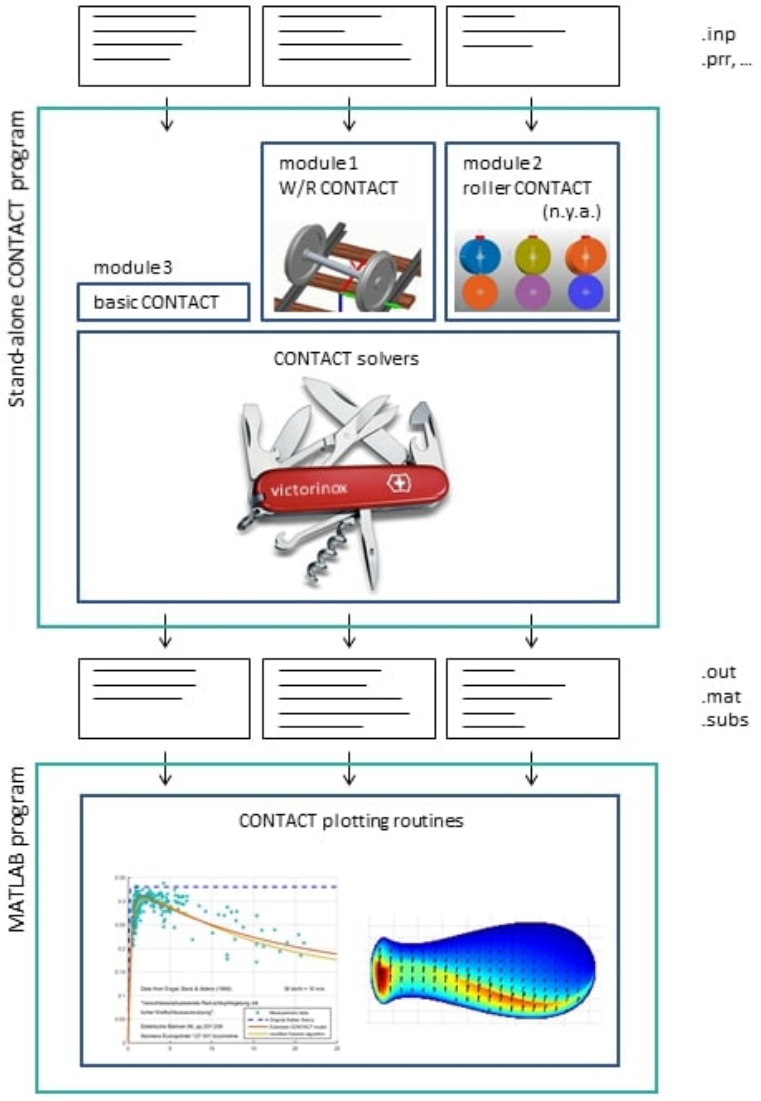



Figure 2.1: *Structure of a calculation using the Stand-alone CONTACT program.*

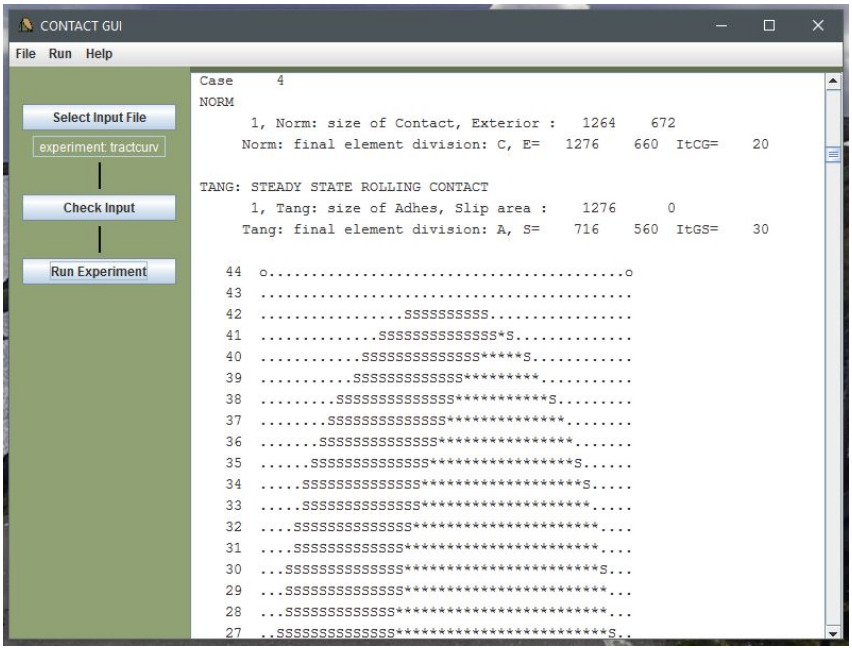



Figure 2.2: *Main window of the CONTACT GUI.*

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 2.2. License management actions are provided in the Help menu.

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 2.2. Other output files are created as well, that may be inspected for instance in MATLAB as indicated in Figure 2.1.

### 2.1.3 License management

CONTACT licenses are distributed via e-mail, providing the license ID and activation password. These are used to **activate** the software on the user PC. This generates a request that's transmitted via the Internet to a web-based licensing server, checked, and used to generate a license file for the user PC.

A license file is stored locally and used to check credentials each time the program is run. The contents of the file can be viewed using the **print** option.

The license file is updated from time to time to let new details be added, such as an extended duration. This is done automatically when the program finds that a refresh is needed. It can be triggered manually using the **refresh** option.

License files for the stand-alone program are named `contact_<licenseid>.lic`. The library version uses `clibrary_<licenseid>.lic`. They are stored in the user profile folder (Windows: `%USERPROFILE%`, Linux: `$HOME`).

Further information is provided in a separate document '**licensing.pdf**' in the installation folder [58].

## 2.2 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 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 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.

## 2.3 The control integers

The specification of each case is largely governed by so-called 'control digits'. 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.

### 2.3.1 Modules

The program CONTACT is built up using different 'modules' (sub-programs) that use a shared computational core, see Figure 2.1. 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. This revolves around a 'half wheelset' on a 'half track' (rail or roller). The module starts from a wheelset at a given track location, uisng 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 is a basic driver routine for Hertzian and non-Hertzian cases. This skips over the contact search and creep calculation, which must be performed by the user.

When there are no more cases to compute this is signalled to the program by the artificial module number 0.

### 2.3.2 Module 1 – wheel-rail contact

**CPBTNFS, VLDCMZE, HGIAOWR** – the control words, compressed form

The following list is ordered the same as the letters in the control words. The first 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.

**$C_1$ - CONFIG** : concerns the *configuration* or *composition* of the problem:

- **0** – half wheelset on half track, left wheel;
- **1** – half wheelset on half track, right wheel.
- **4** – half wheelset on roller, left wheel;
- **5** – half wheelset on roller, right wheel.

A configuration with a single wheel can be described using a left or a right wheel.

**P - PVTIME** : the relation of the current to the *previous case* or *previous time instance*:

- **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** – no continuation; in transient cases (T = 1, 2), the previous tractions vanish entirely (initiation of contact), in other cases (T = 0, 3), they will not be used.

$^*$ The P-digit is relevant only in transient shifting or rolling, T = 1 or 2, which are not yet provided in module 1. Frictionless cases and steady rolling use P = 2, which is the only option available in module 1.

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

- **0** – full linearly elastic model and contact conditions;
- **2**$^{\dagger *}$ – elliptical traction bound derived from Hertzian problem data (requires IPOTCN < 0);
- **3**$^{\dagger *}$ – parabolical traction bound derived from Hertzian problem data (requires IPOTCN < 0);
- **4**$^{\dagger *}$ – simple double half-elliptical contact area and pressure distribution (SDEC [34], requires IPOTCN = −6, N = 1);
- **5**$^{\dagger}$ – non-Hertzian approximation of the normal contact problem using the Kik-Piotrowski approach with Ellipse Correction (KPEC, §B.8).

**6**† – non-Hertzian approximation of the normal contact problem using a modified ANALYN method.

† Options 2 to 6 are approximate by nature. They are primarily meant to be used together with FASTSIM (`M` = 2, 3).

* Options `B` = 2 to 4 are not yet available in module 1.

**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.

* Module 1 does not yet provide for transient computations.

**$N_1$ - 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 new control-digits for the subsurface calculation, compute stresses in the points already stored in memory;

**3** – read new subsurface points, and compute subsurface stresses for these points.

The `F`-digit is not yet available in module 1. This is described in the input for module 3 below.

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

**V - VARFRC** : concerns the *variation* of *friction* across the rail profile:

**0** – a single set of inputs is used across the entire width of the rail;

**1** – multiple sets of inputs are used, varying the coefficient of friction across the rail profile.

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

**0** – Coulomb friction with static and kinetic friction coefficients;

**1** – maintain the same friction method as in the previous case;

**4** – friction law with exponential dependency on slip velocity.

**6** – friction law with piecewise linear dependency on surface temperature;

**$D_1$ - DISCNS** : concerns the *type of grid* and its *discretisation*:

**0, 1** – maintain the same approach as used in the previous case;

**2** – planar contact approach, with rigid slip/creepage on contact reference plane;

**4** – conformal contact approach with curved $(x, \tilde{s}, \tilde{n})$-coordinates and corresponding rigid slip calculation;

**5** – same as 2, using brute force, grid-based algorithm instead of contact locus.

New inputs are read when $D_1 = 2, 4$ or 5.

**$C_3$ - INFLCF** : concerns the material parameters and corresponding *influence coefficients*:

**0, 1** – maintain the same approach as used in the previous case;

**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;

**4** – using influence coefficients for conformal contact, based on the half-space with angle correction (Blanco-approach [2, 3], experimental).

**9*** – using numerical influence coefficients; read parameters and filename, read pre-computed influence coefficients from file.

* Option $C_3 = 9$ is not available in module 1.

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

**0** – purely elastic contact (§4.1.1);

**2** – simplified theory with a single, user-defined flexibility value (modified Fastsim, §4.1.4);

**3** – simplified theory with three automatically computed flexibility values (modified Fastsim, §4.1.4);

**4** – elastic contact with interfacial layer and near-surface plastic deformation (§4.1.5).

**$Z_1$ - ZTRACK** : concerns the *track* design *geometry*, rail profile and optional rail deviations:

**0** – maintain track dimensions, rail profile and deviations;

**1** – read new track dimensions;

**2** – read new rail deviations for the current side of the track;

**3** – read new track dimensions, and read the profile and rail deviations for the current side of the track (dependent on $C_1$-digit).

**$E_1$ - EWHEEL** : concerns the *wheelset geometry*, position and velocity data, and the wheel profile:

**0** – maintain wheelset geometry, position and velocity data, and the wheel profile;

**1** – read new position data;

**2** – read new position and velocity data;

**3** – read new position and velocity data, new wheelset geometry, and read the wheel profile for the current side of the wheelset (dependent on $C_1$-digit);

**4** – read new position and velocity data, including flexible wheelset deviations;

**5** – read new position and velocity data, new wheelset geometry, wheel profile, and read flexible wheelset deviations.

The third control word '**XHGIAOWR**' concerns the output of the program and the flow of the calculations.

**X - XFLOW** : govers the extent of *extra (debug)* grid data printed to the output-file:

**0** – no additional debug output needed for this case;

**1** – read new input for configuration of debug outputs.

**H - HEAT** : determines the calculation of surface temperatures:

**0** – no surface temperature calculation required for this case;

**1** – compute surface temperatures for parameters stored in memory;

**3** – read new input data, and activate the surface temperature calculation for steady rolling.

**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 $\omega$'s, read max. iterations from input file;

**1** – maintain the solver settings as used in the previous case;

**2** – use ConvexGS at all times, read $\omega$'s from input file;

**3** – use SteadyGS when possible ($T = 3$), read $\omega$'s from input file.

**4** – use the default solver, read parameters for slip velocity iteration (page 63) from input file.

**5** – use GDsteady when possible ($T = 3$), read parameters from input file (experimental).

**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;

**3**[*] – previous case gives a good initial estimate. Use its tractions and element division (not touched).

[*] The only option that is provided in module 1 is `I` = 0.

**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 4.9.1.

**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; in conformal contacts ($D_1 = 4$), the curved reference surface is printed;

**4** – the wheel and rail profiles are printed as used in the calculations, after smoothing, in track coordinates;

**5** – the detailed solution inside the contact area (tractions, slip, and other quantities of interest) is printed as well;

**6** – the detailed solution is printed for all elements of the potential contact.

You may typically set `O` = 2 or 3 and `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 4.9.1.

**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.

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

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

A fourth control word '**PSFLRIN**' is used when X = 1, configuring the debug outputs from different parts of the code. This consists primarily of the level of output. Additional configuration may be added later on subsequent lines.

**XP - X_PROFIL** : determines print-output from profile processing

**XS - X_SMOOTH** : determines print-output from spline processing

**XF - X_FORCE** : determines print-output from total force iteration

**XL - X_LOCATE** : determines print-output from contact location

**XR - X_READLN** : determines print-output on readline

**0** – errors only;

**1** – warnings, information messages;

$\geq$ **4** copy inputs.

**XI - X_INFLCF** : determines print-output on influence coefficients

**XN - X_NMBDG** : determines print-output from contact solvers

**0-3** – none;

**4** – global problem inputs hs and outputs igs, ps, us (?);

**5** – intermediate results of the Panagiotopoulos' process (?);

**6** – intermediate results of Veloc.dep iterations (?);

**7** – intermediate results of NORM and TANG algorithms (?); print influence coefficients as well;

**8** – intermediate results of Newton-Raphson loops (?);

**9** – information per iteration of NormCG, TangCG, ConvexGS, SteadyGS (?).

### 2.3.3 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 `F`-digit, whereas the functionality of the `N`-digit is changed:

**$N_3$ - NORM** : specifies whether the *normal force* or *approach* is prescribed:

- **0** – approach `PEN` (penetration) prescribed;
- **1** – total normal force `FN` prescribed.

**$F_3$ - 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 `D`-, `Z`- and `E`-digits are used to describe the grid, geometry and rigid slip, like in module 1, however, the meaning of these digits is different from the earlier interpretation.

**$D_3$ - 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 input parameters and form the discretisation correspondingly.

**$Z_3$ - 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.

**$E_3$ - 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 3

# 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 4.

## 3.1 Overview of wheel/rail contact configurations

**Track coordinates** Wheel-rail contact analysis starts from a track as illustrated in Figure 3.1. This shows the track plane resting on the two rails, that may be inclined (canted) with respect to the overall horizontal direction.

Figure 3.2 shows a world-fixed global reference system '$isys$' with an arbitrary track section. The track center line is indicated by $\mathbf{\Gamma}(s)$ and shown by the red curve. Curvilinear coordinates are developed around this track curve. On the curve we have $x_{fc} = s$, $y_{fc} = z_{fc} = 0$, and the origin $O_{fc}$ is placed at $s = 0$. These coordinates are fixed with respect to $isys$ and are called the 'fixed curvilinear' coordinate system '$fc$'.

A moving rectilinear 'track coordinate system' '$tr$' is introduced for the contact computation. The

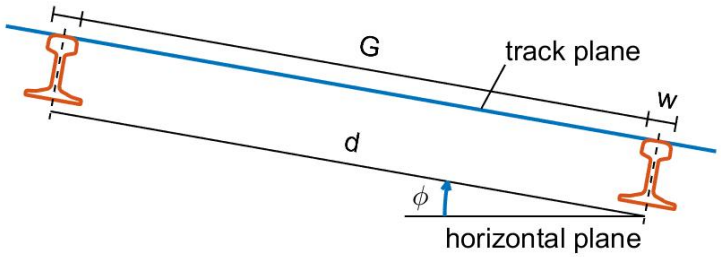



Figure 3.1: *Track viewed in world-fixed coordinates, illustrating rail distance d, rail width w, gauge width G, and track inclination (track cant, elevation) angle $\phi$ (adapted from [39]).*

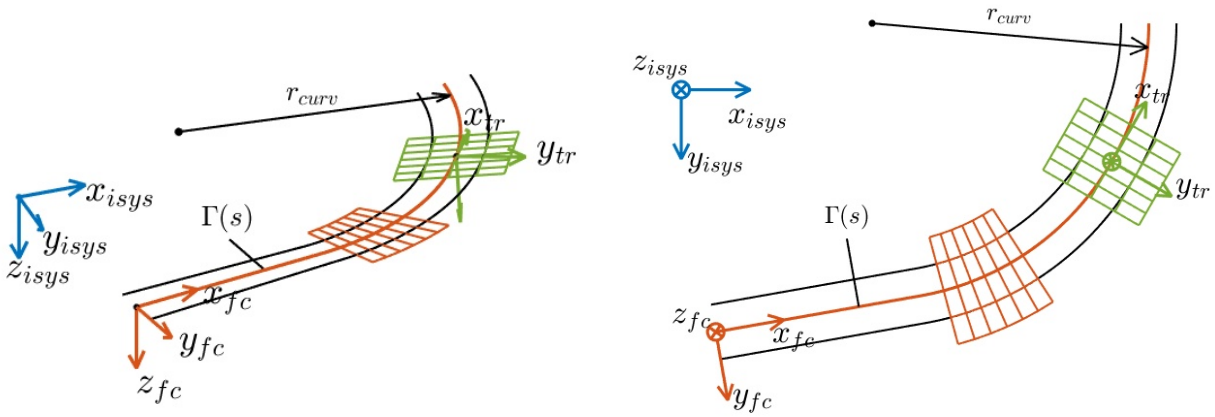



Figure 3.2: *Left: perspective view, right: top view on the track, illustrating a track curve* $\mathbf{\Gamma}(s)$ *with 'fixed curvilinear' track coordinates 'fc' (red) versus 'moving rectilinear' track coordinates 'tr' (green).*

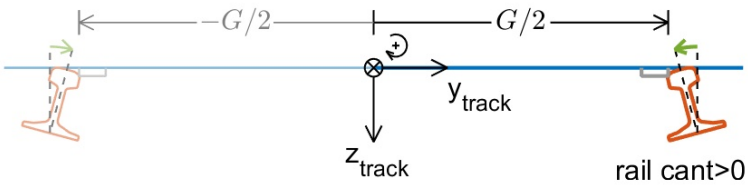



Figure 3.3: *CONTACT considers half of the track, either the left or the right side. When using the gauge point computation, the track coordinate system 'tr' is placed at the center of the plane resting on the (inclined) rails in initial (design) configuration.*

track origin is placed on the track curve at the point closest to the wheelset center, see e.g. [27]. This position is denoted here as $s_{ws}$, with $s$ ($s_1$, longitudinal) the parameter used to describe the track curve.

The moving track coordinate system $tr$ is defined further as shown in Figure 3.3. The $x_{tr}$ direction (also: $x_{track}$) is aligned with the track center line. Right-handed coordinates are used with positive $z_{tr}$ pointing downwards.

**Half wheelset on half track model** Each 'case' in CONTACT considers a half wheelset on a half track. The side that's considered is governed by the CONFIG control digit ($C_1$, page 16).

Inside the program, calculations for the left side are solved using an equivalent right side configuration. This shows up in print output for the flow of computations that is given for this mirrored situation. The mirroring is undone in the final print output and `mat`-file showing the results of the computation.

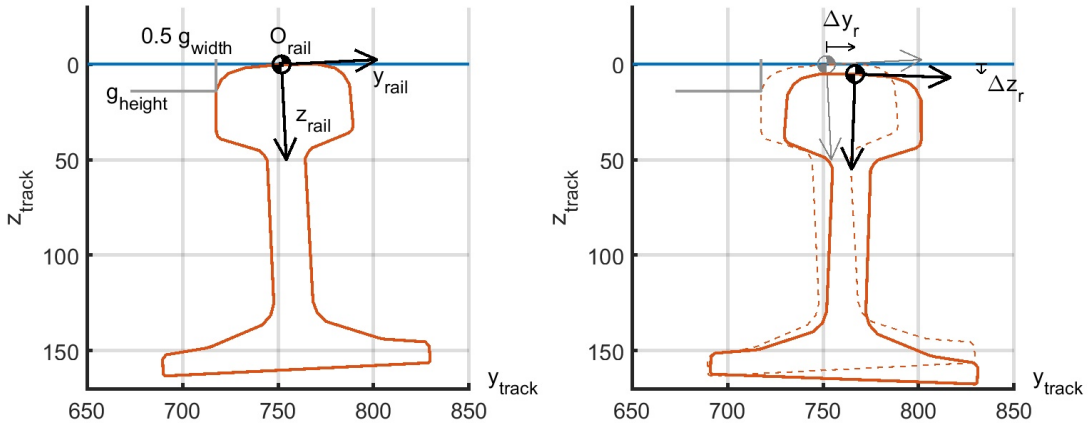



Figure 3.4: *Left: initial (design) placement of the right rail profile in the track system using the gauge point. Right: actual (current) configuration with rotation* $\Delta\phi_r > 0$ *and displacements* $\Delta z_r, \Delta y_r > 0$.

**Gauge point computation** In case the rails are placed symmetrically with respect to the track origin, the position of the rail profile (datum) in the track system can be computed automatically. This uses the gauge width between the inner faces of the two rails (i.e. the grey stops in Figure 3.3). Figure 3.4 (left) shows the right rail as it is seen in the track system. This 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 $O_{rail}$ (datum, denoted also $O_r$). It is first rotated for cant and then shifted up to just touch the track plane and shifted left to touch the gauge stop. Note that $O_{rail}$ may be associated with the highest point in the uncanted rail, which need not be the highest point on the rail after cant is applied.

**Absolute rail placement** An alternative to the gauge point computation is to use absolute rail placement. The rail profile is specified with respect to an origin $O_r$. It is rotated for cant. After this, the origin $O_r$ is shifted to the position $[0, y_{r(tr)}, z_{r(tr)}]^T$ specified by the user. $y_{r(tr)}$ will be $< 0$ on the left side and $> 0$ for the right rail.

**Track deviations** Track irregularities may be defined that displace the rail with respect to its initial (design) position and orientation, see Figure 3.4, right. The rail is rotated about its origin $O_{rail}$ by roll angle $\Delta\phi_r$, and is shifted by $\Delta y_r, \Delta z_r$ with respect to the track origin, which isn't affected by the irregularities. Note that the rail typically will not be 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 $v_{y,r} = \Delta\dot{y}_r, \ldots$ may be specified also.

**Roller rig configurations** For the simulation of roller rigs, it is assumed that the roller axle is fixed in a frame, unable to move except for rotation about its axle. Track coordinates are used largely similarly as above for wheelset on track configurations, with slight differences as indicated in Figure 3.5:

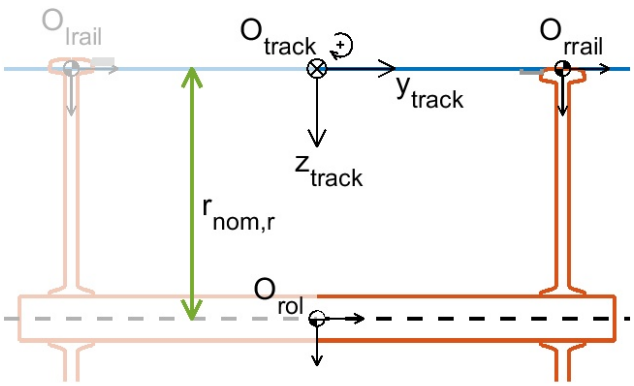



Figure 3.5: *Definition of track coordinates for the simulation of a roller rig: aligned with the rollers' axle, at a distance $r_{nom,r}$ above the rollers' center position.*

- The track plane is aligned with the rollers' axle, at the nominal roller radius $r_{nom,r}$ above the rollers' center position;
- The reference point $(0, 0)$ of the rail profile is placed in the track plane, without shifting to make the profile touching the track plane. Non-zero $z$-values in the profile are thus interpreted as offsets to the nominal rolling radius $r_{nom,r}$.

The cant angle is difficult to interpret in this configuration, and will be ignored.

Figure 3.5 concerns the initial (design) configuration. In the actual (current) configuration, the rail profiles may be rotated and displaced by rail deviations. This feature may also be used to describe the motion of the roller rig as a whole, relative to the frame in which it is contained.

**Switches and crossings** For the simulation of switches and crossings, so called 'variable profiles' may be used, that consist of multiple 'slices' at different $s_{fc}$ positions along the track curve. CONTACT will read the slices and interpolate to the actual position of the wheelset along the track curve.

**Wheelset geometry** Wheel profiles are specified relative to profile reference points $O_{wheel}$ $(O_w)$. In initial (design) configuration, the profile reference is placed at $[0, \pm(d_{flng}/2 - y_{fbpos}), r_{nom,w}]^T$ with respect to the wheelset center (Figure 3.6, left). Here $d_{flng}$ is the flange-back distance, and $y_{fbpos}$ is the position of the flange back with respect to the profile reference.

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

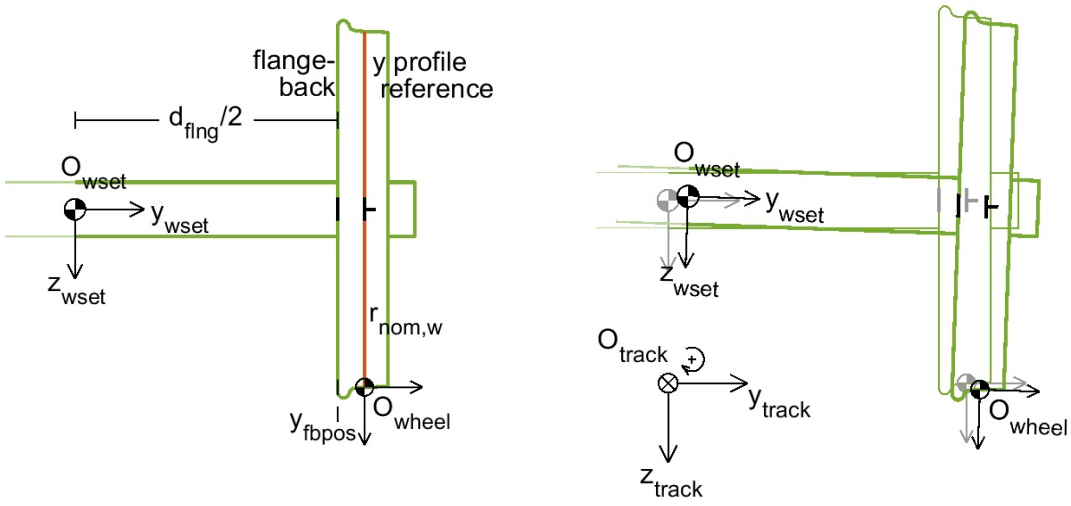



Figure 3.6: *Left: illustration of the parameters used for the geometry of a half wheelset. Right: wheelset center (reference) position, and orientation with respect to the track system.*

**Wheelset degrees of freedom** The position of a wheelset is characterized by the position of its center along the track curve ($s_{ws}$, $fc$-coordinates), which is irrelevant to CONTACT except for switches and crossings, and the position $y_{ws}, z_{ws}$ and orientation $[\phi_{ws}, \psi_{ws}, \theta_{ws}]$ with respect to the $tr$ track reference (Figure 3.6, right). The orientation is defined with Euler angles in roll–yaw–pitch convention: starting with the axle parallel to the $y_{tr}$-direction, the wheelset is rolled about its $x$-axis by $\phi_{ws}$, then yawed by $\psi_{ws}$ about the new vertical axis $z_{wset}$ and then pitched by $\theta_{ws}$ about the axle, i.e. the new $y_{wset}$-axis. After this the wheelset is shifted to its position $[0, y_{ws}, z_{ws} - r_{nom,w}]^T$.

**Wheel out-of-roundness** Out-of-round wheels may be defined using cylindrical coordinates as illustrated in Figure 3.7, left, with radius $r_{wc} = r_{nom,w} + dr_{wc}$, with profile heights $dr$ given as a function $dr_{wc}(\theta_{wc}, y_{wc})$. This is provided to CONTACT using multiple slices at different $\theta_{wc}$ positions around the circumference. CONTACT will read the slices and interpolate as needed to form a 3D surface for the contact calculation.

**Single wheel-rail configuration** Using the mechanisms outlined above, single wheel/rail configurations can be computed as well.

- The gauge width computation of Figures 3.3 and 3.4 is disabled using a negative value for the gauge measuring height, e.g. $g_{height} = -1$;
- Next, the track center is associated with the rail profile reference using absolute rail placement with $y_{r(tr)} = z_{r(tr)} = 0$;

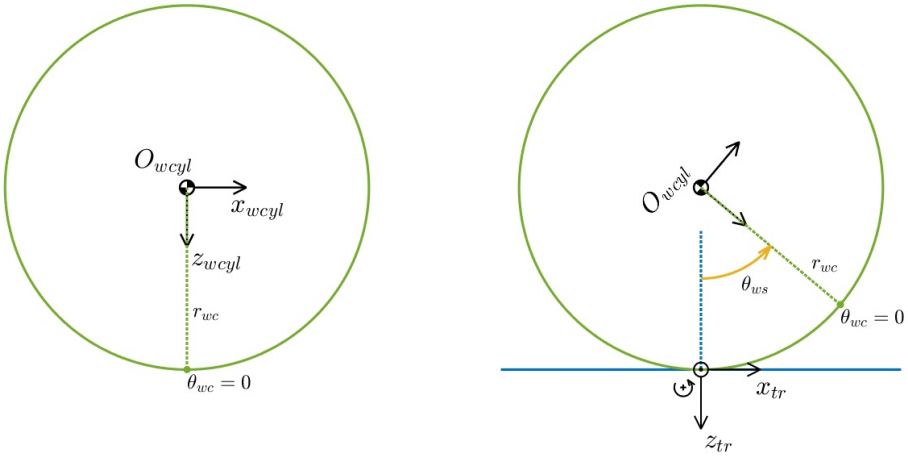



Figure 3.7: *Left: definition of cylindrical coordinates* $(\theta_{wc}, y_{wc}, r_{wc})$ *for wheel out-of-roundness. Right: rotation by the wheelset pitch angle* $\theta_{ws}$.

- Finally, the wheel profile marker is placed directly under the wheelset center (Figure 3.6), using $d_{flng} = 2y_{fbpos}$, such that $y_{wheel} = d_{flng}/2 - y_{fbpos} = 0$.

**Kinematic configuration** CONTACT focuses on contact problems instead of the balance (equilibrium) or motion (dynamics) of the whole wheelset. Each case of the simulation therefore concentrates on a single wheel/rail combination. When using CONTACT from a vehicle dynamics simulation, the full position and velocity states are typically prescribed beforehand. Alternatively, some total forces may be specified also, which is particularly useful in stand-alone calculations.

## 3.2 Inputs regarding wheel and rail profiles

Wheel and rail profiles may be obtained from various sources, in different formats and using different conventions. Different options are provided to read existing files and transform into the form as needed by CONTACT.

**Supported formats** Constant rail profiles may be provided in SIMPACK `prr`-format or Miniprof `ban`-format, with the format taken from the file extension, or as a 2-column table with $(y_i, z_i)$-values if another file extension is used. Variable rail profiles may be provided in a slices-file with `slcs` file extension. Constant wheel profiles may be provided in SIMPACK `prw`-format or Miniprof `whl`-format, with the actual format taken from the file extension, or as a 2-column table if another file extension is used. Out-of-round wheels may be provided in a slices-file with `slcw` file extension.

| | | |
|---|---|---|
| `RFNAME` | [−] | Input filename for a rail profile. |
| `WFNAME` | [−] | Input filename for a wheel profile. |

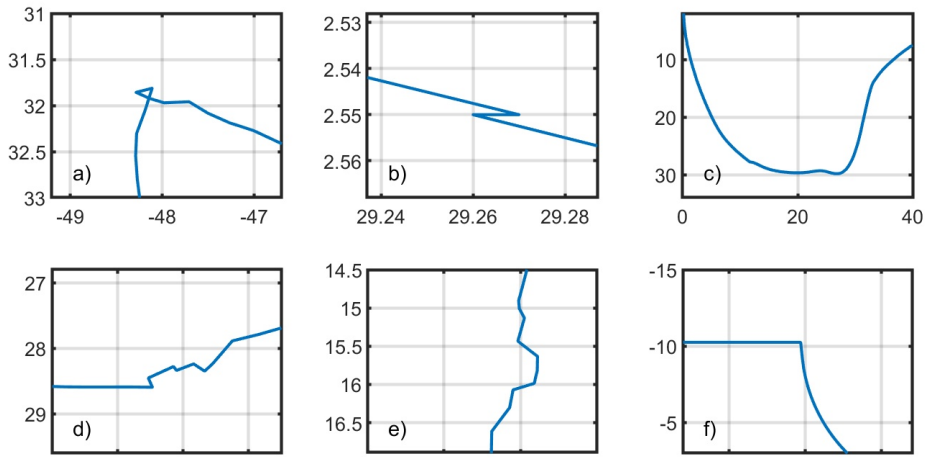



Figure 3.8: *Challenging features that may occur in wheel and rail input profiles.*

**Form as needed by CONTACT** Profiles are transformed into a canonical form to simplify usage and processing in CONTACT. This form considers the wheel and rail on the right side of the wheelset and track, with lateral $y$-coordinates increasing towards the field side. Optional mirroring is provided for cases where left side profiles are given in the actual files.

The canonical form uses mm as the unit of length and has positive $z$ pointing downwards. Optional scaling and mirroring are provided to bring the data into this form. For rails, points will be ordered from the track center to the field side, with $y_r$ coordinates in ascending order, whereas for wheels, $y_w$ will be in descending order. This makes that going through the points, the material stays on the right hand side. The points are reordered automatically if needed to get them in the right order.

| | | |
|---|---|---|
| `MIRROR_Y` | [−] | Use the profile as is (−1 or 0) or mirror $y$-values (1). |
| `MIRROR_Z` | [−] | Auto-detect whether mirroring of $z$-values is needed (0, default), use the profile as is (−1), or mirror its $z$-values (1). |
| `SCALE_YZ` | [−] | Scale factor for unit conversion, e.g. `scale_yz = 1000` to convert profile values given in [m] to [mm]. |

SIMPACK profiles are processed first according to the options that are specified in the `prr`- or `prw`-file (mirroring, reordering, clipping), except smoothing. After this, the conversions specified in the `inp`-file are also applied, concerning mirroring, scaling, and smoothing. Miniprof profiles have no options specified in the `ban`- and `whl`-files themselves. They are subject to the conversions specified in the `inp`-file provided to CONTACT.

**Curation of anomolous input data** Before the application of profile smoothing, CONTACT will try to detect and resolve specific aspects in the input data. A number of possible situations is shown in Figure 3.8.

- Figure 3.8 (a) shows a profile containing a loop, with different segments crossing each other.

CONTACT will detect such cases and reject the profile with an error code/error message.

- Figure 3.8 (b) shows a 'zig-zag pattern' that happens due to improper stitching of profile sections. The same point is repeated as the end-point of one section and the beginning of the next one, with slight offset in coordinate values. CONTACT will detect this pattern, delete the two points, and replace by the average of the two.
- Figure 3.8 (d) shows irregular data in a wheel profile, at the bottom of the flange. This happens due to the measurement system producing mixed signals in this region. The feature may be kept in the profile as long as there's no contact at the bottom of the flange. Else, it may be attenuated or removed using profile smoothing.
- Figure 3.8 (c) shows a wheel profile with an anomalous 'dolphin nose shape'. This may have been introduced by discarding of points with apparent measurement error (cf. situation (d)) followed by agressive smoothing. CONTACT cannot detect and repair this situation.
- Figure 3.8 (e) shows a local dent in the rail side face. It is up to the user to decide whether this feature is real and should be kept in the profile, or unwanted and to be corrected by smoothing.
- Figure 3.8 (f) shows a sharp corner in a profile used for deep groove rolling. Similar corners may occur at the transition from a roller surface to the side faces, and should be maintained in the profile smoothing procedure.

The zig-zag pattern of Figure 3.8 (b) is detected on the basis of changes in surface inclination. A zig-zag is found when both $|\delta\alpha_i|$ and $|\delta\alpha_{i+1}|$ are larger than the threshold $\Delta\alpha_{zig}$.

| | | |
|---|---|---|
| `ZIGTHRS` | [rad] | Angle threshold $\Delta\alpha_{zig}$ for detection of the zig-zag pattern. Default $5/6\,\pi$ rad. Should be larger than $\pi/4$, set to $\geq \pi$ to disable zig-zag detection. |

The situations of Figures 3.8 (d)–(f) are dealt with by detecting sharp corners, called 'kinks', in the input profile. A distinction is made between 'true kinks' that should be kept in the profile and 'roughness' that should be reduced by smoothing. The distinguishing feature is that true kinks are isolated. A true kink consists of a change in surface inclination $\delta\alpha_i$ larger than the high threshold, with no change of inclination $\delta\alpha_j$ in nearby points exceeding the low threshold.

| | | |
|---|---|---|
| `KINKHIGH` | [rad] | Angle threshold $\Delta\alpha_{high}$ for kink pattern detection. Default $\pi/6$ rad. Set to $\geq \pi$ to disable kink detection. |
| `KINKLOW` | [rad] | Angle threshold $\Delta\alpha_{low}$ for neighbouring points at kink pattern detection. Default $\Delta\alpha_{high}/5$. |
| `KINKWID` | [mm] | Half-width of winduw used for kink pattern detection. Default: checking profile points within 2 mm on either side of possible kink locations. |

**Profile smoothing** Smoothing of the profiles may be needed to reduce measurement noise or to suppress small scale features that raise contact pressures in a way undesired to the purpose of one's simulation [59]. Three different smoothing methods are currently provided:

0. An unweighted smoothing spline as presented in [57]. This method is considered obsolete. It is superseded by the weighted splines, and kept in the program chiefly for backward compatibility with earlier versions.

   In this method, the required value of $\lambda$ depends on the number of points in the profile. If a five times higher sampling density is used, then $\lambda$ needs to be increased five times to get the same level of smoothing. For profiles with more or less uniform spacing of the points, this method may be replaced by method 1 using

   $$\bar{\lambda}^{(2)} = \text{mean}(\delta s_i) \cdot \lambda. \tag{3.1}$$

1. The weighted smoothing spline with parameter $L_{filt}$ and second order penalty. Denoting the input profile points as $\mathbf{x}_i$, at arc-length position $s_i$, and the smoothing spline as $\mathbf{f}(s)$, this minimizes the following function:

   $$\sum_i w_i \left(\mathbf{x}_i - \mathbf{f}(s_i)\right)^2 + \bar{\lambda}^{(2)} \int \mathbf{f}''(s)^2 \, ds, \quad w_i = \delta s_i = \frac{s_{i+1} - s_{i-1}}{2}. \tag{3.2}$$

   The first part describes the adjustments made to the input data, while the latter part places a penalty on the roughness (curvature) of the spline.

   The value of $\bar{\lambda}^{(2)}$ is computed from the input $L_{filt}$ as

   $$\bar{\lambda}^{(2)} = \left(\frac{L_{filt}}{2\pi}\right)^4 \quad \leftrightarrow \quad L_{filt} = 2\pi \left(\bar{\lambda}^{(2)}\right)^{1/4}. \tag{3.3}$$

   By this approach, the spline acts like a low pass filter, diminishing fluctuations with wavelengths $L < L_{filt}$ and keeping fluctuations in the output with $L > L_{filt}$ [59].

2. (Reserved for weighted B-spline smoothing with third order penalty.)

   $$\bar{\lambda}^{(3)} = \left(\frac{L_{filt}}{2\pi}\right)^6 \quad \leftrightarrow \quad L_{filt} = 2\pi \left(\bar{\lambda}^{(3)}\right)^{1/6}. \tag{3.4}$$

| | | |
|---|---|---|
| `ISMOOTH` | [−] | Use smoothing method 0 (unweighted spline), 1 (weighted with 2nd order penalty) or 2 (weighted with 3rd order penalty). |
| `LAMBDA` | [−] | Input parameter $\lambda$ for the unweighted smoothing spline (`ismooth` = 0), using $\lambda \leq 0$ to disable smoothing. |
| `L_FILT` | [mm] | Wavelength $L_{filt}$ for which fluctuations are approximately halved by the weighted spline approaches (`ismooth` = 1, 2). Set $L_{filt} \leq 0$ to disable smoothing. |
| `SMOOTH` | [−/mm] | A single parameter `smooth` is used in the input, with $\lambda$ = `smooth` when `ismooth` = 0 and $L_{filt}$ = `smooth` when `ismooth` = 1, 2. |

**Rail slices-file** A rail slices-file starts with configuration data, and then gives a list of rail profiles with corresponding $s_{fc}$ positions along the track curve, and concludes with information on 'features' and corresponding 'parts' in lateral direction.

```
% data for S+C benchmark, UK crossing '56E1-R245-1:9.25'

   0.000    1000.0   S_OFFSET [m], S_SCALE [mm/m]
     148             NSLC
       7    0    0   NFEAT, NKINK, NACCEL
       1             S_METHOD

%  - slice positions S_SLC [m] and filenames RFNAME per slice

 -75.00000   '../switch-benchmark/DataSet1_56E1_v2/Crossing/Crossing_1.txt'
 -49.33684   '../switch-benchmark/DataSet1_56E1_v2/Crossing/Crossing_2.txt'
 -24.66842   '../switch-benchmark/DataSet1_56E1_v2/Crossing/Crossing_3.txt'
  -0.99474   '../switch-benchmark/DataSet1_56E1_v2/Crossing/Crossing_4.txt'
          ...
```

| | | |
|---|---|---|
| S_OFFSET | [x] | Offset $s_{ofs}$ added to input $s_{slc}$ positions to align with track $s_{fc}$-coordinates as used in CONTACT, with dimension 'x' defined by the user. |
| S_SCALE | [mm/x] | Multiplication factor to convert input $s_{slc}$ positions to mm. |
| S_SLC | [x] | Input $s_{slc}$-position of a slice along the track curve, converted to $fc$ coordinates as $s_{fc} = s_{scale} \cdot (s_{ofs} + s_{slc})$. |
| S_METHOD | [−] | Interpolation method used between successive slices. Method 1 is to use an interpolating spline, whereas method 2 uses a more smooth approximation that does not pass through the data positions. |

Note that the example has $s_{slc}$-positions in meters, converted to millimeters using $s_{scale} = 1000$. The slices must be given with $s_{slc}$-positions in increasing order and at least 0.001 mm apart.

**3D surface building** The slices will each be resampled to a uniform spacing in order to define 'interpolation paths' in longitudinal direction. To obtain a good 3D surface model, interpolation paths need to have smooth functions $y(s_{fc}), z(s_{fc})$, and should therefore not run across sharp geometrical features. This is steered by dividing slices laterally into different 'parts' that are resampled separately as illustrated in Figure 3.9. The break points in a slice, also called 'features', are $s_f$, $f = 1 \cdots n_{feat}$, with $s_f$ the 'summed chord length' to the start of the slice (see below) given in mm.

```
%  - S_F positions at geometrical features at start/end of each part per slice
%  S_SLC    S_F [mm],  F = 1 .. NFEAT
          ...
 0.08076     -1      -1       -1       -1       0.      30.27   999.
```

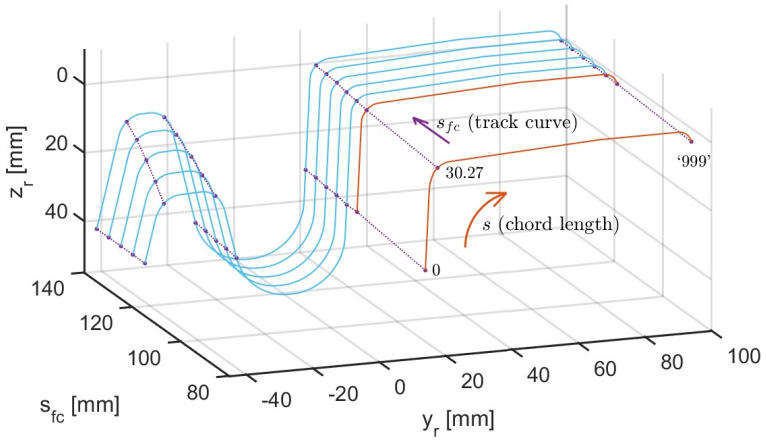



Figure 3.9: *Successive profile slices defining the geometry for a crossing, with geometrical features connected into 'interpolation paths'.*

```
0.11395     -1      -1      -1      -1      0.      29.93   999.
0.11894      0.    18.00   34.51   54.51  110.53  141.01   999.
0.12393      0.    22.00   36.99   60.49  116.98  147.21   999.
0.12892      0.    25.74   40.48   66.96  122.92  153.63   999.
          ...
```

In cases where certain features do not occur in a slice, this is indicated using a negative value for the corresponding $s_f$. This implicitly defines 'interruptions' of the profile in longitudinal direction.

The start and end of the profile are included in the definition of features, permitting trimming of the input slices. No input is needed when $n_{feat} \leq 1$, using default values $s_1 = 0$ and $s_2 = 10^6$ mm covering the whole profile.

For a profile with $n$ points $(y_i, z_i)$ the summed chord length is computed as

$$\delta y_i = y_{i+1} - y_i,\ \delta z_i = z_{i+1} - z_i,\ \delta s_i = \sqrt{\delta y_i^2 + \delta z_i^2},$$
$$s_1 = 0,\ s_{i+1} = s_i + \delta s_i,\ \ i = 1 \cdots n-1. \tag{3.5}$$

This uses the numbering of the points as obtained after mirroring and reordering, with point 1 at the track center side of a rail profile and at the field side of a wheel profile.

The file-structure for slices files is specified in Appendix A.3.

**Wheel slices-file** Wheel slices-files are structured similarly as rail slices files, except that longitudinal $s_{fc}$ positions along the track curve are replaced by angles $\theta_{wc}$ around the circumference of the wheel, and vertical heights $z$ are replaced by radial heights $dr$ with respect to the nominal radius.

The wheel surface is defined on the domain $\theta_{wc} = [-\pi, \pi)$ rad using periodical extension for evaluation at other values. The slices file does not need to cover the domain fully, especially for localized

defects such as a wheel flat. If data are given on a smaller interval $[\theta_0, \theta_1]$ fully contained in $[-\pi, \pi)$, constant extrapolation is used for $\theta_{wc} \in [-\pi, \theta_0)$ and $\theta_{wc} \in (\theta_1, \pi)$. This works best with small derivatives $\partial dr_{wc}/\partial\theta$ at $\theta_0$ and $\theta_1$.

If out-of-roundness exists around the whole circumference of the wheel, as for polygonization, then the input data need to be extended periodically by the user in order to create a wheel surface with smooth wrap-around from $\pi$ to $-\pi$. Inputs should be given on a domain such as $[\theta_0, \theta_1] = [-3.2, 3.2]$ rad, fully encompassing the range $[-\pi, \pi)$. Periodic extension means that $dr(\theta) = dr(\theta + 2\pi)$ for $\theta \in [\theta_0, -\pi)$ and $dr(\theta) = dr(\theta - 2\pi)$ for $\theta = [\pi, \theta_1)$.

| | | |
|---|---|---|
| `TH_OFFSET` | [x] | Offset $\theta_{ofs}$ added to input $\theta_{slc}$ positions to align with wheel $\theta_{wc}$-coordinates as used in CONTACT, with dimension 'x' defined by the user. |
| `TH_SCALE` | [rad/x] | Multiplication factor to convert input $\theta_{slc}$ positions to rad. |
| `TH_SLC` | [x] | Input $\theta_{slc}$-position of a slice along the wheel circumference, converted as $\theta_{wc} = \theta_{scale} \cdot (\theta_{ofs} + \theta_{slc})$. |
| `TH_METHOD` | [−] | Interpolation method used between successive slices. Method 1 is to use an interpolating spline, whereas method 2 uses a more smooth approximation that does not pass through the data positions. |

The slices must be given with $\theta_{slc}$-positions in increasing order and at least 0.001 rad apart. Slice positions can be given in degrees using $\theta_{scale} = \pi/180 = 0.0174533$.

## 3.3 Track and roller geometry

The track geometry consists of a rail profile, the track dimensions and track deviations. New geometry is input when the control integer ZTRACK is 1–3 (page 18).

In the computation of wheelsets on tracks (configurations $\mathtt{C}_1 = 0, 1$), 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 set using either the gauge width computation or using absolute placement.

The gauge point computation is selected by setting a positive 'gauge measuring height' $g_{height} > 0$.

| | | |
|---|---|---|
| `GAUGHT` | [mm] | Height $g_{height}$ below the track plane at which the gauge width is measured. |
| `GAUGSQ` | – | Reserved for selecting second or further gauge faces instead of the leftmost one (n.y.a.) |
| `GAUGWD` | [mm] | Distance $g_{width}$ between the inner faces of the two rails. |
| `CANT` | [rad] | Rail cant: rotation angle from track vertical ($z_{track}$) to rail vertical ($z_{rail}$), positive for inclination towards the track center. |

The gauge computation is disabled and replaced by absolute rail placement by setting $g_{height} \leq 0$. The rail datum point (origin) is then placed at the specfied position in the track $[0, y_{r(tr)}, z_{r(tr)}]^T$.

| | | |
|---|---|---|
| RAILY0 | [mm] | Position $y_{r(tr)}$ of the rail origin with respect to track coordinates. |
| RAILZ0 | [mm] | Position $z_{r(tr)}$ of the rail origin with respect to track coordinates. |

RAILY0 will typically be $< 0$ for the left rail and $> 0$ for the right rail. A value $y_{r(tr)} = 0$ may be appropriate for a configuration in which one wheel and one rail are used.

Zero cant is used in the computations for roller rigs ($C_1 = 4, 5$). Furthermore, the gauge computation is used there for the lateral positioning of the profile only. In radial direction, the profile origin $z_r = 0$ is placed at the nominal radius $r_{nom,r}$ from the roller axle.

| | | |
|---|---|---|
| NOMRADR | [mm] | Nominal radius $r_{nom,r}$ of the rollers, i.e. the radius at which the profile heights $z_r$ are zero. |

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 | [mm] | Offset $\Delta y_{rail}$ of the rail profile reference from the design to the actual position in track coordinates. |
| DZRAIL | [mm] | Offset $\Delta z_{rail}$ of the rail profile reference with respect to the design position in track coordinates. |
| DROLLR | [rad] | Rotation $\Delta\phi_{rail}$ of the rail profile, from design to actual 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 $v_\phi^{rail}$ of the rail origin with respect to the design orientation in track coordinates, positive using the right-hand rule. |

## 3.4 Wheelset geometry & state

The wheelset geometry and state consist of the wheelset dimensions, the wheel profile, and the position and velocity of the wheelset with respect to the track. New data are input when the control integer EWHEEL is 1–5 as described on page 19.

| | | |
|---|---|---|
| FBDIST | [mm] | Lateral distance $d_{flng}$ 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 3.6, $y_{fbpos}$ could be $-70$ mm.) |
| NOMRADW | [mm] | Nominal radius $r_{nom,w}$ of the wheel, i.e. the distance of the wheel profile reference point to the wheelset axle. |

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, 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 in terms of track coordinates. |
| `Z_WS` | [mm] | Vertical position $z_{ws}$ of the wheelset center in terms of track coordinates, zero at the nominal position $z_{tr} = -r_{nom,w}$. |
| `ROLL` | [rad] | Wheelset roll angle $\phi_{ws}$ with respect to the track plane. |
| `YAW` | [rad] | Wheelset yaw angle $\psi_{ws}$ with respect to the track center line $x_{track}$. |
| `PITCH` | [rad] | Wheelset pitch angle $\theta_{ws}$, i.e. rotation about the wheelset axle. This is of relevance primarily for out-of-round wheels using a wheel slices file. |

Using the $\mathtt{N}_1$-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.

In the computation of roller rigs, the variable $s_{ws}$ is replaced by $x_{ws}$, defined as follows:

| | | |
|---|---|---|
| `X_WS` | [mm] | Longitudinal position $x_{ws}$ of the wheelset center in terms of track coordinates. |

The orientation of roll, yaw and pitch angles is defined using the right hand rule. For instance, the configuration of Figure 3.6 shows a positive roll angle $\phi_{ws}$, 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 3.6 if a positive yaw angle is given. This shows up in the results of the Manchester benchmark example (Figure 5.10), 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 $\omega_{ws} = \dot{\theta}_{ws}$. |

In the computation of curving scenarios, the track coordinate system changes its forward direction by $1/r_{curv}$ rad/mm for a curve radius $r_{curv}$ in mm. The yaw velocity is measured relative to this

changing orientation, producing $\dot{\psi}_{ws} = 0$ (constant $\psi_{ws}$) in steady curving.

In the computation of roller rigs, the forward velocity $v_x$ is assumed to be negligible, and is replaced by the angular velocity of the rollers:

| | | |
|---|---|---|
| `RPITCH` | [rad/s] | Rollers' angular velocity $\omega_{rol} = \dot{\theta}_{rol}$. |

According to the right hand rule, the pitch velocity $\omega_{ws}$ is negative for forward rolling, whereas $\omega_{rol}$ is positive.

Flexible wheelset deviations provide offsets from the initial (design) geometry to the actual (current) configuration. Wheelset coordinates are used, such that $\Delta y > 0$ is towards the wheelset center for a left wheel, but to the field side when the right wheel is considered.

| | | |
|---|---|---|
| `DXWHL .. DZWHL` | [mm] | Displacements $\Delta x_{whl}, \Delta y_{whl}, \Delta z_{whl}$ of the wheel profile reference point from the design to the actual position in wheelset coordinates. |
| `DROLLW, DYAWW, DPITCHW` | [rad] | 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` | [mm/s] | 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` | [rad/s] | 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. |

## 3.5 Identification of contact patches

In module 1, CONTACT determines the regions of the wheel and rail where virtual interpenetration occurs, producing the potential extent of contact patches. Multiple contact patches may be detected. Patches that lie close together may be joined, to include interactions between them in the actual contact solution. See Figure 3.10 for a case with a conformal contact situation, solved three times with different settings.

The separation or combination of contact patches is governed using three threshold values for the distance $d$ between interpenetration areas, $d_{turn} \geq d_{sep} \geq d_{comb}$, and one threshold $\alpha_{sep}$ for the difference in contact reference angle. Two patches are processed independently when $|\alpha_{cp1} - \alpha_{cp2}| > \alpha_{sep}$ or when $d > d_{turn}$. Else,

- Two separate contact patches are used when $d_{sep} < d \leq d_{turn}$, with contact reference angles $\alpha_{cp1}$ and $\alpha_{cp2}$ turned towards their weighted average;
- One combined contact patch is used with a 'blending approach' when $d_{comb} < d \leq d_{sep}$, attenuating the cross-influence between the different parts;
- One combined contact patch is used with full solution when $d \leq d_{comb}$.

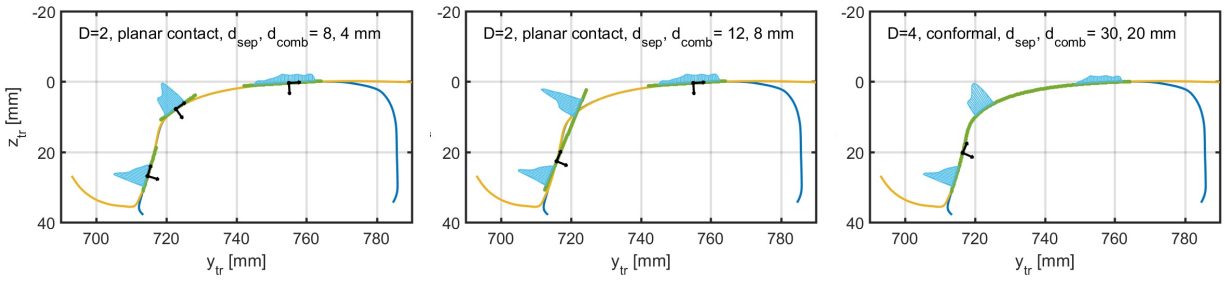



Figure 3.10: *Illustration of 'contact patches' with local coordinate systems used for the solution. Multiple 'interpenetration regions' may be combined into a single 'patch' if the distance and angle difference are smaller than the threshold values.*

The blending approach is described in [57, 56]. The turning of contact angles is a further extension, suppressing jump discontinuities in total forces at slight changes in the input positions.

| | | |
|---|---|---|
| `A_SEP` | [rad] | Threshold angle $\alpha_{sep}$ for the angle difference above which patches are processed separately. |
| `D_SEP` | [mm] | Threshold distance $d_{sep}$ above which patches are processed separately. |
| `D_COMB` | [mm] | Threshold distance $d_{comb}$ below which patches are combined fully, if not prohibited by the angle threshold. |
| `D_TURN` | [mm] | Threshold distance $d_{turn}$ below which reference angle turning is used for patches that are processed separately. Default $2d_{sep} - d_{comb}$. |

For each contact patch that remains after combination, CONTACT determines a so-called *contact reference position*. This is the origin of the local coordinates used for solving the contact problem [57]. 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.

In Figure 3.10, the contact reference position of each contact patch is indicated by a black marker. In the output it 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. |
| `DELTTR` | [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. |

These values are all given with respect to right handed coordinate systems with positive $y$ to the

right. For a left-side wheel/rail pair, positive $y_r$ is towards the track center, whereas for a wheel/rail pair on the right side, positive $y_r$ is towards the field side.[1]

## 3.6 Potential contact area and discretisation

The actual solution of pressures and frictional stresses is implemented using module 3 for basic contacts (Chapter 4). This is activated separately for each contact patch identified in the contact search phase.

A rectangular *potential contact area* is defined that encompasses 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 parameters, defined as follows:

| | | |
|---|---|---|
| `DX` | [mm] | Size $\delta x$ of each grid element in longitudinal ($x_{track}$ or $x_{rail}$) direction. |
| `DS` | [mm] | Size $\delta s$ 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 $\delta q$ traversed per time step compared to the grid size $\delta x$. |

The program automatically sets $c = 1$ in the computation of steady rolling. For transient rolling, the value 1 is also preferred [46, 73].

In module 1, CONTACT determines the undeformed distance function and the rigid slip at each element of the grid, as needed for the detailed contact calculation.

## 3.7 Material & friction description

The calculations for each contact patch need information on the materials (e.g. elastic half-space, simplified model), geometrical composition (interfacial layer, conformal geometry) and on the friction description (Coulomb friction, with velocity or temperature dependence). Additionally there are parameters for the calculation of surface temperatures and subsurface stresses, and for configuration of the solvers used. These parameters are the same in module 1 as in module 3, and are discussed in Chapter 4.

## 3.8 Global outputs for module 1

Global output quantities are written to the output file `<experim>.out` when the `0`-digit is set to values $1 - 5$.

[1] This convention was introduced in version v23.1.

The global part of the output firstly contains a description of the problem that is solved (`O` $\geq 2$), 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 that had to be found.

With positive $z$ pointing downwards (Figure 3.6), the rail (roller) 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 firstly the contact reference point that was discussed in Section 3.5. Next, there are the creepages and penetration, and the forces `FX`, `FS` and moment `MN` of each separate patch as obtained from module 3 (Section 4.7; `FY` and `MZ` are renamed to `FS`, `MN` to emphasize that these are given in contact local coordinate system).

# Chapter 4

# Inputs and outputs for module 3: basic contact

'Basic contact' means that the user takes care of locating the potential region where contact occurs, defining local coordinates with $z$ ($n$) pointing into the upper body, $a = 1$, and defining the contact problem using this coordinate system. The inputs needed are then:

- The material composition of the two bodies (§4.1);
- Frictional processes: (§4.2);
- Definition of the potential contact and its grid discretization (§4.3);
- The undeformed distance (§4.4) and relative motion between the surfaces (§4.5).

## 4.1 Material & geometrical composition

The simplest material model used in CONTACT concerns the elastic half-space (§4.1.1). Various extensions are provided that may change the material behavior,

- viscoelasticity (§4.1.3), simplified model (§4.1.4), local plasticity (§4.1.5),

and/or geometrical composition,

- interfacial layer (§4.1.5), and conformal shapes (§4.1.6).

### 4.1.1 Homogeneous elastic materials

The input parameters for elastic materials are:

| | | |
|---|---|---|
| `GG(ia)` | $[\mathrm{N/mm^2}]$ | Modulus of rigidity $G^{(a)}$ of body `ia`, $G^{(a)} > 0$. |
| `POISS(ia)` | $[-]$ | Poisson's ratio $\nu^{(a)}$ of body `ia`, $0 \le \nu^{(a)} \le 0.5$. |

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

$$G^{(a)} = \frac{E^{(a)}}{2(1+\nu^{(a)})} \tag{4.1}$$

From these values the program computes the following combined quantities:

| | | |
|---|---|---|
| `GA` | $[\mathrm{N/mm^2}]$ | Combined modulus of rigidity $G$. |
| `NU` | $[-]$ | Combined Poisson's ratio $\nu$. |
| `AK` | $[-]$ | Difference parameter $K$. |

These values are defined by

$$\frac{1}{G} = \frac{1}{2}\left(\frac{1}{G^{(1)}} + \frac{1}{G^{(2)}}\right), \quad \frac{\nu}{G} = \frac{1}{2}\left(\frac{\nu^{(1)}}{G^{(1)}} + \frac{\nu^{(2)}}{G^{(2)}}\right), \quad \frac{K}{G} = \frac{1}{4}\left(\frac{1-2\nu^{(1)}}{G^{(1)}} + \frac{1-2\nu^{(2)}}{G^{(2)}}\right) \tag{4.2}$$

They characterize the combined influence function $\mathbf{A}(\mathbf{x}, \mathbf{x}')$ of the two bodies (equation (B.5)).

### 4.1.2 Heat related material parameters

The input parameters for the calculation of surface temperatures (`H`-digit, page 19) are:

| | | |
|---|---|---|
| `BKTEMP(ia)` | $[\,^\circ\mathrm{C}]$ | (Initial) Bulk temperature $T_0^{(a)}$ of body `ia`. |
| `HEATCP(ia)` | $[\mathrm{J/kg\,^\circ C}]$ | Specific heat capacity $c_p^{(a)}$ of body `ia`. |
| `LAMBDA(ia)` | $[\mathrm{W/mm\,^\circ C}]$ | Thermal conductivity $\lambda^{(a)}$ of body `ia`. |
| `DENS(ia)` | $[\mathrm{kg/mm^3}]$ | Density $\rho^{(a)}$ of body `ia`. |

An additional parameter $\beta_{pl}$ is needed in models employing an interfacial layer with plastic deformation (`M` = 4, §4.1.5). This parameter concerns the partitioning of plastic work into heat and energy stored in the material.

| | | |
|---|---|---|
| `BETAPL` | $[-]$ | Fraction of plastic work dissipated as heat. |

### 4.1.3 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 [17], appendix D, with further information provided in [70, 71].

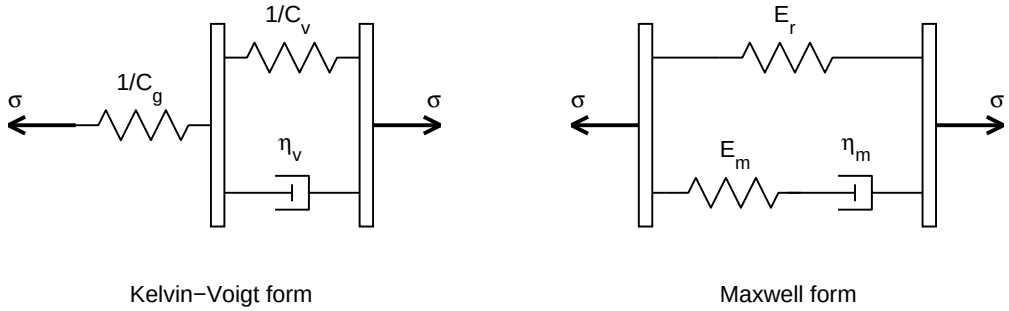



Figure 4.1: *Mechanical analog networks of the Standard Linear Solid (SLS) model of viscoelastic materials, featuring the equivalent Kelvin-Voigt (left) and Maxwell forms (right).*

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 [37].

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

$$C_{crp}(t) = \varepsilon(t)/\sigma_0 = C_g + (C_r - C_g)(1 - e^{-t/\tau_C}). \tag{4.3}$$

This employs the initial compliance $C_g$, the final compliance $C_r > C_g$ and the creep relaxation time $\tau_C$.[1]

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

$$E_{rel}(t) = \sigma(t)/\varepsilon_0 = E_r + (E_g - E_r)\, e^{-t/\tau_E}. \tag{4.4}$$

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

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 4.1. 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 $\nu$, 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

| | | |
|---|---|---|
| `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

[1] The subscripts $_g$ and $_r$ refer to the initial 'glassy' and the final 'rubbery' states of the viscoelastic material [37].

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

Table 4.1: *Relations of the viscoelastic input quantities to the model parameters for a number of different descriptions of viscoelastic relaxation.*

Table 4.1. From these relations it follows that

$$f_g = \frac{E_g - E_r}{E_r} \quad \leftrightarrow \quad E_r = \frac{E_g}{1+f_g}, \quad \text{and consequently} \quad G_r = \frac{G_g}{1+f_g}. \tag{4.5}$$

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.

### 4.1.4 Flexibilities of the simplified theory

CONTACT provides the 'Modified FASTSIM' algorithm [43] for quick approximation of the tangential surface tractions $\mathbf{p}_t$ using the simplified theory [13, 15]. This uses the simplified material model (`M` = 2, 3), also called the Winkler foundation approach. The surface particles are assumed to be moving independently of each other, with response $\mathbf{u}_t$ linear in the surface traction $\mathbf{p}_t$. This may be viewed as if the bodies consisted of a set of independent springs with (combined) flexibility parameter $L$.

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

In Hertzian problems (Sections 4.3.1–4.3.3), three different flexibilities $L'_\xi$, $L'_\eta$ and $L'_\phi$ may be used for longitudinal, lateral and spin creepage respectively. These flexibilities are computed automatically when `M` = 3, using the blending approach of [67]. For non-Hertzian problems (§4.3.4, 4.4), these flexibilities are based on an equivalent ellipse with semi-axes $a_{eqv}$, $b_{eqv}$, estimated for the actual contact area.

The modifications proposed in [43] are based on a variable flexibility, increasing with the ratio of

the slip area to the area of adhesion:

$$L_{eff} = \frac{L}{k}, \quad \text{with} \quad k = k_0 \left( \alpha_{inf} + \frac{1 - \alpha_{inf}}{1 + \beta\varepsilon} \right). \tag{4.6}$$

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

| | | |
|---|---|---|
| `K0_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 $\beta$ 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 $\varepsilon$ 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. For non-Hertzian cases, two approximate methods provided are KPEC (`B` = 5) or ANALYN (`B` = 6).

When using the simplified theory, the material parameters of Section 4.1.1 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.

### 4.1.5 Interfacial layer of contaminants

Option `M` = 4 concerns the contact between two homogeneous, elastic bodies as described in Section 4.1.1, separated by a so-called third-body layer that may be formed of metal oxides (wear debris), friction modifiers, sand and clay, etc. This model is shown schematically in Figure 4.2.

The layer is sheared elastically with

$$\mathbf{u}_{el}^{(3)} = \mathbf{p}_t^{(1)} \cdot \frac{h^{(3)}}{G^{(3)}}. \tag{4.7}$$

Here $G^{(3)}$ is the shear modulus of the layer's material, and $h^{(3)}$ the layer thickness.

Next, plasticity is introduced in the layer at locations where the tangential stress reaches the yield limit $\tau_c$. This yield limit is characterized using linear work-hardening or work-softening according to the characteristics of Figure 4.3, right:

$$\tau_c = \tau_{c0} + k_\tau u_{pl}^*, \quad u_{pl}^* = \int \|\dot{\mathbf{u}}_{pl}\| \, dt. \tag{4.8}$$

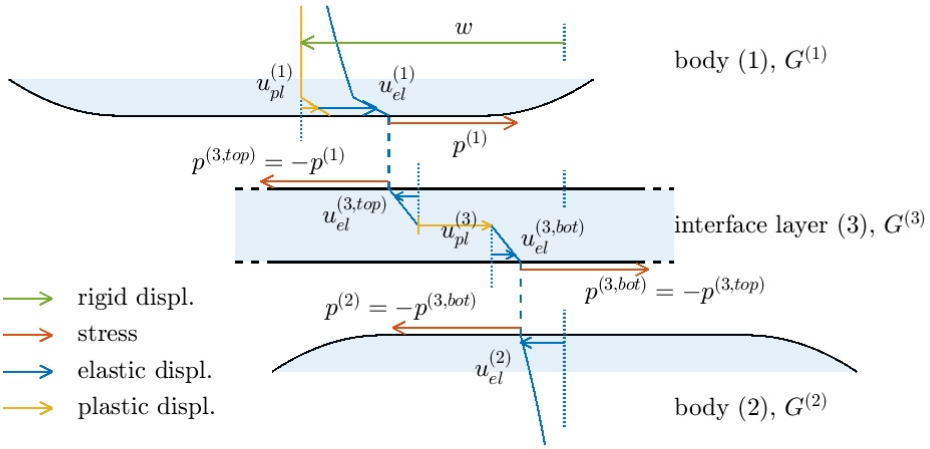



Figure 4.2: *Illustration of tractions $p^{(a)}$ acting on bodies $a = 1..3$ and displacements $u^{(a)}$ in the bodies as a result of (gross) relative movement (creepage) $w$, in absense of micro-slip $s$.*

Here $\tau_{c0}$ is the initial yield limit of fresh material, entering the contact area, while $u_{pl}^{*}$ is the amount of plastic deformation accumulated while passing through the contact. Work-hardening is obtained using $k_\tau > 0$, whereas work-softening occurs when $k_\tau < 0$. The value $k_\tau = 0$ results in elastic-perfectly plastic behavior.

The slopes $k_u$ and $k_\tau$ used in Figure 4.3 are related by

$$k_\tau = \frac{G^{(3)}/h^{(3)} \cdot k_u}{G^{(3)}/h^{(3)} - k_u}, \quad k_u = \frac{G^{(3)}/h^{(3)} \cdot k_\tau}{G^{(3)}/h^{(3)} + k_\tau}. \tag{4.9}$$

In the input, the strength of the interfacial layer is described by four parameters:

| | | |
|---|---|---|
| `GG3` | [N/mm$^2$] | Shear elastic modulus of the interface layer, $G^{(3)} > 0$. |
| `LAYTHK` | [mm] | Thickness of the interface layer, $h^{(3)} \geq 0$. |
| `TAU_C0` | [N/mm$^2$] | Initial yield limit $\tau_{c0}^{(3)}$ at which plasticity starts to occur. No plasticity is computed when $\tau_{c0} \leq 0$. |
| `K_TAU` | [N/mm$^3$] | Rate of increase $k_\tau$ of the yield limit $\tau_c$ with accumulated plastic deformation. |

It is possible to use the tangential plasticity feature of CONTACT without using a third body layer. This is accomplished by setting $h^{(3)} = 0$. In that case, plasticity is thought to occur near the surface of the softer one of the primary bodies, cf. $u_{pl}^{(1)}$ in Figure 4.2.

### 4.1.6 Influence coefficients for conformal contact

The half-space approach relies on contact patches that are small compared to the characteristic sizes of the two bodies. This assumption may be violated if the contact patch is curved, for instance at

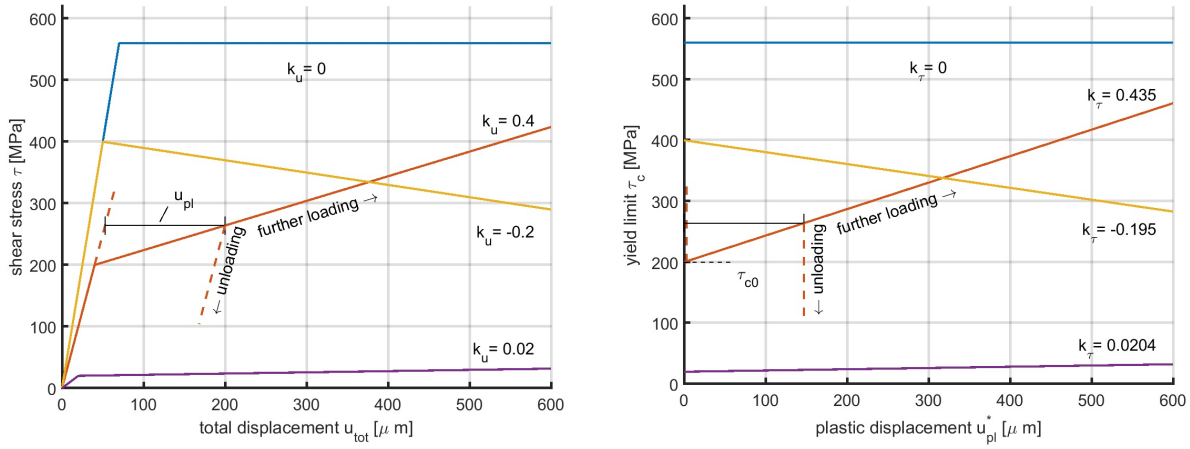



Figure 4.3: *Left: Interfacial layer characteristics cf. Hou et al. [10]. Right: corresponding change of yield limit $\tau_c$ with accumulated plastic deformation: work-hardening ($k_\tau > 0$) or softening ($k_\tau < 0$), or elastic-perfectly plastic behavior ($k_\tau = 0$).*

the rail gauge corner (Figure 5.14). Conformal contacts therefore require extension of the half-space approach, accounting for the conformal shapes of the two bodies. This is provided in CONTACT through influence coefficient options $\mathtt{C}_3 = 4$ and 9 (page 18).

Influence functions ($\mathbf{A}(\mathbf{x}, \mathbf{x}')$ of equation (B.5)) describe how a body deforms at one position ($\mathbf{x}$) when loaded at another position ($\mathbf{x}'$). This relationship depends on the shape of the body:

- A quarter-space that's loaded near the corner, will have larger elastic deformations than an elastic half-space with the same load (Figure 4.4, middle/right).
- Similarly, a three-quarterspace will have smaller deformations than an elastic half-space.

These differences between convex and concave bodies introduce a coupling between normal and tangential contact problems.

The influence functions of the two bodies may be calculated separately, using finite element methods, for instance, and provided to CONTACT in tabular form, using an auxiliary file. This option is activated by selecting $\mathtt{C}_3 = 9$ (page 18). 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 in Section A.4. |

A quick way to estimate these influence functions was presented by Blanco-Lorenzo et al. [2, 3]. This uses the surface inclination $\alpha$ to make a first order correction. This is illustrated in Figure 4.5 for pressures $p_n$ exerted at $J$, and their influence on elastic displacements at $I$. This option is provided in CONTACT for prismatic bodies, where the bodies are conformal in $y-z$ directions only. It is activated with (experimental) $\mathtt{C}_3 = 4$.

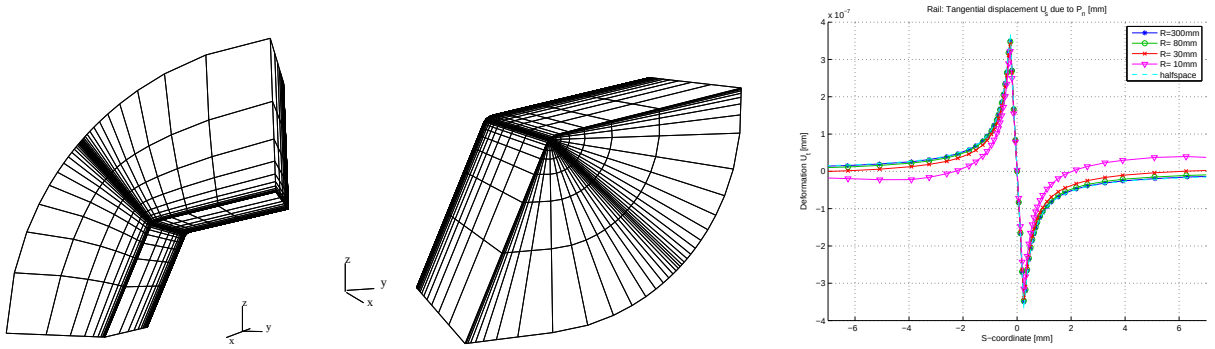



Figure 4.4: *Left/middle: finite element meshes used for numerical calculation of influence coefficients. Right: tangential response of the rail to a localized normal load for different radii of curvature, illustrating deviations from the half-space approach [64].*

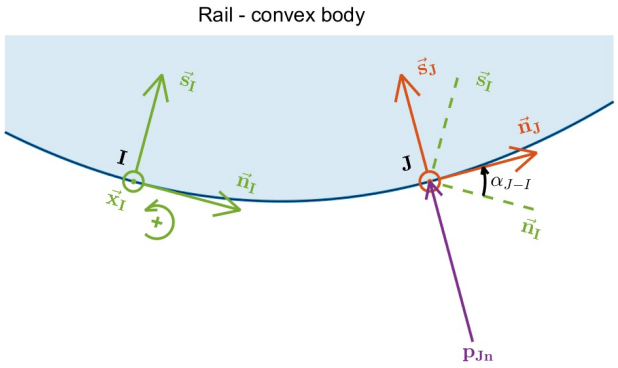



Figure 4.5: *Idea of Blanco's IF correction approach: the influence of loads exerted at J on the displacements at I depends on the angle variation $\alpha_J - \alpha_I$ between the two points.*

The main input is the lateral surface inclination, using a table $\{(y_j, \alpha_j)\}, j = 1 \cdots n$ with linear interpolation.

| | | |
|---|---|---|
| `NN` | – | Number of points $y_j = \tilde{s}_j$ used along the (curved) contact surface. |
| `Y(J)` | [mm] | Positions $y_j$ where $\alpha_j$ is given. |
| `ALPHA(J)` | [rad] | Surface inclination $\alpha_j$ at position $y_j$. |

Surface inclinations may be given relative to an arbitrary reference, e.g. using a global direction ($y_{tr}$), or a convienient (planar) local direction ($\bar{s}$). Positive rotation is defined using the right-hand rule as shown in Figure 4.5.

Two methods are provided that each come in four variants.

0. A fast approximation is provided using a linear fit of the table, corresponding to a constant radius of curvature.

1. The full method uses these surface inclinations as given, including all detail, which increases the computational work.

| | | |
|---|---|---|
| `IF_METH` | – | Fast (0) or detailed (1) method for surface inclinations. |
| `VARIANT` | – | Variant used for the IF correction (1–4). |

The variants concern the detailed formula used for the IF correction.

Numerically calculated influence coefficients can be combined with linearly elastic materials (`M` = 0, Section 4.1.1) and with interfacial layer (`M` = 4, Section 4.1.5). These material models will be used as 'best guess' when subsurface stresses are requested.

## 4.2 Friction description

The local coefficient of friction $\mu = \mu(\mathbf{x}, t)$ plays a central role in the build up of tangential stresses between the contacting bodies. Rather than being a constant, it may depend on the state of the two surfaces, i.e. the presence of contaminants and fluids, and the local surface temperature. This can be modelled using different approaches:

- The basic model consists of Coulomb friction with a prescribed, constant coefficient of friction.
- 'Falling friction' concerns situations where the total force is found to decrease with increasing creepage, after attaining a maximum value. This may be modelled with $\mu$ dependent on the slip velocity $s$.
- Another approach to falling friction uses the surface temperature calculation, with $\mu$ dependent on $T$.
- Different coefficients of friction may be experienced on the tread and flange of a wheel, for instance due to gauge face lubrication. This is facilitated using the `V`-digit in module 1 (Section 2.3.2), using different friction parameters for different sections on the rail.

### 4.2.1 Dry Coulomb friction

The basic friction law that is used by the program is dry, Coulomb friction with a single coefficient of friction,

$$\mu(\mathbf{x}, t) = \mu_{stat} = \mu_{kin}. \tag{4.10}$$

This option is obtained when `L` = 0. Two parameters must be given for backward compatibility reasons.

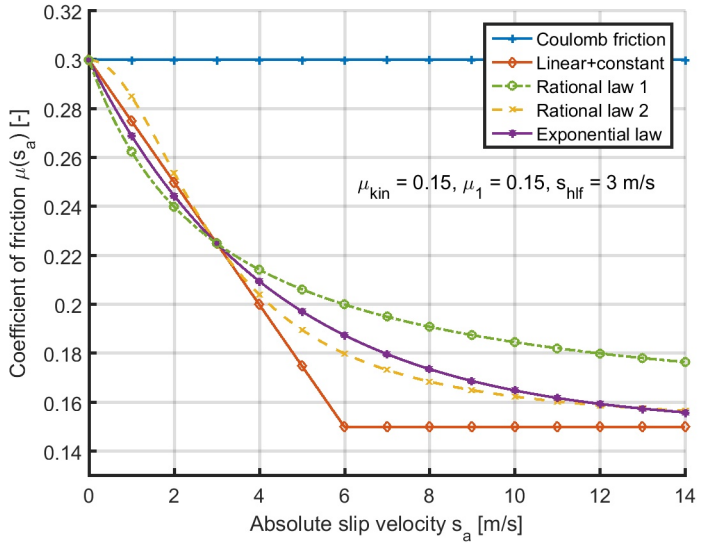



Figure 4.6: *Dependence of friction coefficient $\mu$ on absolute slip velocity $s_a$ for the friction laws of equations (4.11)–(4.14).*

| | | |
|---|---|---|
| `FSTAT` | [−] | Static coefficient of friction, $\mu_{stat} > 0$. |
| `FKIN` | [−] | Kinetic coefficient of friction $\mu_{kin} = \mu_{stat}$. |

Previous versions of CONTACT allowed $\mu_{stat}$ to be larger than $\mu_{kin}$. This is no longer supported, because the results are mostly the same compared to using $\mu_{kin}$ throughout the entire contact. This was discussed by Nielsen and Theiler for 2D situations [32]. The remaining differences due to $\mu_{stat} > \mu_{kin}$ are considered artifical effects of grid discretization.

### 4.2.2 Slip velocity dependent friction

Other friction laws, particularly concerning slip-velocity dependent friction, have been implemented too. These are illustrated in Figure 4.6. Results for these methods are included in the example in Section 5.9.

When `L` = 4, an exponential decrease of the friction coefficient is used:

$$\mu_s(s_a) = \mu_{kin} + \mu_{exp1} \exp(-\log(2) \cdot s_a/s_{h1}) + \mu_{exp2} \exp(-\log(2) \cdot s_a/s_{h2}) \tag{4.11}$$

Here $s_a(\mathbf{x}, t)$ is the magnitude of the absolute slip velocity ($\geq 0$) at position $\mathbf{x}$.[2] $\mu_{exp1}$, $\mu_{exp2}$, $s_{h1}$ and $s_{h2}$ are the coefficients of the friction law. Two terms are provided for flexibility, and allow two different time-scales to be incorporated. $\mu_{exp1}$ and $\mu_{exp2}$ are the maximum sizes of the exponential

[2] In 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.

terms (at $s_a = 0$, $\mu_{stat} = \mu_{kin} + \mu_{exp1} + \mu_{exp2}$). Setting one of these to zero disables a term. $s_{h1}$ and $s_{h2}$ are the absolute slip velocities at which the size of the terms is halved compared to $s_a = 0$.

| | | |
|---|---|---|
| `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. |
| `FEXP1` | [−] | Coefficient $\mu_{exp1}$ of (4.11), used when `L` = 4. |
| `FEXP2` | [−] | Coefficient $\mu_{exp2}$ of (4.11), used when `L` = 4. |

Note that equation (4.11) incorporates Polach's exponential formula [35],

$$\mu_s(s_a) = \mu_{stat}\left((1 - A)e^{-Bs_a} + A\right). \tag{4.12}$$

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× smaller than the reference values provided in [35], with unit [s/m].

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

$$\mu_s(s_a) = \mu_{kin} + \mu_{lin1} \cdot \max(0, 1 - s_a/2s_{h1}) + \mu_{lin2} \cdot \max(0, 1 - s_a/2s_{h2}) \tag{4.13}$$

$\mu_{lin1}$ and $\mu_{lin2}$ are the maximum sizes of the linear terms.

| | | |
|---|---|---|
| `FLIN1` | [−] | Coefficient $\mu_{lin1}$ of (4.13), used when `L` = 2. |
| `FLIN2` | [−] | Coefficient $\mu_{lin2}$ of (4.13), used when `L` = 2. |

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

$$\mu_s(s_a) = \mu_{kin} + \frac{\mu_{rat1}}{1 + s_a/s_{h1}} + \frac{\mu_{rat2}}{1 + (s_a/s_{h2})^2} \tag{4.14}$$

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

| | | |
|---|---|---|
| `FRAT1` | [−] | Coefficient $\mu_{rat1}$ of (4.14), used when `L` = 3. |
| `FRAT2` | [−] | Coefficient $\mu_{rat2}$ of (4.14), used when `L` = 3. |

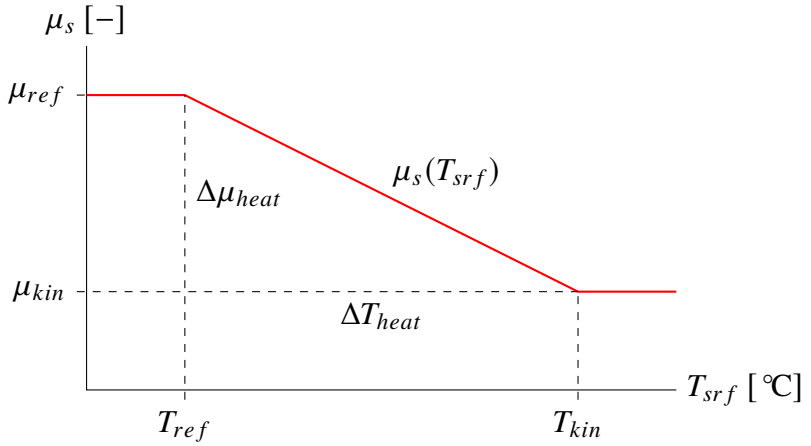



Figure 4.7: *Piecewise linear dependence of friction on surface temperature cf. equation (4.15). In this case $\Delta\mu_{heat}$ is negative, resulting in a decreasing coefficient of friction.*

### 4.2.3 Temperature dependent friction

Option `L` = 6 uses a piecewise linear relationship between the surface temperatures $T_{srf}$, computed with the `H`-digit, and the coefficient of friction $\mu$:

$$\mu_s(T_{srf}) = \begin{cases} \mu_{ref} & T_{srf} \le T_{ref} \\ \mu_{ref} + \Delta\mu_{heat}\frac{T_{srf}-T_{ref}}{\Delta T_{heat}} & T_{ref} \le T_{srf} \le T_{kin}, \quad T_{kin} = T_{ref} + \Delta T_{heat} \\ \mu_{kin} & T_{srf} \ge T_{kin}, \quad \mu_{kin} = \mu_{ref} + \Delta\mu_{heat} \end{cases} \tag{4.15}$$

Here $T_{srf}$ is the bulk surface temperature in ℃, $T_{ref}$ is the lower temperature at which $\mu$ starts changing, $T_{kin}$ is the upper temperature at which $\mu$ stops changing, $\mu_{kin}$ is the ultimate value for $\mu$ at high temperatures, and $\Delta\mu_{heat}$ the temperature dependent part of $\mu$. Equation (4.15) is illustrated in Figure 4.7.

| | | |
|---|---|---|
| `FREF` | [−] | Reference value $\mu_{ref} > 0$ for the coefficient of friction at low surface temperatures. |
| `TREF` | [℃] | Reference temperature $T_{ref}$ of equation (4.15). |
| `DFHEAT` | [−] | Coefficient $\Delta\mu_{heat}$ of (4.15), with $\mu_{ref} + \Delta\mu_{heat} > 0$. |
| `DTHEAT` | [℃] | Coefficient $\Delta T_{heat} > 0$ of (4.15). |

Equation (4.15) allows for gradual transitions, for instance using $T_{ref} = 0$ ℃ and $\Delta T_{heat} = 800$ ℃, as well as sharp transitions at any given temperature, e.g. $T_{ref} = 400$ ℃, $\Delta T_{heat} = 100$ ℃. Positive values for $\Delta\mu_{heat}$ may be used, to explore the effects of friction increasing with temperature.

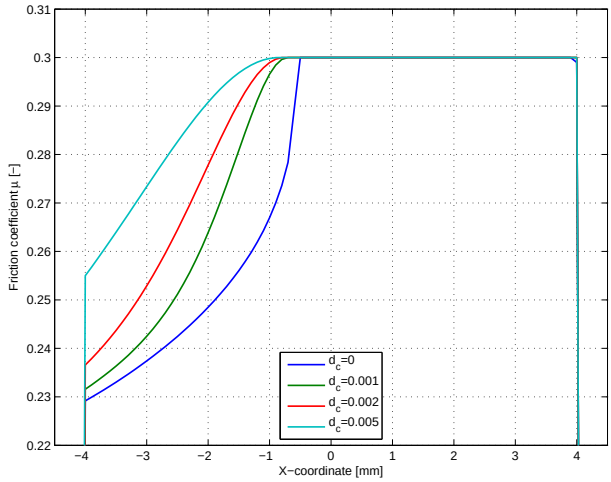



Figure 4.8: *Typical effect of friction memory (equation (4.16)) on the friction coefficient $\mu$: gradual instead of abrupt change of $\mu(\mathbf{x}, t)$.*

### 4.2.4 Friction memory

The friction laws of Section 4.2.2 lead to unwanted effects in transient calculations when they are based on the instantaneous slip velocity $s_a$ at time $t$. This is analysed in [63]. 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 4.8.

To circumvent these unwanted effects, the coefficient $\mu_s$ 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 $\mu$ and $s_a$:

$$\dot{\mu}(\mathbf{x}, t) = -\frac{\max(s_a(\mathbf{x}, t), s_0)}{d_c} \left(\mu(\mathbf{x}, t) - \mu_s(s_a(\mathbf{x}, t))\right) \tag{4.16}$$

This form implies that the transient behavior consists of an exponential decay towards the steady state value $\mu_s$. $d_c$ is the characteristic sliding distance over which the adaptation occurs (typically in the order of $\mu m$). Its effect is illustrated in Figure 4.8, showing the actual friction coefficient $\mu(\mathbf{x}, t)$ for a steady rolling cylinder. The coefficient $s_0$ is a small velocity ($O(mm/s)$) 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 ($\geq 0$). An instantaneous friction law is obtained by setting $d_c = 0$. |
| `MEM_S0` | [mm/s] | Minimum velocity $s_0$ for the friction memory effect ($\geq 0$). |

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

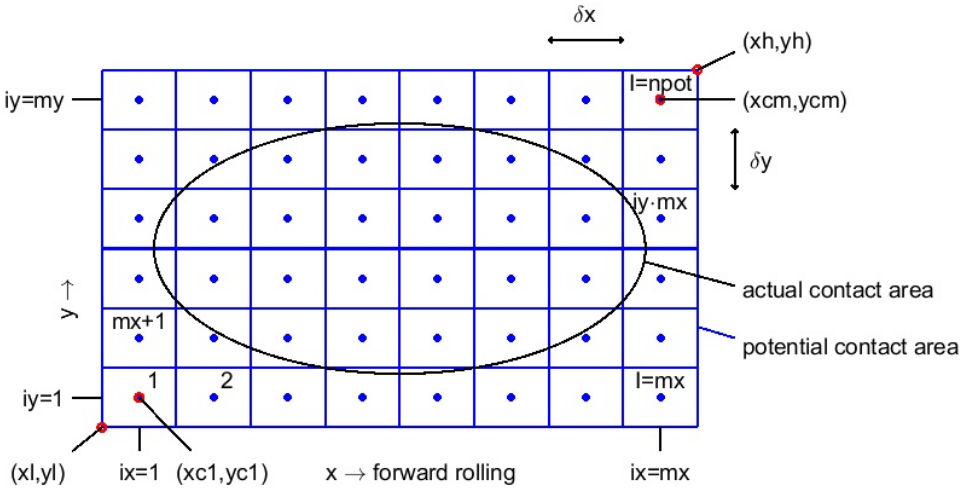



Figure 4.9: *The potential contact area, its discretisation, and the numbering of the elements.*

### 4.2.5 Friction variation

The input parameters for the friction laws `L` = 0, 2–4, 6 as discussed above, may be varied laterally, along the rail profile (module 1), using the option `V` = 1. The parameters are specified at two or more control points.

| | | |
|---|---|---|
| `NVF` | | Number of control points $n_{vf}$ for friction variation. |

Linear interpolation is used between these points, while the parameters are held constant outside of this range.

The control points on the rail profile are defined indirectly, using the rail surface inclination:

| | | |
|---|---|---|
| `ALPHVF` | [rad] | Rail surface inclinations at the control points, $\alpha_{vf,i}, i = 1 \cdots n_{vf}$. |

The surface inclination is measured in rail profile coordinates, from the horizontal $y_r$-axis to the inclined surface. Negative values are found on gauge face, small values at the top of the rail, increasing to 90° at the field side of the rail. The values must be entered in increasing order.

## 4.3 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 $mx \times my$ rectangular elements of size $\delta x \times \delta y$, see Figure 4.9.

| | | |
|---|---|---|
| `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` · `MY`. |

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

$$I = ix + (iy - 1) \cdot mx. \tag{4.17}$$

The centers of the elements are given by:

$$\mathbf{x}_I = \left[ x_l + (ix - \frac{1}{2}) \cdot \delta x, y_l + (iy - \frac{1}{2}) \cdot \delta y \right]^T \tag{4.18}$$

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.
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. |

### 4.3.1 Elliptical contacts – 3D Hertzian geometries

The variables in this section concern the geometry of the bodies when the problem is Hertzian, with constant radii of curvature in $x$- and $y$-directions. 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

$$\texttt{A1} = \frac{1}{2R^{(1)}_{x,eff}} + \frac{1}{2R^{(2)}_{x,eff}}, \quad \texttt{B1} = \frac{1}{2R^{(1)}_{y}} + \frac{1}{2R^{(2)}_{y}}. \tag{4.19}$$

In wheel-rail contact, the effective rolling radius for the wheel is $R_{x,eff} = R_w(y)/\cos(\delta)$ [55]. That is, the contact angle $\delta$ makes the surface look flatter in rolling direction. This is illustrated in Figure 4.10 using constant wheel radius $R_w$. (In practice, $R_w$ varies along the profile. This is a second order effect, because the actual values of $R_w$ are much bigger than shown in the figure.)

The Hertzian solution may be described in the input in different combinations:

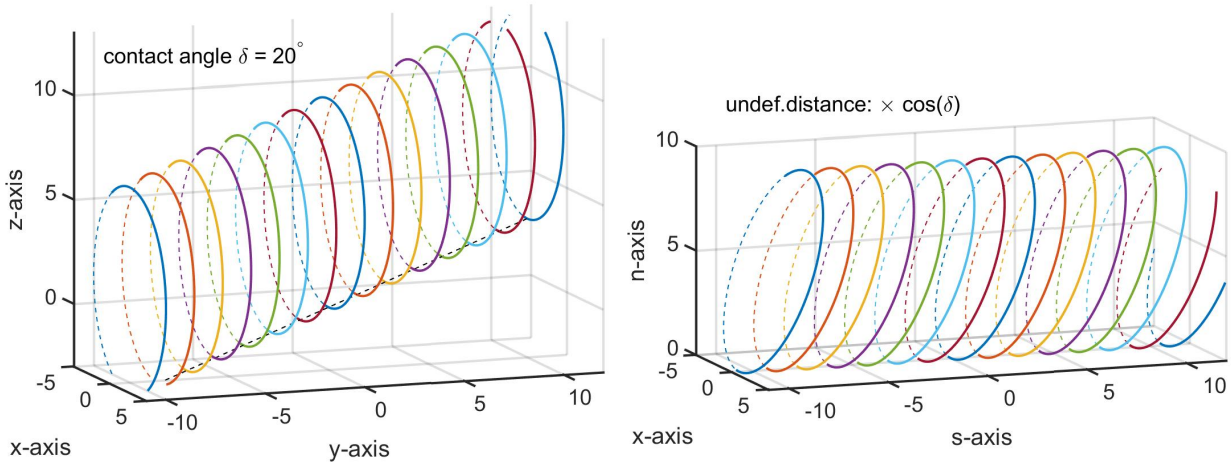



Figure 4.10: *Left: vertical sections of a wheel in global Oxyz-coordinates, circles of almost the same radius $R_w(y)$. Right: rotated to contact local Oxsn-coordinates, all circles touching at $n = 0$. The effective radius becomes $R_{x,eff} = R_w(y)/\cos(\delta)$ [55].*

- `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 4.1) and either the approach `PEN` or the normal force `FN`, see Section 4.5.

The pressure distribution may be computed from the geometry (`B` $= 0$), or can be prescribed beforehand.

$$\text{elliptical, B} = 2 \quad : \quad p_n(x,y) = p_{max}\sqrt{1-(x/a)^2-(y/b)^2}, \quad p_{max} = \frac{3F_n}{2\pi ab}, \tag{4.20}$$

$$\text{parabolical, B} = 3 \quad : \quad p_n(x,y) = p_{max}\left(1-(x/a)^2-(y/b)^2\right), \quad p_{max} = \frac{2F_n}{\pi ab}. \tag{4.21}$$

The potential contact area is taken as the rectangle $[-\texttt{AA}\cdot s, \texttt{AA}\cdot s]\times[-\texttt{BB}\cdot s, \texttt{BB}\cdot s]$ with $s =$ `SCALE`.

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

Choose `SCALE` $= 1.1$ for a potential contact area that is 10% larger than the actual Hertzian contact area, or `SCALE` $=$ `MX`$/($`MX` $- 4)$ to add two unused rows and columns around the actual contact ellipse.

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 setting `SCALE` $> 1$ in those cases.

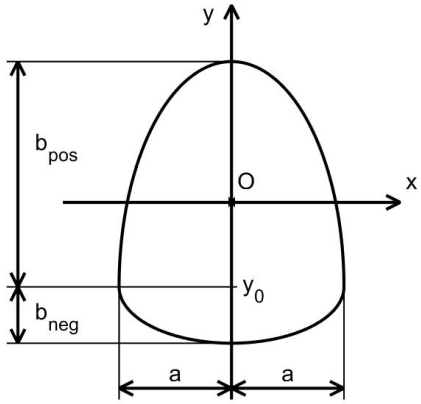



Figure 4.11: *Definitions for the double half-elliptical contact region of the SDEC approach.*

### 4.3.2 Rectangular contacts – 2D Hertzian geometries

Finite line contacts are defined by the assumption that the pressure distribution is uniform on $y \in [-b, b]$, conforming with the 2D Hertzian solution, and zero outside this range.

- `IPOTCN` = −4: rectangular contact specified by curvature `A1` and half width `BB`;
- `IPOTCN` = −5: rectangular contact with half length `AA` and half width `BB`.

The pressure distribution is then prescribed using options `B` = 2 or 3, using Hertzian formulas in each strip $y = const$:

$$\text{elliptical, B = 2} \quad : \quad p_n(x,y) = p_{max}\sqrt{1-(x/a)^2}, \quad p_{max} = \frac{2F_n}{\pi a L}, \quad L = 2b, \tag{4.22}$$

$$\text{parabolical, B = 3} \quad : \quad p_n(x,y) = p_{max}\left(1-(x/a)^2\right), \quad p_{max} = \frac{3F_n}{4aL}. \tag{4.23}$$

These options require that the total force be prescribed (`N` = 1, page 22).

### 4.3.3 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 [34].

- `IPOTCN` = −6: SDEC approach, using `AA`, `BNEG` and `BPOS`.

The dimensions regarding the two half ellipses are illustrated in Figure 4.11. Note that the origin of the contact coordinates is chosen differently than in [34], as discussed in [54].

| | | |
|---|---|---|
| `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. |

The pressure distribution is then prescribed using option `B` = 4, using a Hertzian-like formula in each strip $y = const$:

$$b = \frac{b_{pos} + b_{neg}}{2}, \quad \psi = \frac{b_{pos} - b_{neg}}{2b}, \quad y_0 = -b\psi, \tag{4.24}$$

$$y - y_0 > 0 : x_{l,pos} = a\sqrt{1 - \frac{(y - y_0)^2}{b^2(1+\psi)^2}}, \tag{4.25}$$

$$y - y_0 < 0 : x_{l,neg} = a\sqrt{1 - \frac{(y - y_0)^2}{b^2(1-\psi)^2}}, \tag{4.26}$$

$$p_{max} = \frac{3}{2}\frac{F_n}{\pi ab}, \quad p_n(x, y) = \frac{p_{max}}{a}\sqrt{x_l^2(y) - x^2}. \tag{4.27}$$

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 4.1), the normal force `FN`, see Section 4.5, and the `SCALE` parameter of the Hertzian approach (Section 4.3.1).

### 4.3.4 Direct specification of the potential contact area

The parameters related to the potential contact area and its discretisation are shown in Figure 4.9:

| | | |
|---|---|---|
| `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 `IPOTCN`.

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

## 4.4 Non-Hertzian geometry specification

In Hertzian problems the geometry is specified by the loading parameters and curvatures, see Section 4.3.1. 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:

$$h(x, y) = b_1\, x^2 + b_2\, x\, y + b_3\, y^2 + b_4\, x + b_5\, y + b_6 \tag{4.28}$$

| | | |
|---|---|---|
| `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

$$h(x, y) = p(y) + \frac{(x - x_m)^2}{2\, R_m} \tag{4.29}$$

| | | |
|---|---|---|
| `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 $y_k$, $k = 1 \cdots$ `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 $\delta$, the effective radius of curvature is $R/\cos\delta$, see Section 4.3.1 (Figure 4.10) and [17, eq. (1.61b)].

The formula used with IBASE = 3 is the following.

$$h(x,y) = b_1 \sin(b_2(x-b_3)) - b_4 \sin(b_5(x-b_6)) + x^2/b_7 + y^2/b_8 \tag{4.30}$$

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 (4.17) and (4.18) 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 $\geq 0$. The function $pl$ is specified by six parameters analogously to $h(x,y)$ in (4.28).
- IPLAN = 3: Union of two rectangles, specified by eight parameters.

  $$[x_l^{(1)}, x_h^{(1)}] \times [y_l^{(1)}, y_h^{(1)}] \quad \cup \quad [x_l^{(2)}, x_h^{(2)}] \times [y_l^{(2)}, y_h^{(2)}] \tag{4.31}$$

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

## 4.5 Kinematic variables

The normal problem is specified either using the approach or the total normal force, depending on the $N_3$-digit (page 22).

| | | |
|---|---|---|
| PEN | [mm] | Approach $\delta_n$ 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, $\delta_n$ is the maximum penetration of the undeformed surfaces. |
| FN | [N] | Total normal force in the contact area. |

### 4.5.1 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 | [mm] | Rigid shift, i.e. the displacement of the two bodies' axles in $x$- and $y$-directions respectively in a time step $\delta t$. |
| CPHI | [rad] | Rotation shift, angular displacement of the two bodies around the $z$-axis in a time step $\delta t$. |

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

| | | |
|---|---|---|
| EXRHS(I,:) | [mm] | 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

$$\mathbf{W}_{It} = [\mathtt{CKSI} - \mathtt{CPHI} \cdot y_I + \mathtt{EXRHS}_{Ix}, \quad \mathtt{CETA} + \mathtt{CPHI} \cdot x_I + \mathtt{EXRHS}_{Iy}]^T \tag{4.32}$$

The true time step size $\delta t$ is unknown to the program. It is set to 1 s, and the corresponding variables are set to CHI = 0°, DQ = 1 mm and VELOC = 1 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.

### 4.5.2 Tangential quantities in case of rolling (T = 2 − 3)

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

| | | |
|---|---|---|
| CHI | [rad] | Rolling direction $\chi$. 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 $\pi$ (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 preferrably be of the order of the grid size DX [73]. |
| 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

$$\mathbf{W}_{It} = \mathtt{DQ} \cdot \begin{bmatrix} \mathtt{CKSI} - \mathtt{CPHI} \cdot (y_I + sn \cdot \delta q/6) + \mathtt{EXRHS}_{Ix} \\ \mathtt{CETA} + \mathtt{CPHI} \cdot (x_I + cs \cdot \delta q/6) + \mathtt{EXRHS}_{Iy} \end{bmatrix} \tag{4.33}$$

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` $\cdot$ `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).

## 4.6 Solution processes

Using CONTACT's full solution approach, 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 alternatingly until the update between consecutive iterations is smaller than a tolerance.

2. The slip velocity iteration.

   (a) The traction bound $g$ 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 [17]. 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 [17].

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

   (a) If the total tangential forces $F_x, F_y$ are prescribed, then the creepages $\xi, \eta$ 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 [50, 74, 45, 48].

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

$$e_I = h_I^* + \sum_{J \in C} A_{InJn}\, p_{Jn} = 0, \quad \text{for } I \in C \tag{4.34}$$

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 [49]. 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 [50]. 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 [74], based on BCCG and using FFTs, or the older and slower ConvexGS solver [45]. For steady state problems the SteadyGS solver is used [48]. 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.

These iterations are mosly bypassed when using the KPEC, ANALYN and FASTSIM approaches.

The nesting of algorithms is changed in the analysis of wheel/rail contact (Chapter 3), adding a new outermost level 0 for wheelset position and velocity variables, and skipping the Newton-Raphson loops at level 4.

These iteration 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 solvers 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

$$\|\mathbf{p}^k - \mathbf{p}^{k-1}\|_{rms} < \texttt{EPS} \cdot \|\mathbf{p}^k\|_{rms} \quad (k\text{: iteration counter}). \tag{4.35}$$

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 2.3).

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 $\omega_h$ for ConvexGS and SteadyGS for elements in the adhesion area. |
| OMEGAS | [−] | Relaxation parameter $\omega_s$ 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 $\omega_{slp}$ 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 [45] and [46]. 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 $\omega_h, \omega_s$;
- Small step sizes $c = \delta q/\delta x \ll 1$ may need reducing $\omega_h, \omega_s$ in order to work well [46].

When using velocity-dependent friction, there may exist multiple valid solutions to a case [6, 63]. The initial estimate for the slip-velocity then determines which one of the possible solutions is found. Three different strategies are provided. When INISLP $< 0$, the slip velocity is approximated from below, such that the 'low-slip' solution is found. When INISLP $> 0$, the slip velocity is approximated from above and the 'high-slip' solution is obtained. Finally the choice INISLP $= 0$ is in between and uses the slip velocity of the previous case as initial estimate.

## 4.7 Overall output quantities per contact patch

Overall output quantities are written to the output file <experim>.out when the O-digit is set to values $1 - 5$.

The global part of the output firstly contains a description of the problem that is solved (O $\geq 2$), 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.

### 4.7.1 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 4.5. 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:

$$\texttt{MZ} = \delta x\, \delta y \sum_I \left( p_{Iy}\, x_I - p_{Ix}\, y_I \right), \tag{4.36}$$

$$\texttt{ELEN} = \frac{\delta x\, \delta y}{2 \cdot 1000} \sum_{i \in \{x,y,n\}} \sum_I p_{Ii}\, u_{Ii}, \quad \texttt{FRIC} = \frac{V}{1000} \cdot \delta x\, \delta y \sum_{\alpha \in \{x,y\}} \sum_I p_{I\alpha} s_{I\alpha} \tag{4.37}$$

Note that $s_{I\alpha}$ is the relative slip velocity, hence the multiplication with $V$. Note: the elastic energy is not computed when 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:

$$\texttt{FRIC} = V \cdot 10^{-3} \cdot \left( \texttt{CKSI} \cdot F_x + \texttt{CETA} \cdot F_y + \texttt{CPHI} \cdot M_z \right), \tag{4.38}$$

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 (4.2)), and when the creepages are constant throughout the contact patch ($\texttt{E}_3 = 0$). This is illustrated for instance in [30].

The sensitivities are calculated when the Newton-Raphson loop is used for presribed total forces. This is currently only the case when F = 1 or 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 4.5). |

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

### 4.7.2 Output in case of shifts (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 | [$\text{mm}^{-1}$] | Sensitivity of relative tangential force FX to a change of rigid shift CKSI. |

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

$$\texttt{FRIC} = \delta x\, \delta y \sum_\alpha \sum_I p_{I\alpha} S_{I\alpha}. \tag{4.39}$$

## 4.8 Main solution arrays

The detailed output of quantities inside the contact area for `O` = 5 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, 3: Plasticity). Refer to (4.17) and (4.18) 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. (4.18). |
| `H-PEN` | [mm] | True undeformed distance $h(\mathbf{x}_I) - \delta_n$, taking into account the approach $\delta_n$. |
| `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 $\mathbf{W}_{It}$. |

## 4.9 Subsurface stresses

The calculation of subsurface stresses is activated by the control digit `S` (Section 2.3) when using the input-file, or subroutine `subs_calculate` in the CONTACT library (Section 7.4). 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 $\sigma_{hyd}$, the equivalent tensile stress $\sigma_{vm}$ of the von Mises criterion, 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.

### 4.9.1 Control digits for the subsurface stress calculation

The input of the subsurface points in `<experim>.inp` starts with the control digits $\mathtt{A}_s$ and $\mathtt{O}_s$, governing the level of output of subsurface stresses to the Matlab-file `<experim>.<case>.subs` and the to the `out`-file.

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

- **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.

**$\mathtt{O}_s$ - 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 $\mathtt{S} = 2$ or $\mathtt{S} = 3$. The same value is used for all blocks of subsurface points.

### 4.9.2 Input for the subsurface stress calculation

The locations $[x, y, z]^T$ for the subsurface stress calculation are specified in 'blocks' of $\texttt{NX} \cdot \texttt{NY} \cdot \texttt{NZ}$ 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 ($\texttt{C}_3 = 2$, page 18) [24, 72]. This can be resolved using bilinear loading elements, setting $\texttt{C}_3 = 3$, at the expense of longer computations. A practical alternative is to sample the subsurface stresses at the element centers only [72]. 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:

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

The coordinates per block can be specified in different ways.

| | | |
|---|---|---|
| `ISUBS` | – | 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.

  | | | |
  |---|---|---|
  | `NZ` | – | Number of layers $n_z$ in vertical direction. |
  | `ZL` | [mm] | $z$-value $z_l$ of the lowest layer in vertical direction. |
  | `DZ` | [mm] | Step size $\delta z$ between layers in vertical direction. |

  The values $z_{i_z}, i_z = 1, \ldots, n_z$ 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.

  | | | |
  |---|---|---|
  | `IXL` | – | Starting column number $i_{x,l}$ for the selection. |
  | `IXINC` | – | Column number increment $i_{x,inc}$ for the selection. |
  | `IXH` | – | 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 $i_{x,l} \le i_x \le i_{x,h}$. 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) | [mm] | 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.

### 4.9.3 Output of subsurface stress calculation

The output of the subsurface stress calculation is written to the file <experim>.<ncase>.subs (Section A.7), which can be imported directly in Matlab using the script loadstrs.m (Section 6.3). Maximum values per block are written to the file <experim>.out according to the $O_s$-digit, and for $O_s \ge 4$ 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 [16], which is reproduced in Appendix C of [17].

| | | |
|---|---|---|
| `XTABL(k,3)` | [mm] | Expanded list of coordinates of points where subsurface stresses are calculated. Each point `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 `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 $\boldsymbol{\sigma}(\mathbf{x})$:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}. \tag{4.40}$$

| | | |
|---|---|---|
| `SIGMA(k,i,j)` | [N/mm$^2$] | Stresses $\sigma_{ij}(\mathbf{x}_k)$ at subsurface point `k`. |

The components $\sigma_{ij}$ usually aren't 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 $\sigma_{hyd}$, one-third of the first stress invariant $I_1$:

$$I_1 = trace(\boldsymbol{\sigma}) = \sigma_{xx} + \sigma_{yy} + \sigma_{zz}, \quad \sigma_{hyd} = \frac{1}{3} I_1. \tag{4.41}$$

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

| | | |
|---|---|---|
| `SIGHYD(k)` | [N/mm$^2$] | Mean hydrostatic stress $\sigma_{hyd}$. |

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

$$\mathbf{s} = \boldsymbol{\sigma} - \sigma_{hyd}\,\mathbf{I}, \quad \text{with } \mathbf{I} \text{ the } 3 \times 3 \text{ identity.} \tag{4.42}$$

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

| | | |
|---|---|---|
| `SIGVM(k)` | [N/mm$^2$] | The 'equivalent (tensile) stress' or 'von Mises stress' $\sigma_{vm}$ of the von Mises criterion. |

These values are computed as

$$J_2 = \frac{1}{2}\sum_{ij}(s_{ij})^2 = \frac{1}{6}\left((\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right), \tag{4.43}$$

$$\sigma_{vm} = \sqrt{3J_2}. \tag{4.44}$$

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

$$J_2 \geq k_{yield}^2 \quad \rightarrow \quad \sigma_{vm} \geq \sqrt{3}\, k_{yield}, \tag{4.45}$$

with $k_{yield}$ the yield strength of the material in simple shear, or as

$$\sigma_{vm} \geq \sigma_{yield}, \tag{4.46}$$

with $\sigma_{yield}$ the yield strength in simple tension.

Finally CONTACT provides the principal stresses[3] 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

$$\sigma_{tresca} \geq 2k_{yield} \quad \text{or} \quad \sigma_{tresca} \geq \sigma_{yield}. \tag{4.47}$$

Again, $k_{yield}$ is the yield strength in simple shear, and $\sigma_{yield}$ the yield strength in simple tension.

[3] This calculation uses the function `dsyevc3` provided by Kopp under the LGPL license [22].

# Chapter 5

# 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. The chapter starts with classical test-cases like those of Cattaneo, Carter, Bentall-Johnson, etc, in Sections 5.1–5.6. Further examples are given in Sections 5.7–5.13 that are targeted on wheel/rail contacts: non-Hertzian geometries, creep curves, including the effects of interfacial layers and surface temperature.

## 5.1 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, i.e. `IPOTCN` $< 0$. The radius of the sphere is 50 mm such that the curvatures at the contact point are $A = B = 0.01\ \mathrm{mm}^{-1}$ (see equation (4.19)). The sphere and plane are assumed to consist of the same (soft) polyethylene material, with $G = 200\,\mathrm{N/mm^2}$ 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

$$a = b = \left(\frac{3}{2} F_n \frac{1-\nu^2}{E} \frac{1}{A+B}\right)^{1/3}. \tag{5.1}$$

This gives $F_n = 9.1954\,\mathrm{N}$ and for the maximum pressure $p_{n,max} = 3F_n/2\pi ab = 4.3905\,\mathrm{N/mm^2}$. 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 $15 \times 15$ elements, two additional elements are used on all sides by setting `SCALE` $= 19/15 = 1.267$.

```
 200100     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
0000331   H-G-I-A-O-W-R  HEAT, 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`:

```
 3D 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
 0101541   H-G-I-A-O-W-R  HEAT, 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 2.3) are

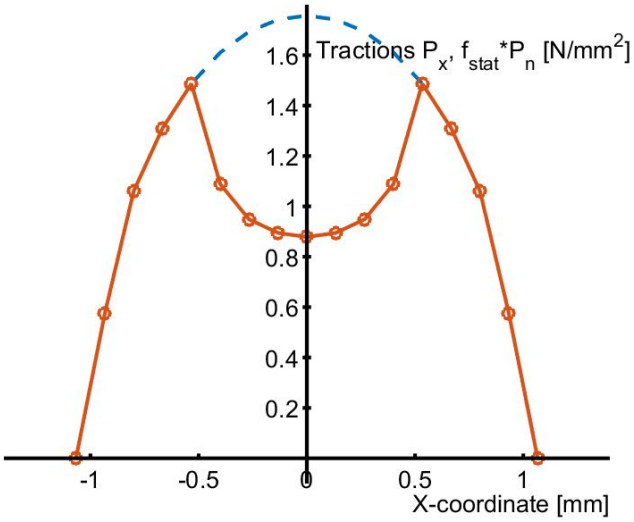



Figure 5.1: *Results for the Cattaneo shift problem. Distribution of tangential traction $p_x$ along the center-line $y = 0$.*

- the `P`-digit, which describes that the sphere was free from tangential tractions initially;
- 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 $(1-\theta^3)\,\mu F_n$, with $\theta = 1/2$. According to Cattaneo this yields a circular adhesion area with radius $(1-\theta)$ concentric with the contact area [17, sec. 5.2.1.1]. The corresponding shift of the sphere with respect to the plane is given by

$$W_x = (1-\theta^2)\,\frac{3\mu F_n}{2\pi G}\left(\frac{\pi}{2} - \nu\frac{\pi}{4}\right) \tag{5.2}$$

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 $\mu F_n$, hence $F_x = -7/8$. The shift is $W_x = 0.00817$ mm, and the theoretical tangential traction is

$$\begin{aligned} p_x &= -\mu p_{n,max}\left(\sqrt{1-r^2} - \tfrac{1}{2}\sqrt{1-4r^2}\right), && \text{when } 0 \le r \le \tfrac{1}{2} \\ &= -\mu p_{n,max}\sqrt{1-r^2} && \text{when } \tfrac{1}{2} \le r \le 1 \\ &= 0 && \text{when } r \ge 1. \end{aligned} \tag{5.3}$$

This theoretical solution by Cattaneo is not altogether correct; there is a traction component $p_y$ orthogonal to $x$-axis that is ignored. This component is included in the full solution computed by CONTACT.

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     PMAX
4.598E-02   8.155E-03       0.000   1.998E-02        4.393
```

The results may be inspected further with the Matlab package, see Figure 5.1. The commands for producing this figure are (see Section 6.2):

```
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 $\mu p_n$, with maximum $\mu p_{n,max} = 1.756\,\mathrm{N/mm^2}$.

## 5.2 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 $\delta y$. 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 [11].

We consider a steel cylinder with a radius of $R^{(1)} = 500\,\mathrm{mm}$ and a steel half-space ($R^{(2)} = \infty$, combined radius $R = R^{(1)}$), with elastic constants $G^{(a)} = 82\,000\,\mathrm{N/mm^2}$, $\nu^{(a)} = 0.28$. According to the 2D Hertz solution, the normal force required per unit of width to achieve a contact area $[-a, a]$, with $a = 1\,\mathrm{mm}$ is

$$F_n = \frac{\pi\, a^2\, G}{4\, R\, (1-\nu)} = 178.90\,\mathrm{N/mm} \tag{5.4}$$

This is entered in CONTACT multiplied by the element width $\delta y$ as `FN` = 35 780. 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:

$$\xi = -\mu \frac{a - a'}{R}, \quad F_x = (1 - (a'/a)^2)\, \mu F_n\,. \tag{5.5}$$

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.

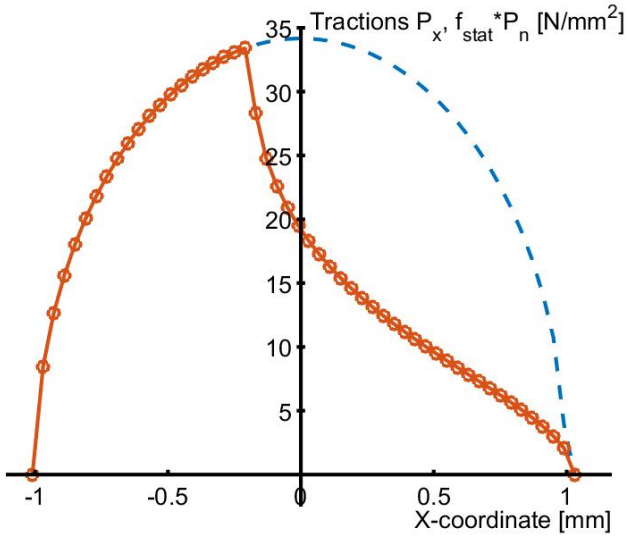



Figure 5.2: *Tangential tractions $p_x$ for the 2D Carter/Fromm problem.*

```
 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
 0001541   H-G-I-A-O-W-R  HEAT, 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                             FSTAT, FKIN
   0.000    0.040    30000.                   CHI, DQ, VELOC
   0.280    0.280    82000.   82000.          POISS 1,2,  GG 1,2
    3                                         IPOTCN
  55   1   -1.0500   0.000    0.0400   200.0   MX,MY,XC1,YC1,DX,DY
   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 5.2.

## 5.3 The 2D Bentall-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 Bentall and Johnson [1] for the contact of aluminum and steel rollers.

An aluminum roller ($G^{(1)} = 23\,000\,\mathrm{N/mm^2}$, $\nu^{(1)} = 0.31$) with radius $R^{(1)} = 50\,\mathrm{mm}$ is pressed onto

a steel roller ($G^{(2)} = 82\,000\,\mathrm{N/mm^2}$, $\nu^{(2)} = 0.28$) of the same size. The normal load is defined using equation (5.4) such that the Hertzian half contact width becomes $a_h = 0.1$ mm. Note that the effective radius is $R = 25$ 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 $R = 25$ mm. In the input this shows up via $b_1 = 1/2R = 0.02\,\mathrm{mm^{-1}}$. Further, the combined material parameters $G, \nu$ and $K$ are required, that are defined according to [17, eq.(1.44)]:

$$\frac{1}{G} = \frac{1}{2}\left(\frac{1}{G^{(1)}} + \frac{1}{G^{(2)}}\right), \quad \frac{\nu}{G} = \frac{1}{2}\left(\frac{\nu^{(1)}}{G^{(1)}} + \frac{\nu^{(2)}}{G^{(2)}}\right), \quad \frac{K}{G} = \frac{1}{4}\left(\frac{1-2\nu^{(1)}}{G^{(1)}} - \frac{1-2\nu^{(2)}}{G^{(2)}}\right). \tag{5.6}$$

This gives $G = 35\,924$, $\nu = 0.3034$, $K = 0.100$, which in turn gives $F_n = 16.20\,\mathrm{N/mm^2}$.

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
0000011   H-G-I-A-O-W-R  HEAT, 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      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 specfying 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 `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, $\chi = 180°$, such that particles enter the contact area at $x = -0.1$ and leave at $x = 0.1$ 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

$$\xi = 0.457\frac{\kappa a}{R}, \quad \text{with} \quad \kappa = 2\beta = \frac{2K}{1-\nu} = 0.2877. \tag{5.7}$$

Here $\beta$ is Dundur's constant [11, 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 $\kappa$ 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.

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 5.3. These results show excellent agreement with the corresponding results presented in [1, Fig.9].

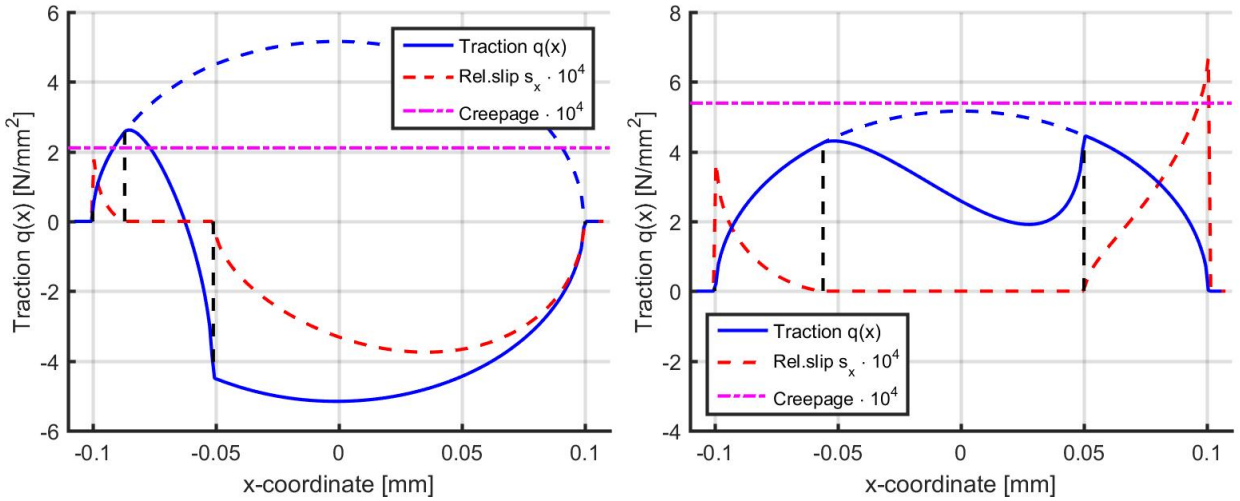



Figure 5.3: *Traction* $q(x) = p_x^{(2)}$ *on the bottom roller in the Bentall-Johnson test-case for two scenarios of tractive rolling, cf. [1, Fig.9]. Left:* $F_x = 0.75$ *on upper roller, right:* $F_x = -0.75$. *In both cases* $\mu = 0.05$.

## 5.4 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 [71]. 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 $\texttt{FG} = f = 1$ (final Young's modulus $E_r = 1600\,\text{MPa}$) and creep relaxation time $\texttt{TC} = \tau_C$ varying between three different test-cases. The corresponding initial modulus of rigidity that is input to CONTACT is (cf. last row of Table 4.1)

$$\texttt{GG} = G_g = E_g/2(1+\nu) = 1143\,\text{MPa}. \tag{5.8}$$

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 $\tau_C \to \infty$), 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 (4.3) and (4.4)). 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 4.1 (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 (5.4)), the half-width of the contact area $a_h$ and the maximum normal pressure $p_0$ are given as

$$a_h = \left(\frac{4RF_n(1-\nu)}{\pi G}\right)^{1/2}, \quad p_0 = \left(\frac{F_n}{\pi R}\frac{G}{(1-\nu)}\right)^{1/2}. \tag{5.9}$$

For $G_g = G_\infty$ (stiff, glassy) and $G_r = G_0$ (soft, rubbery), equations (5.9) 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 $\tau_C = 0$, $0.009, 0.045$ and $2.0\,\text{s}$, and two elastic cases with varying $G$ ($517.5$ and $1143\,\text{N/mm}^2$). 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
 0001341   H-G-I-A-O-W-R  HEAT, 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                                         IPOTCN
  99   1   -7.920   0.000    0.160    1000.0   MX,MY,XC1,YC1,DX,DY
    1    1                                    IBASE, IPLAN
%    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 $\delta y$.
- The potential contact area is defined similarly as in the Carter/Fromm example, Section 5.2. 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 5.4. 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 $\tau_C$ 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 $\tau_C$, 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.

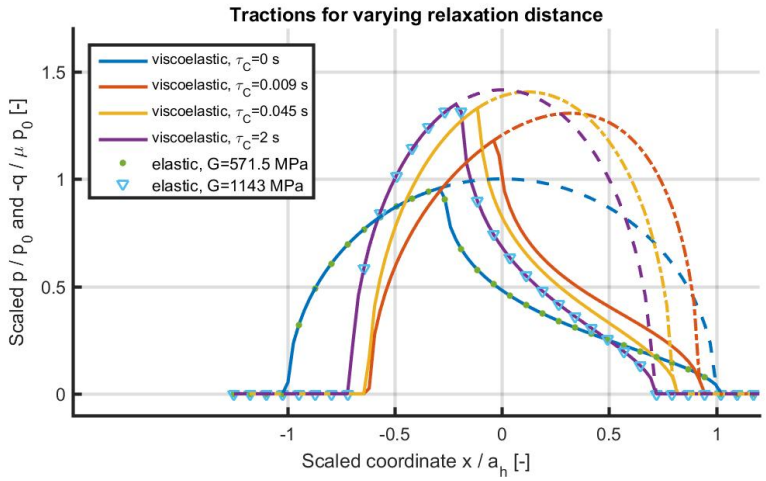



Figure 5.4: *Results for the rolling contact of two viscoelastic cylinders. Distribution of normal and tangential traction $\mu p_n(x)$ and $p_x(x)$ for different relaxation distances $V\tau$. The axes are normalized with $a_h = 6.34$ mm and $p_0 = 60.3$ MPa.*

## 5.5 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 [17, 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 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 [17].

Results of this example are presented in Figure 5.5. 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 [17], 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                                                    IBASE, IPLAN
0.002963    0.000    0.002963    0.000    0.000    0.000    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 $\delta q$ in

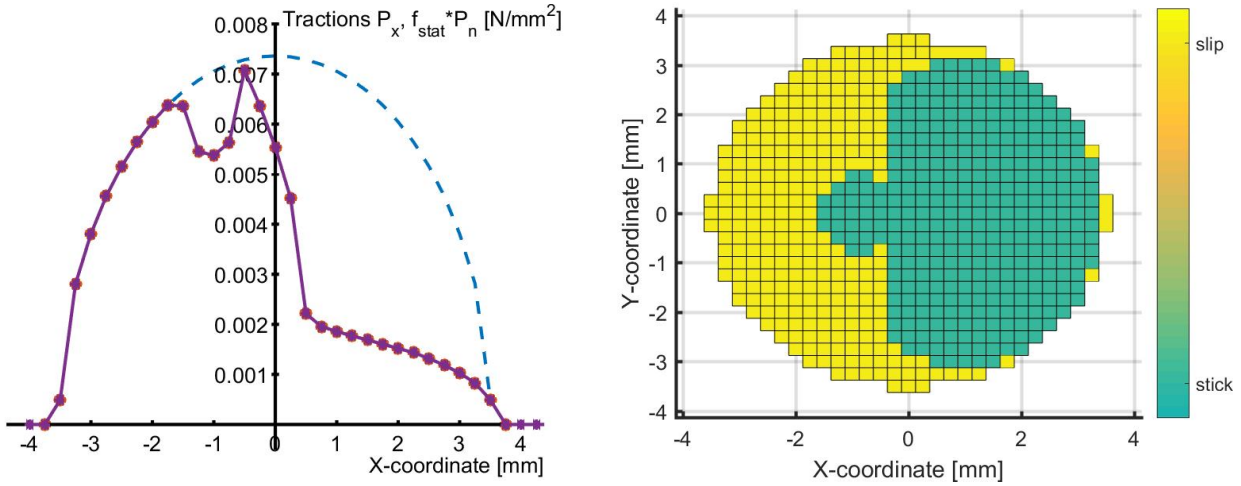



Figure 5.5: *Left: tangential tractions $p_x$ at centerline $y = 0$ for the transient rolling problem 'from Cattaneo to Carter' after '3 units' of time. Right: corresponding element division.*

each step:

$$h(x, y) = b_1 \cdot (x - k \cdot \delta q)^2 + b_3 \cdot y^2 \tag{5.10}$$

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

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

A larger grid is used ($90 \times 33$ instead of $33 \times 33$ elements) in order to accomodate 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 5.5 (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 6.2):

```
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 5.5 (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);
```

## 5.6 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 4.9.2):

```
% 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:
   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.

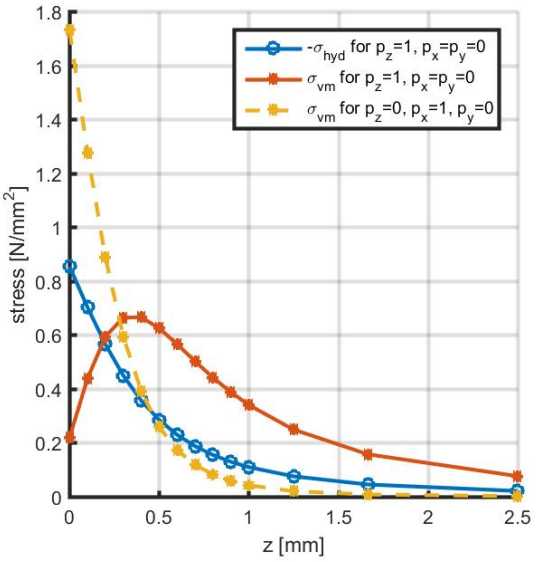



Figure 5.6: *Subsurface stresses in the half-space ($z \geq 0$), due to unit loadings on a square with unit sides in normal and tangential directions.*

The contact problem in this example concerns the experiment described by Kalker in [17, paragraph 5.2.2.4]. It concerns the subsurface field resulting from a unit load in a square element with size $1 \times 1$ 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 5.6. Note that Kalker's $\sigma_{ii}$ is $3\sigma_{hyd}$.

The Matlab commands used for producing this figure are (see Section 6.3):

```
% 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,:)), '--*')
```

Here '`diffstrs`' is used to obtain the stresses due to the tangential traction alone. For more information type '`help diffstrs`' at the Matlab command 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 5.7. 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];
```

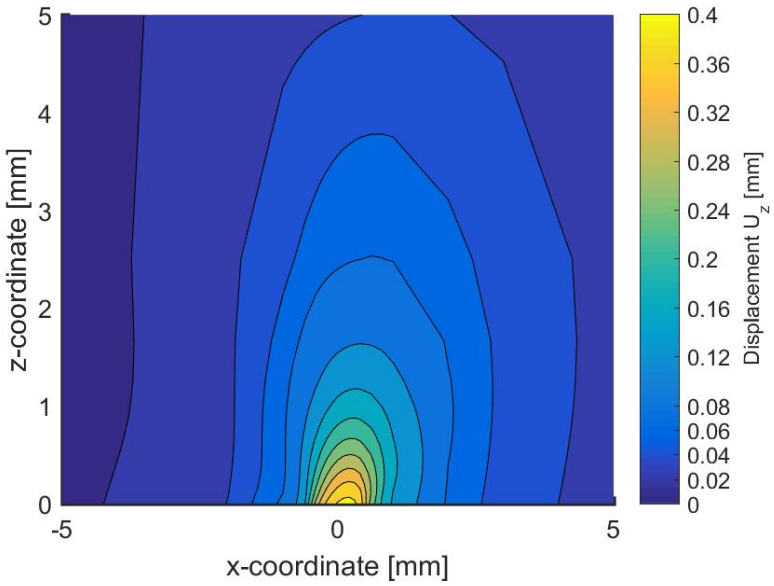



Figure 5.7: *Normal displacements $u_z$ at $y = 0$ inside body 1 ($z \geq 0$), due to unit loads $p_n = 1$ and $p_x = 1$ on a square with unit sides.*

```
plotstrs(s2b, opt);
set(gca,'clim',[0 0.40]);
h=findobj(gcf,'type','colorbar');
set(h, 'ylim',[0 0.40], 'ytick',opt.cntrlvl);
```

The file `matlab_subsurf.m` shows how the same cases are computed using the CONTACT library version (Sections 7.4, 7.5.3).

## 5.7 The Manchester wheel-rail benchmark

An important application area for CONTACT concerns the detailed study of wheel-rail problems. 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 [40] and is presented together with initial results in [41]. Data for the profiles are provided[1] 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 5.8. 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

[1] `www.cmcc.nl/downloads/manch-benchmark.zip`

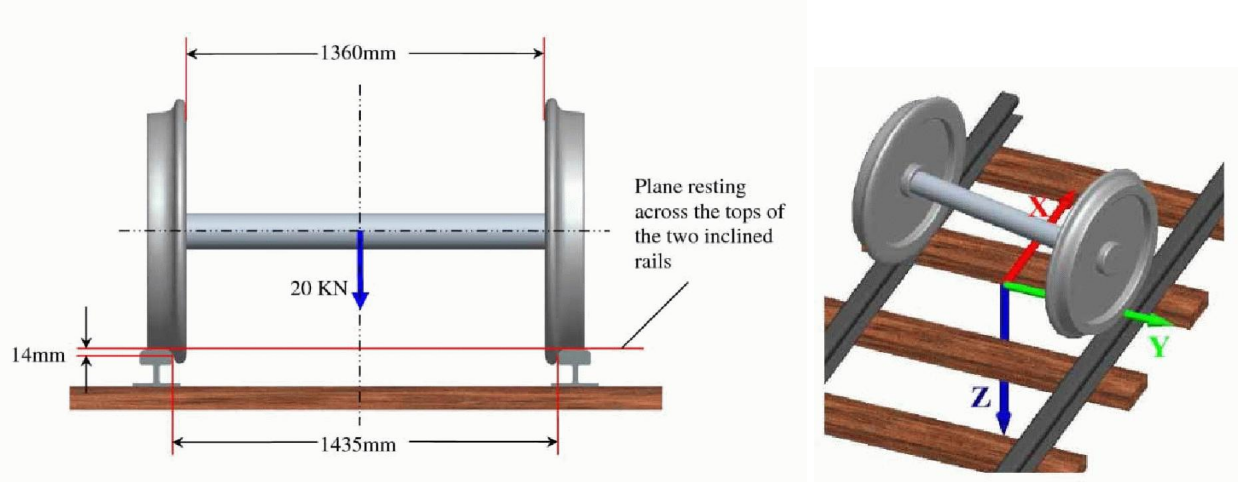



Figure 5.8: *Illustration of the simple wheelset model used in Manchester benchmark simulation Case A [40].*

wheels of the wheelset respectively. Each file defines 21 cases corresponding to benchmark case A-2.2 with steady rolling and including yaw. These problems use a constant coefficient of friction $\mu = 0.30$. The use of friction variation is shown in `mbench_a22_varfric`. This side uses $\mu = 0.20$ on the inside of the rail (surface inclination $\delta_r \leq -20°$), $\mu = 0.30$ at the top of the rail ($\delta_r \geq -10°$), and linear variation in between.

The example uses module 1 for wheel/rail contact analysis (Chapter 3), 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 $\mathtt{N}_1 = 1$ with $\mathtt{FZ} = 10\,\mathrm{kN}$. The primary unknown in this case is the roll angle $\phi_{ws}$: 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 $\phi_{ws}$ 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 $V = 2\,\mathrm{m/s}$. Here, the difficulty is to choose the wheelset angular velocity $\omega_{ws}$ in an appropriate way. In the end, at steady rolling, there should be no resulting moment in rolling direction, i.e. $M^{lft}_{y(ws)} + M^{rgt}_{y(ws)} = 0$. The procedure we used consisted of finding two pitch velocities $\omega_{ws}$ that give opposite net moments and then use iteration to shrink this interval to the desired resolution. The full moments were used as obtained from the CONTACT library version, including the effect of the longitudinal shift of the vertical force.

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

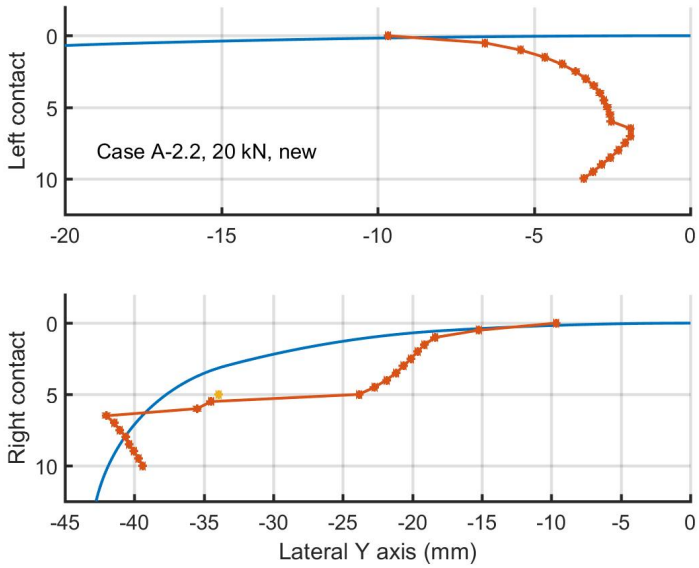



Figure 5.9: *Results for Manchester benchmark example: contact locations at left and right wheels, as function of the lateral wheelset displacement.*

```
% 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
 0122033    V-L-D-C-M-Z-E   VARFRC, FRCLAW, DISCNS, INFLCF, MATER,  ZTRACK, EWHEEL
 0101321    H-G-I-A-O-W-R     HEAT, GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW,   RETURN
  0.280   0.280   82000.  82000.                 POISS 1,2,  GG 1,2
  0.200   0.200   1.000   90d   8.0   4.0       DX, DS, DQREL, A_SEP,D_SEP,D_COMB
   14.0     0    1435.0   0.000                 GAUGHT, GAUGSQ, GAUGWD, CANT
  'MBench_UIC60_v3.prr'   0     1.0   0.0       RFNAME, MIRRORY, SCALE, SMOOTH
     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, NOMRADW
  'MBench_S1002_v3.prw'   0     1.0   0.0       WFNAME, MIRRORY, SCALE, SMOOTH
  0.0  6.0  10000. -0.0006272  0.0144  0.0      S,  Y,  FZ, ROLL,  YAW,  PITCH
 2000. 0.0    0.0   0.0        0.0    -4.329372 VS, VY, VZ, VROLL, VYAW, VPITCH
```

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

One interesting result for this example concerns the contact locations that are found at the different wheelset positions. These are shown in Figure 5.9, 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

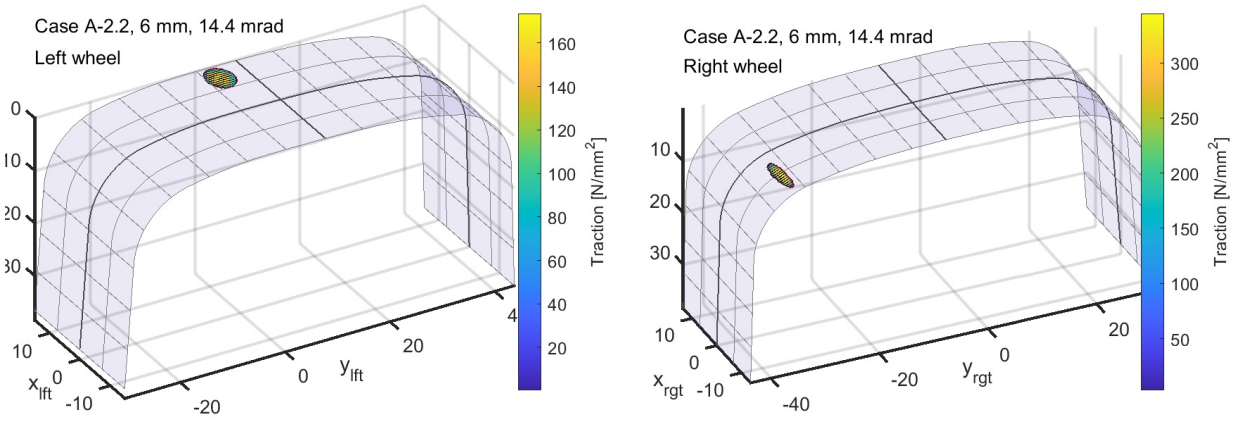



Figure 5.10: *Results for Manchester benchmark example: wheelset with lateral displacement* $y_{ws}$ = 6 mm*, yawed at* $\psi_{ws}$ = 14.4 mrad.

are found on the right rail if the wheelset is displaced to $y_{ws} = 5$ mm. The results are generally in line with those presented in [41], showing that the yaw angle has little effect on the lateral contact position.

The results are illustrated further in Figure 5.10, for a lateral displacement of 6 mm, displaying the contact patches at the rail surface. These pictures are created as follows:

```
l13 = loadcase('mbench_a22_left',13);
prr = read_profile('MBench_UIC60_v3.prr');
opt = plot3d; opt.rw_surfc = 'prr';
plot3d(l13, opt, prr);
```

The file `matlab_mbench.m` shows how this problem is computed using the CONTACT library version in Matlab (Section 7.5.2). The file `test_mbench.f90` shows a corresponding implementation in Fortran.

## 5.8 Simulation of wheel out-of-roundness

CONTACT typically represent wheels using a 2-dimensional profile for a cross-section in lateral direction. A 3-dimensional surface is imagined using a uniform revolution about the axis of the wheel, thereby exluding wheel out-of-roundness. Out-of-roundness can be included using the so-called slices-file, using multiple 2-dimensional profiles at different positions along the circumference of the wheel. This is illustrated in the example `wheelflat.inp`. The example is based on a damaged wheel for which geometry measurements were obtained by means of 3D laser scans [25, 26]. The data are provided thanks to prof. Nielsen of Chalmers University of Technology.

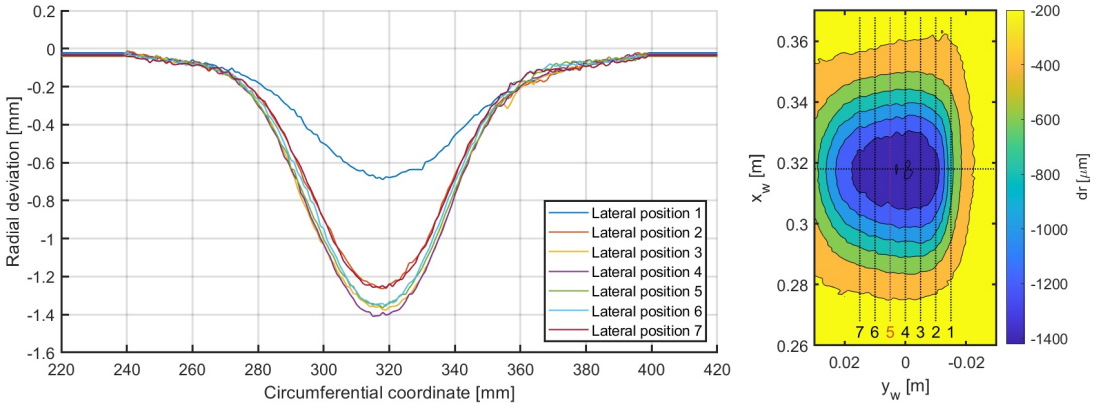



Figure 5.11: *3D laser scan data for a wheel flat used in the* `wheelflat` *example [26, 28].*

The laser scan data were provided as radial deviations $dr_{dev}$ with respect to the nominal profile as illustrated in Figure 5.11. An S1002 profile is used, resampling the tread region at 1 mm spacing, and adding the deviations to the profile data. 3075 samples were given along the circumference of the wheel, corresponding to 1 mm spacing at the tape circle line ($r_{nom} = 490$ mm). They are thinned out by keeping 80 samples on either side of the maximum depth and using every 30th sample along the remainder of the circumference. After this there are 258 slices remaining. Circumferential $x_w$ positions for the slices are converted to radians in the basic interval $[-\pi, \pi)$. This places the flat in the interval $\theta_{wc} = [0.486, 0.812]$ rad $= [27.8°, 46.5°]$ with center at $\theta_{wc} = 0.649$ rad $= 37.2°$.

The slices file `S1002_flat.slcw` used looks as follows:

```
% wheel flat on S1002 profile,
% data courtesy M. Maglio / Chalmers University of Technology

     0.0       1.0   TH_OFFSET [x], TH_SCALE [rad/x], x=rad
     258             NSLC
       0    0    0   NFEAT, NKINK, NACCEL
       2             S_METHOD (1=intpol, 2=approx)

% slice positions TH_SLC [x] and filenames WFNAME per slice

   -3.0995   'S1002_flat/Wheel_section_208.txt'
              ...
   -0.0383   'S1002_flat/Wheel_section_258.txt'
    0.0000   'S1002_flat/Wheel_section_1.txt'
    0.0612   'S1002_flat/Wheel_section_2.txt'
              ...
    0.4857   'S1002_flat/Wheel_section_9.txt'   % start of wheel flat
    0.4878   'S1002_flat/Wheel_section_10.txt'
              ...
    3.1224   'S1002_flat/Wheel_section_207.txt'
```

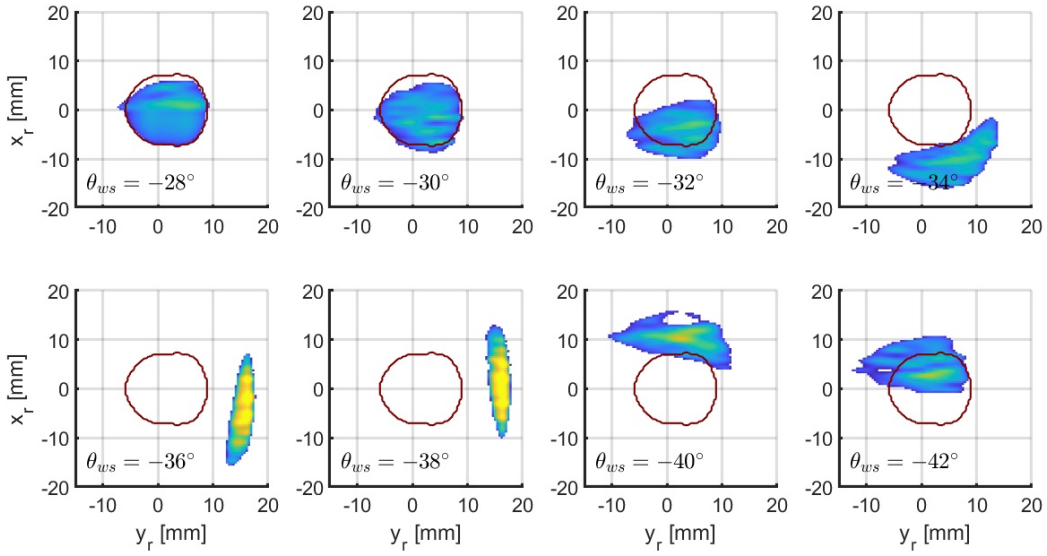



Figure 5.12: *Computed contact patch shapes and pressures $p_n$ for Chalmers' measured wheel flat approaching, passing through and leaving contact at constant vertical force. Contour: corresponding contact patch on nominal wheel profile without out-of-roundness.*

Three values highlighted in blue are $\theta_{scale} = 1$ indicating that the slice positions are given in radians such that no scaling is needed, $n_{slc} = 258$ for the number of slice positions presented, and `s_method` = 2 to select the approximating spline method introducing a little smoothing. In the example, the slice filenames are obtained from the numbering after selecting every 30th slice, before wrap-around of $[\pi, 2\pi)$ to $[-\pi, 0)$. From this we see that the numbers in the filenames are ignored by CONTACT. In fact, it would be possible to repeat the same file ('`Nominal_profile.txt`') at different positions (all $\theta_{slc}$ outside $[0.485, 0.813]$). The slices before number 6 ($\theta_{slc} = 0.3061$) and after number 172 ($\theta_{slc} = 0.9796$) could be deleted with no difference to the calculation.

The calculations in `wheelflat.inp` concern this wheel at 26 pitch positions $\theta_{ws}$. Negative values $\theta_{ws}$ are needed to bring a flat into contact that is located at positive $\theta_{wc}$ (see Figure 3.7). The pitch velocity $v_\theta$ is negative as well. So we find the flat approaching the contact zone at $\theta_{ws} = -25°$ and leaving the contact zone at $\theta_{ws} = -50°$. Dynamic effects are excluded here using a constant prescribed vertical force $F_z = 125\,\text{kN}$. These would need to be computed by multibody dynamic simulation including the track stiffness and wheelset dynamic response.

```
'S1002_flat.slcw'   0   1.0   5.0                WFNAME, MIRRORY, SCALE, SMOOTH
   0.0   0.0  125000.  0.0   0.0   -25.0d       S, Y, FZ, ROLL, YAW, PITCH
2000.0   0.0    0.0   0.0   0.0   -4.08191     VS,VY,VZ, VROLL,VYAW,VPITCH
```

Results are shown in Figure 5.12 for eight wheelset pitch angles $\theta_{ws}$. The dark-red contour shows the outline of the contact patch for the nominal profile. From this we see how the contact patch shifts back at first, moves sideways and then advances ahead of the nominal patch, after which it finally comes back to the reference shape and position.

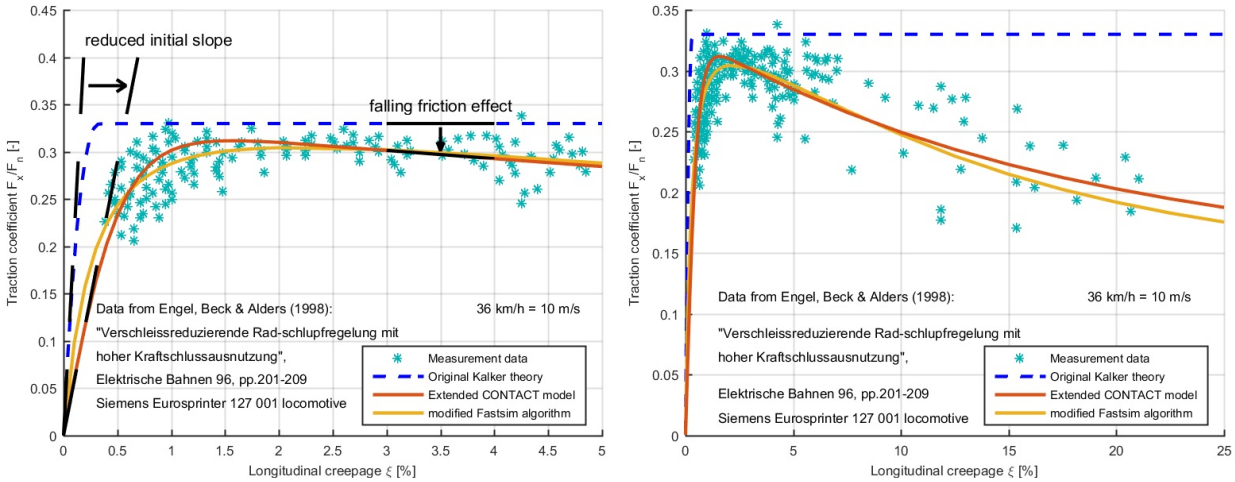



Figure 5.13: *Measured and computed creep forces for the Siemens locomotive Eurosprinter 127001 for pure longitudinal creepage.*

## 5.9 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 Eurosprinter 127 001. Measurements of the creep force were presented by Engel et al. for a study of traction control strategies [7]. The measurements concern straight running on tangent track, i.e. with pure longitudinal creep. The measurements are shown in Figure 5.13 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 $\xi$ 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
0000341   H-G-I-A-O-W-R  HEAT, GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW, RETURN
   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
   80   200    5    1      1e-6    MAXGS , MAXIN , MAXNR , MAXOUT, EPS
```

Noteworthy points are as follows:

- The wheel radius is $R = 625\,\text{mm}$, which gives a curvature $A = 0.0008\,\text{mm}^{-1}$ in rolling direction.
- The Hertzian input option 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 $= 0$, $\mu = 0.33$.

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
 0100341   H-G-I-A-O-W-R  HEAT, GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW, RETURN
 106700.      0.0004     0.000      0.000           FN, CKSI, CETA, CPHI
```

In the $31^{st}$ 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 5.13 (left). These effects are incorporated via the interfacial layer of material model M $= 4$ (Section 4.1.5) and the velocity dependent friction of L $= 4$ (Section 4.2). 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                              MEMDST, MEM_S0
...
% 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 [43]. The friction parameters are derived with the help of equation (4.12) 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       -247.0        0.000        0.000        3.265  -4.124E-05
FN/G       FX/FSTAT/FN  FY/FSTAT/FN  APPROACH      PMAX
    1.301  -7.015E-03        0.000  7.651E-02        640.7
```

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

```
sol  = parse_out3('tractcurv.out');
cksi = 100 * reshape(sol.creep.cksi, 30, 4);
fx   =       reshape(sol.force.fx, 30, 4);
fx   = -fx * diag(fstat);
plot(cksi, fx, '-o');
```

Note that the conventions used in CONTACT make that a positive creepage $\xi$ 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 $\xi$ 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.

The file `matlab_tractcurv.m` shows how this problem is computed using the CONTACT library version (Section 7.5.1).

## 5.10 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 5.14 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 [55], and focuses on the different aspects related to solving conformal contact problems. Three different aspects are taken into account:

**undeformed distance** The undeformed distance between wheel and rail is computed in a different way [64]. Approximating a circular arc by a quadratic function is no longer appropriate, and the changing normal direction is taken into account.

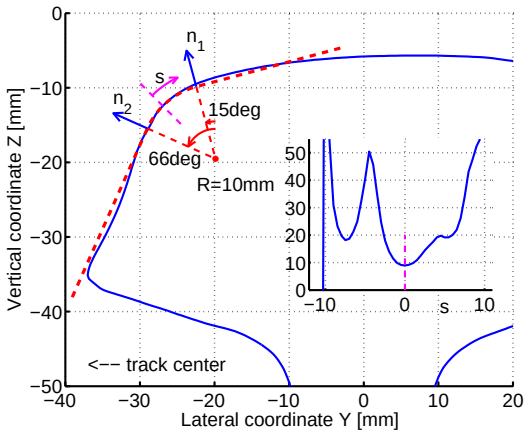



Figure 5.14: *Measured worn profile with conformal contact situation at the rail gauge corner.*

**varying spin creepage** The rigid slip originating from spin cannot be represented anymore by $\mathbf{w} = [\phi y, -\phi x]^T$ (equation (4.33)), but must take the varying normal direction into account [23].

**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 ([64], see Section 4.1.6, Figure 4.4).

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 5.14). This defines the spin creepage $\phi = -0.001537\,\text{rad/mm}$ for a planar contact analysis. The longitudinal and lateral creepages $\xi$ and $\eta$ can take any values in principle, depending on the angle of attack (yaw angle $\psi$) 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 (4.28), with the coefficients computed using (4.19).

2. In the second case, the true undeformed distance is computed in Matlab and entered into CONTACT via option `IBASE` = 9. The approach $\delta_n$ is used already in Matlab such that $\delta_n = 0$ is entered in the input-file.

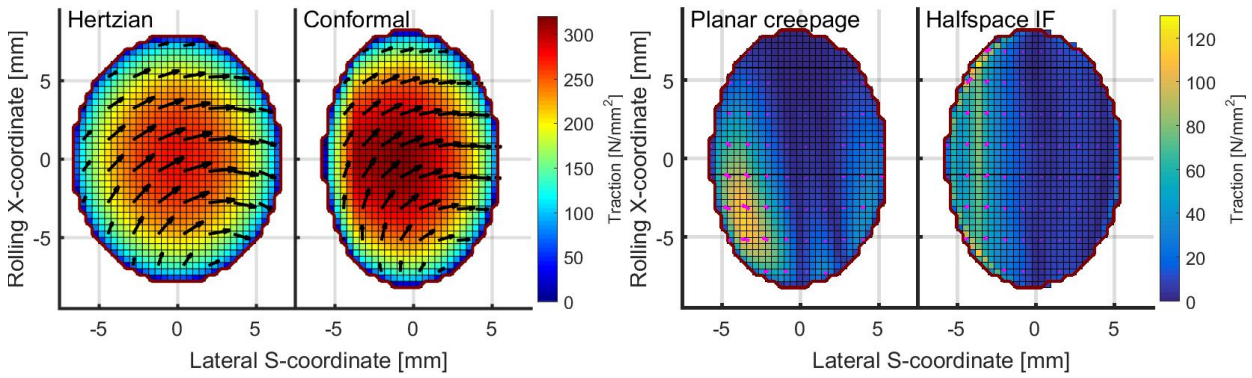



Figure 5.15: *Comparison of results for different approximations to a conformal contact problem. Left: results for case 1, Hertzian approach, and case 5, full conformal approach. Right: contribution of planar → varying creepage (case 3 → 5) and of halfspace → numerical influence functions (4 → 5).*

The third, fourth and fifth cases re-use the grid and undeformed distance of the second case, by setting `D` = 0 and $Z_3 = 0$.

3. In the third case, numerically computed influence coefficients are used. This is entered via $C_3 = 9$ and `CFNAME = 'inflcf_r10_mx51.txt'`. The file is prepared in advance, using finite element calculcations for meshes such as the ones displayed in Figure 4.4 [64]. There is a close correspondence between the CONTACT grid and the input-file: these 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 5.15. 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 5.15 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$^2$]). The colors in Figure 5.15, right, indicate the magnitude of the difference, $\|\mathbf{p}^i - \mathbf{p}^j\|$, showing the length of the arrows on these pictures.

## 5.11 Calculation of contact temperatures

The example `ertz_temperature.inp` shows the calculation of surface temperatures using module 3. This example considers the scenario used by Ertz and Knothe [8]: a Hertzian case with semi-axes $a = 5.88$ and $b = 10.54$ mm, normal force 100 kN, a vehicle velocity of 30 m/s, and a sliding velocity of 1 m/s, which is obtained using creepage $\xi = -0.03333$. Rolling is to the left (`chi` = 180°) such that particles traverse the contact from left to right, with increasing $x$-values corresponding to increasing time.

The temperature calculation is activated setting the `H`-digit to 3. This requires additional material parameters: the specific heat capacity $c_p = 450\,\mathrm{J/kg\,°C}$, thermal conductivity $\lambda = 50 \cdot 10^{-3}\,\mathrm{W/mm\,°C}$, and the density of the materials, $\rho = 7850 \cdot 10^{-9}\,\mathrm{kg/mm^3}$.

For each body, the initial temperature is at the background value, which may be different for the rail (body 1) and wheel (2). The surface temperatures instantaneously jump to an intermediate value for the material that enters the contact area. In the adhesion area this value remains constant, and may be increased in the slip area due to frictional heat input. After leaving the contact area, the heat energy will be spread out inside both bodies, by which the surface temperature relaxes back to the background value again.

Different cases are computed for (1) equal background temperatures with sliding, (2) different background temperatures with no sliding, and (3) different background temperatures with sliding. Finally, the fourth case shows results for partial sliding, using $\xi = 0.2\%$.

We use the non-Hertzian grid specification in order to add extra elements at the trailing side of the contact area. The `A`-digit is set to 2 to export all values in the `mat`-file, for the whole potential contact area.

```
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
3002231  H-G-I-A-O-W-R  HEAT, GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW, RETURN
  999   100    30      1     1e-5     MAXGS , MAXIN , MAXNR , MAXOUT, EPS
 100000.  -0.03333  0.000    0.000          FN, CKSI, CETA, CPHI
  0.300    0.300                           FSTAT, FKIN
  180d     0.200    30.0                   CHI, DQ, VELOC
  0.280    0.280    82000.   82000.        POISS 1,2,  GG 1,2
  0.000    450.     50e3     7.85e-6       BKTEMP1, HEATCP1, LAMBDA1,DENS1
  0.000    450.     50e3     7.85e-6       BKTEMP2, HEATCP2, LAMBDA2,DENS2
   1                                       IPOTCN
```

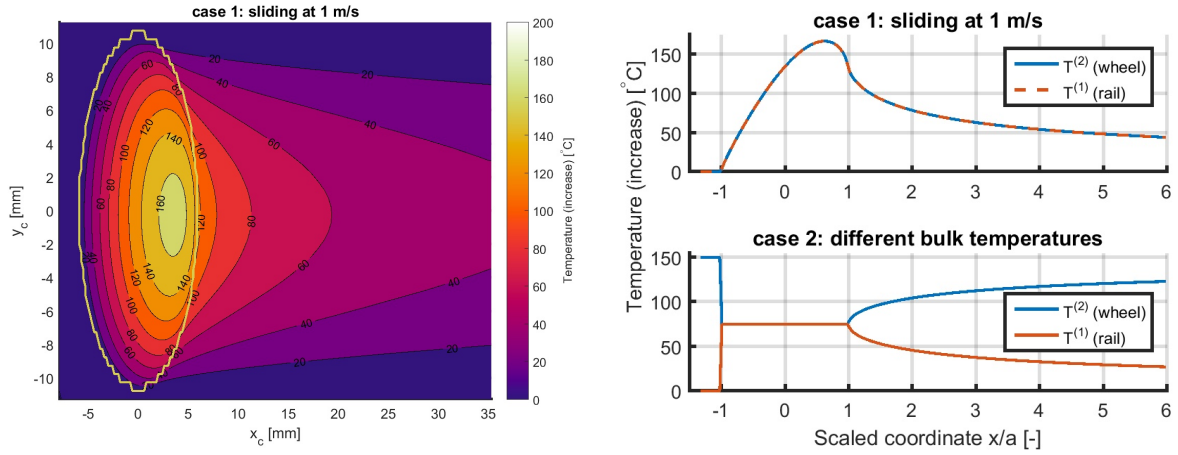



Figure 5.16: *Surface temperatures for example* `ertz_temperature.inp`. *Left & top-right: results for case 1, with full sliding (*1 m/s*) and equal bulk temperatures. Bottom-right: results for case 2, with no sliding and different bulk temperatures (centerline* $y = 0$*).*

```
  211   45   -6.90   -11.25    0.200    0.500     MX,MY, XL,YL, DX,DY
    1    1                                        IBASE, IPLAN
0.001015  0.0  0.0004215  0.0  0.0  0.0          B1,B2,B3,B4,B5,B6
```

The results for cases 1 and 2 are shown in Figure 5.16.

## 5.12 Shearing of interfacial layers

The example `plastic_3bl` illustrates CONTACT's submodel for interfacial layers. This considers the rheometer experiments as reported by Hou et al. [10], where different powders were placed on an anvil, pressed and sheared, measuring the shear stresses as a function of slip distance. Results were presented for magnetite, clay, sand and molybdenum sulfide (MoS2), compressed at 900 N/mm$^2$.

In the simulation, we approximate the measured rheologies by piecewise linear functions. These are characterized by the threshold point $(u_{c0}, \tau_{c0})$ where the slope of the curve changes, and by the slope $k_u$ in the plastic regime. The values used are shown in the table in Figure 5.17, left, the resulting curves in Figure 5.17, right. See also Figure 4.3 in Section 4.1.5.

The simulations concern a single element in contact with size 1 mm$^2$, carrying a normal load $F_n$ = 900 N. The upper body is shifted to the left with respect to the lower, using `T` = 1, with 13–17 steps like `cksi` = −0.035 or −0.050 mm. Additional steps are inserted in the 'clay' series, using one step to the right, `cksi` = +0.036 mm, to show the corresponding behavior. The resulting stresses $\tau = p_x$ are computed by CONTACT as shown in Figure 5.17, right.

In CONTACT, the slope $\tau_{c0}/u_{c0}$ comes about by elastic deformation in the layer and in the primary bodies. The flexibility of the latter is found using a test without interfacial layer: $F_x$ = 84.97 N at

| $3^{rd}$ body layer | $u_{c0}$ $mm$ | $\tau_{c0}$ $N/mm^2$ | $k_u$ $N/mm^3$ |
|---|---|---|---|
| 'magnetite' | 0.070 | 560 | 0. |
| 'clay' | 0.040 | 200 | 400. |
| 'sand' | 0.050 | 400 | −200. |
| 'MoS2' | 0.020 | 20 | 20. |

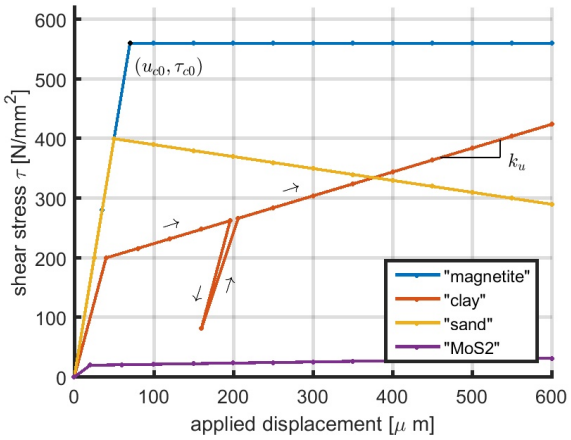



Figure 5.17: *Shear stress curves computed by CONTACT for several compounds, mimicking the results of Hou et al. [10, Fig. 4].*

`cksi` = −0.001 mm. For a given layer thickness, the elastic modulus is then obtained as

$$L_{tot} = L_{prim} + L_{lay} \quad \rightarrow \quad \frac{h^{(3)}}{G^{(3)}} = \frac{u_{c0}}{\tau_{c0}} - \frac{0.001}{84.97} \quad \rightarrow \quad G^{(3)} = h^{(3)} \cdot \left( \frac{u_{c0}}{\tau_{c0}} - \frac{0.001}{84.97} \right)^{-1} \tag{5.11}$$

Using the values of Figure 5.17, left, with a layer $h^{(3)} = 20\,\mu$m, this gives $G^{(3)} = 176.6, 106.3,$ 176.6 and 20.4 N/mm$^2$ for the four materials. The slopes $k_\tau$ used in the input are computed with equation (4.9):

```
% Series 2: Clay,  u_c0 = 0.040 mm, tau_c0 = 200 N/mm2, k_u = 400 N/mm2 / mm

 3  MODULE
  201100     P-B-T-N-F-S        PVTIME, BOUND , TANG  , NORM , FORCE, STRESS
  102400     L-D-C-M-Z-E        FRCLAW, DISCNS, INFLCF, MATER, RZNORM, EXRHS
 0101441   H-G-I-A-O-W-R  HEAT, GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW, RETURN
   900.0      -0.040       0.000      0.000      FN, CKSI, CETA, CPHI
   0.280       0.280       82000.     82000.     POISS 1,2, GG 1,2
   106.25      0.020       200.0      435.0      GG3, LAYTHK, TAU_C0, K_TAU
```

## 5.13 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
 0100341   H-G-I-A-O-W-R  HEAT, 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                                               IPOTCN
   44   44      8.000      4.000      1.100         MX,MY,AA,BB,SCALE
```

Figure 5.18 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 [60]. For a comparison of parabolical and elliptical traction bounds and one versus three flexibilities see [69, 67].

The pictures are created with the Matlab program `plot3d` (Section 6.2), in which the following settings are used:

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

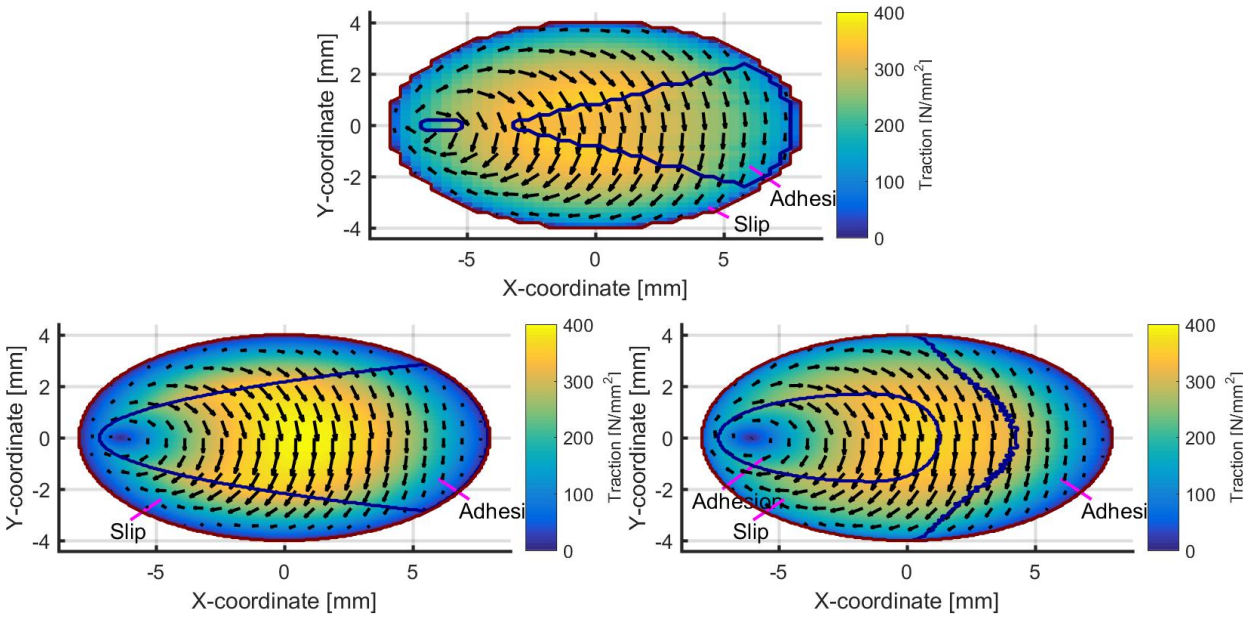



Figure 5.18: *Tangential tractions in a steady state rolling problem with large spin. Top: results for the CONTACT (half-space) algorithm on a* $40\times40$ *grid. Bottom: results for the FASTSIM algorithm with parabolical (B=3, left) and elliptical traction bound (B=2, right), computed on* $200\times200$ *grids.*

# Chapter 6

# 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 5.9 for an example of its use.

## 6.1 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\Vtech CMCC\contact_v23.2\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 ('`<MATLAB>\toolbox\local\pathdef.m`').

You can check whether the path is set correctly by typing `help contact_v23.2\matlab`. This should show the following output:

```
Matlab scripts for visualisation of output of the CONTACT program (trunk).

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_slices     - lower-level routine for reading variable profile slcs file
  read_simpack    - lower-level routine for reading Simpack prr/prw files
  read_miniprof   - lower-level routine for reading Miniprof ban/whl files
  modify_profile  - lower-level routine for making some profile adjustments
  resample_slices - helper routine for 2D interpolation of variable profile
  write_simpack   - routine for writing Simpack prr/prw files
  write_miniprof  - routine for writing Miniprof ban/whl files
```

## 6.2 Inspecting the surface stresses

### 6.2.1 Loading the results into Matlab

The contents of the `.mat`-files (surface tractions) are specified in Section A.6. These are loaded with the script `loadcase`:

```
>> s = loadcase('mbench_a22_left', 13)
s =
       config: 0
      h_digit: 0
         meta: [1x1 struct]
        mater: [1x1 struct]
         fric: [1x1 struct]
       kincns: [1x1 struct]
           mx: 49
           my: 40
           xl: -4.9000
           yl: -3.9000
           dx: 0.2000
           dy: 0.2000
      d_digit: 2
     x_offset: []
     y_offset: []
            x: [1x49 double]
            y: [1x40 double]
        eldiv: [40x49 double]
            h: [40x49 double]
           mu: [40x49 double]
           pn: [40x49 double]
           px: [40x49 double]
           py: [40x49 double]
           un: [40x49 double]
           ux: [40x49 double]
           uy: [40x49 double]
         srel: [40x49 double]
         shft: [40x49 double]
       trcbnd: [40x49 double]
```

With this command the results for the thirteenth case of the Manchester benchmark example are loaded (Section 5.7). The members `meta`, `mater`, `fric` and `kincns` are structs themselves, e.g.

```
>> s.fric
    frclaw: 0
```

```
    fstat: 0.3000
     fkin: 0.3000
```

The result is a structure of which the different components can easily be recognized.

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 an array of structures. For example, two patches are obtained in the `mbench` example when using `dcomb` = `dsep` = 0:

  ```
  >> s11 = loadcase('mbench_a22_right', 11)

  s11 =
         1x2 struct array with fields: ...
  ```

- Output may be retrieved for a single patch by specifying the patch number:

  ```
  >> s11b = loadcase('mbench_a22_right', 11, 2);
  ```

  The second argument '`11`' is the case number, the third argument '2' indicates the requested patch number.

- `loadcase` may be also be instructed to return outputs for all patches in separate structures, using the patch number `-1`:

  ```
  >> [s11a, s11b] = loadcase('mbench_a22_right', 11, -1);
  ```

  This example tries to load output for three patches for case 11 of experiment `mbench_a22_right`. The output structs will be empty if no corresponding `mat`-file is found.

### 6.2.2 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: 'default'
    rw_surfc: 'none'
    exterval: NaN
     typplot: 'surf'
```

```
      view: 'default'
    xrange: []
   xysteps: []
    zrange: []
   ixrange: 'auto'
   iyrange: 'auto'
   numvecx: 15
   numvecy: 15
  xstretch: 1
  vecscale: []
  veccolor: []
  vecwidth: []
  addeldiv: []
  eldivcol: [3x3 double]
  eldivwid: []
  colormap: 'parula'
   addplot: 0
```

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 5.5 (right);
- 'h': the undeformed distance of the two bodies;
- 'mu', 'taucrt': the actual local coefficient of friction and the critical shear stress (yield limit) for the tangential plasticity model;
- 'pn', 'px', 'py': the normal and tangential tractions acting on body (1), i.e. the rail in module 1;
- '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 5.15 and 5.18;
- 'un', 'ux', 'uy': the displacement differences in normal and tangential directions;
- 'uplsx', 'uplsy': the components of the accumulated plastic deformation (if M = 4);
- 'utabs+vec', 'upls+vec': elastic and plastic displacements in tangential directions, magnitude and direction;
- 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;
- `'temp1'`, `'temp2'`: surface temperatures of bodies 1 and 2 (if `H` $\geq 1$).

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 5.10 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' );
```

In case the profiles need additional preparations, like scaling, smoothing, or mirroring, they can be read and processed separately, e.g.:

```
mirror_y = 1; mirror_z = -1; scale_yz = 1000;
prr = read_profile('MBench_UIC60_v4.prr', [], mirror_y, mirror_z, scale_yz);
plot3d( s1a, opt, 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 `opt.exterval=NaN`.

**typplot** Type of plot for contact patches: using a color plot (`'surf'`), filled contour plot (`'contourf'`), or plot wheel/rail contacts in rear view (`'rw_rear'` or side view (`'rw_side'`);

An example of the rear view is shown in Figure 3.10. This is created using

```
s = loadcase('siteB_z117',1);
opt          = plot3d;
opt.typplot  = 'rw_rear';
opt.rw_surfc = 'both';
opt.field    = 'pn';
opt.vecscale = 10/2000; % mm/MPa
plot3d( s, opt, prr, prw );
```

**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]`.

**xrange** The range of the $x$-axis to be displayed in `'surf'` and `rw_side'` plots.

**xysteps** The spacing `dxy` or `[dx dy]` of thin lines on 3d surfaces in `'surf'` plots.

**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 and tractions in rear view/side view plots. Length of the vector in *mm* that will be used for a stress of $1\,\mathrm{N/mm^2}$;

**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, cf. `'eldiv_contour'` (only in 2D plots).

**eldivcol** Set colors of lines around whole contact area and the slip and plasticity areas for `'eldiv_contour'` and `'addeldiv'`. Default: dark blue, dark red, magenta.

**eldivwid** Line width for contours of element division.

**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.

**addplot** Typically `plot3d` clears the figure before creating a new plot. Sometimes it is convenient to turn this off and add to an existing plot.

### 6.2.3 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 ($g = \mu p_n$). 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`, $x$ =const) or rows (`2`, $y$ =const, default);

**pxory** plot tangential tractions `px` (`'x'`, default) or `py` (`'y'`);

**xslc** $x$-coordinate(s) for a column-wise plot (`orie` = 1). Can be an array of values for plotting multiple slices at once. Note: the grid column(s) closest to `xslc` is/are used. Default: 0.0;

**yslc** $y$-coordinate(s) for a row-wise plot (`orie` = 2), see `xslc`;

**negpn** flag for the vertical range to be plotted.

- `1` = show positive traction bound $\mu p_n$;
- `0` = show positive and negative traction bounds;
- `-1` = show negative traction bound $-\mu p_n$.

**facpt** multiplication factor for negating `px`/`py`: `1` for plotting the tangential tractions themselves, `-1` for plotting their negative `-px` or `-py`;

**xlim** $x$-range for a row-wise plot (`orie` = 2). Default: `[xl-dx, xh+dx]`;

**ylim** $y$-range for a column-wise plot (`orie` = 1). Default: `[yl-dy, yh+dy]`;

**plim** vertical range of the plot. The default depends on `negpn`. For `negpn` = 1 it is the range `[0,pmax]` 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 5.1 and 5.2, Figures 5.1 and 5.2.

## 6.3 Inspecting subsurface stresses

### 6.3.1 Loading the results into Matlab

The contents of the `.subs`-files (subsurface stresses) are specified in Section A.7, see further Section 4.9.3 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.

### 6.3.2 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: 'mises'
         dir: 'y'
        xslc: 0
        yslc: 0
```

```
    zslc: 0
  addplot: 0
  typplot: 'contourf'
  cntrlvl: 'auto'
   clabel: 'off'
    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 $\sigma_{hyd}$, see equation (4.41);
- `'sigvm'` or `'mises'`: the von Mises stress $\sigma_{vm}$ (equation (4.44)).
- In case the full stress tensor is exported (`A` = 2):
  - `'sigxx'`, `'sigxy'`, `'sigxz'`, `'sigyy'`, `'sigyz'`, `'sigzz'`: the components $\sigma_{ij}$ of the stress tensor `SIGMA` at the subsurface points, see equation (4.40);
  - `'sigma1'`, `'sigma2'`, `'sigma3'`: the principal stresses $\sigma_1, \sigma_2, \sigma_3$ (page 73);
  - `'sigtr'` or `'tresca'`: the maximum shear stress $\sigma_1 - \sigma_3$ of the Tresca criterion (equation (4.47)).

Note that 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;

**clabel** show numerical values on contour lines (`'on'`) or not (`'off'`);

**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 5.7 of Section 5.6.

# Chapter 7

# 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 7.1. The library is a dll (Windows) or so-file (Linux) that provides a well-defined interface. This interface can be accessed from Matlab and Python as well.

## 7.1 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 stored using containers that are called 'result elements'. CONTACT does not care how these containers are used.

- It's possible to re-use a single result element over and over again for all kinds of contact problems;
- Separate result elements may be defined for different axles, and for the left and right wheels on each axle;
- The contact of one wheel with a switch and a stock rail could be separated into different contact problems, using different containers.

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 patches' 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 patch or contact problem. Therefore practically all subroutines start with the arguments `ire` and `icp`. The code `icp` $= -1$ is used to

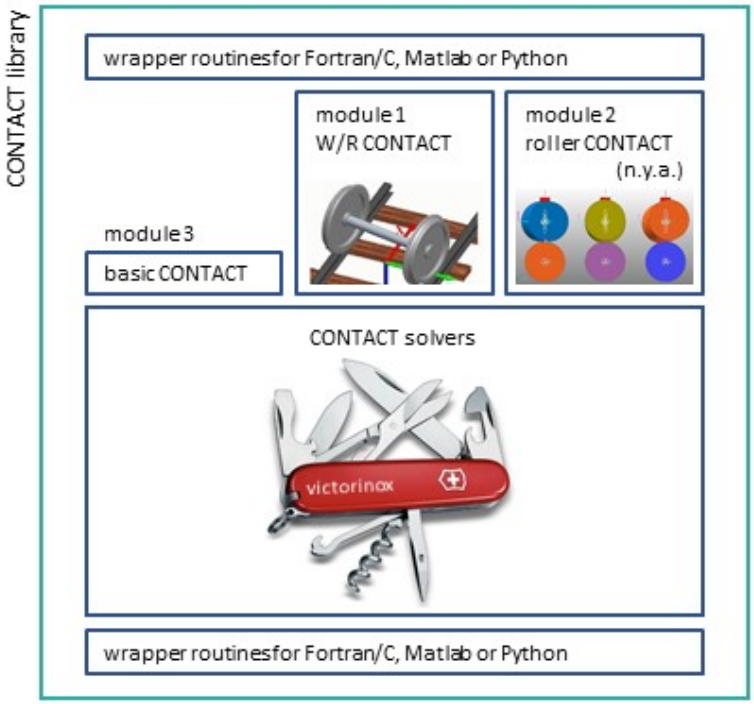



Figure 7.1: *Software architecture for the CONTACT library.*

indicate a task working on all patches of a result element. The memory requirement is proportional to the total number of contact patches 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 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.

## 7.2 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 $\text{mm}^2$. Forces in N and stresses in $\text{N/mm}^2$. Angles are in rad in the calculations, but may be printed in $^\circ$ for easier interpretation. Time is in s, velocities are in mm/s, and temperatures in ℃.

**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 $\text{m}^2$. Forces in N and stresses in $\text{N/m}^2$. All angles are in rad. Times are in s, velocities in m/s, and temperatures in K.

**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 are not affected by the unit convention that's used. The values are needed using mm as the unit of length. An optional scaling factor may be provided to convert to this convention explicitly as discussed in Section 3.3.

## 7.3 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_intfc

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

```
>> help cntc_setwheelsetposition

   function [ ] = cntc_setwheelsetposition(ire, ewheel, nparam, params)

   set the wheelset position state data for a wheel-rail contact problem
     ewheel         - type of position specification (E-digit)
     nparam         - number of parameters provided
     params(nparam) - depending on method that is used

   E=0  : keep wheelset position from previous specification, ignore params
   E=1-8: new wheelset position   params = [s, y, z, roll, yaw, pitch]

   dimensions:   s_ws, y_ws, z_ws [length],       roll, yaw, pitch [angle]
```

There are slight differences between the interfaces for Fortran, C, Matlab and Python, for instance skipping certain length parameters in Matlab and Python that must be provided in Fortran and C. Specific information for C programmers is provided in the file `contact_addon.h`. For Fortran programmers the interfaces are specified in `contact_addon.ifc`.

### 7.3.1 Preparations

```
cntc_initlibrary          - load library into matlab (Matlab)
cntc_initializefirst      - open out-file, initialize data-structures (Fortran/C)
cntc_initialize           - further initializations per result element
cntc_setfileunits         - configure logical file units (Fortran)
cntc_setglobalflags       - configuration of general settings
cntc_managelicense        - perform license activation, refreshing or printing
```

**cntc_initlibrary** The first step in Matlab is to load the library, which is done with `cntc_initlibrary`. Optional arguments can be used to change the output folder or experiment name (default '`contact_addon`').

```
   [CNTC, ifcver, ierror] = cntc_initlibrary;
```

The function returns a struct `CNTC` with so-called 'magic numbers' that are used in the interface. The error code will be $-12$ if no appropriate license could be found, or if a license is found that isn't valid for the actual computation.

In Fortran and C, the analogue is `cntc_initializefirst`. 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, experiment name, file I/O etc. This is done by subroutine `cntc_initia-`

`lizefirst`. This routine will be called by `cntc_initialize` if it isn't called explicitly. In Matlab, this routine is called by `cntc_initlibrary`.

The output-file for the experiment will be opened (default `contact_addon.out`), removing the previous contents if already present. Note that `cntc_initializefirst` will be called again if a new computation is started after all previous result elements have been cleared (`cntc_finalizelast`).

**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 initalizations 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` provides 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.

Configuration flags 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);
```

**cntc_managelicense** The CONTACT library version uses a different license file than the standalone program. This license needs to be activated before the first run and may then be refreshed and printed (Section 2.1.3). These activities can be started using `cntc_managelicense` in Matlab, e.g.

```
license_id = 123456; password = 'abcdef';
ierror = cntc_managelicense('activate', license_id, password);
ierror = cntc_managelicense('print');
```

Print output and error messages are written to `contact_addon.out` (default experiment name). A sample program `caddon_license` is provided that performs the same from the Windows command prompt or Linux command line. The source code of this program is given in the `src` folder. It is configured to print output to the temp folder, `C:\Temp` on Windows, `/tmp` on Linux.

### 7.3.2 Use of parallel computing

CONTACT supports parallel computing in two different ways, both using 'shared memory' parallel programming.

1. Multiple contact problems may be computed concurrently – using multiple threads in the user's program;
2. the calculations per contact problem may use parallel computation as well – using internal parallelization in CONTACT.

The first type requires that the user's (Fortran/C) program is adapted and parallelized, using OpenMP or explicit threading. A few restrictions should be kept in mind:

- The initializations, setting of global flags and finalizations may not be called from parallel regions.
- 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's used. This can be inquired using

```
cntc_getmaxnumthreads       - get max.number of concurrently active threads
```

The second type, internal parallelization, is governed by the flag `CNTC_if_openmp`. The parameter value provided is the number of threads to use per contact problem. This can be set to $-1$ to request as many threads as possible, as instructed by the environment `OMP_NUM_THREADS` and the available cpu cores in the machine.

Internal parallelization doesn't gain much speedup in CONTACT unless certain conditions are met. It is disabled by default, setting the number of threads per contact problem to 1 in `cntc_initializefirst`. This strategy is also disabled when multi-threading of the first type is detected.

### 7.3.3 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 4). 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 (2.3)
cntc_setmetadata            - configuring various metadata in CONTACT (-)
cntc_setsolverflags         - configuration of solver parameters (4.6)

cntc_setmaterialparameters  - set material parameters (4.1)
cntc_settemperaturedata     - set heat related material parameters (4.1.2)
cntc_setfrictionmethod      - set friction parameters (4.2)
cntc_settimestep            - set time step used in shift problems (4.5)
cntc_setreferencevelocity   - set rolling velocity (4.5)
cntc_setrollingstepsize     - set rolling direction and step size (4.5)

cntc_sethertzcontact        - set Hertzian problem specification (4.3.1)
cntc_setpotcontact          - set pot.contact for non-hertzian cases (4.3)

cntc_setpenetration         - set approach/penetration (4.5)
cntc_setnormalforce         - set total normal force (4.5)
cntc_setundeformeddistc     - set undeformed distance function (4.4)

cntc_setcreepages           - set creepages (4.5)
cntc_setextrarigidslip      - set extra term of rigid slip elementwise (4.5)
cntc_settangentialforces    - set total tangential forces (4.5)
```

**cntc_setflags** Subroutine `cntc_setflags` allows for the detailed configuration of contact problems, such as the unit convention used and the setting of control digits. Configuration flags have been given 'magic numbers' that are obtained from `cntc_initlibrary` (Matlab) or `caddon_flags.inc` (Fortran/C). For instance CONTACT's O-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 O = 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 7.3.1). 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 4.6. Additionally, this routine allows configuration of the accuracy with which sensitivities are computed.

**cntc_setmaterialparameters** This subroutine configures the material parameters of Section 4.1: basic elasticity parameters $G, \nu$, needed always, and added parameters for visco-elastic materials (`M` = 1, Section 4.1.3), the Modified Fastsim algorithm (`M` = 2 or 3, Section 4.1.4), and the interfacial layer of `M` = 4 (Section 4.1.5).

**cntc_setfrictionmethod** This subroutine configures the friction parameters corresponding to the `L`-digit, and to the `V`-digit in module 1 (Section 4.2).

**cntc_sethertzcontact** As explained in Section 4.3, 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` = −6 to −1 as described in Sections 4.3.1–4.3.3.

**cntc_setpotcontact** Subroutine `cntc_setpotcontact` provides non-Hertzian options `IPOTCN` = 1 to 4 (Section 4.3). The undeformed distance is consequently described using `cntc_setundeformeddistc`, which provides the various options of `IBASE` as described in Section 4.4. 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 `FSTAT` · `FN`.

### 7.3.4 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_setprofileinputfname   - set a wheel or rail profile via a filename
```

```
cntc_setprofileinputvalues  - set a wheel or rail profile using a table
cntc_getprofilevalues       - retrieve w/r profile after smoothing

cntc_settrackdimensions     - set the track geometry for a w/r problem (3.3)
cntc_setwheelsetdimensions  - set wheelset geometry specification (3.4)
cntc_setwheelsetposition    - set wheelset position specification (3.4)
cntc_setwheelsetvelocity    - set wheelset velocity specification (3.4)
cntc_setwheelsetflexibility - set wheelset flexibility specification
```

**cntc_setprofileinputfname/values** Each wheel/rail problem may hold up to four profiles, for two wheels and two rails. These are configured one by one using the `setprofile` routines, providing appropriate options for their interpretation.

`iparam(1) - itype` : 0 for rails, 1 for wheels, −1 (default) for auto-detect on the basis of the filename extension (`prr` or `ban` for rails, `prw` or `whl` for wheels);

`iparam(2)` : not used;

`iparam(3) - mirrory` : −1 or 0 (default) for no mirroring, 1 for mirroring $y$-values;

`iparam(4) - mirrorz` : 0 (default) for automatic mirroring if needed, −1 for no mirroring, 1 for mirroring $z$-values;

`iparam(5) - errhndl` : configuration of error handling. −2 to continue as much as possible, suppressing error and warning messages; −1 to suppress warning messages; 0 warn and continue (default); 1 signal errors and abort.

`iparam(6) - ismooth` : selection of the smoothing method: 0 for original smoothing spline (default, for now), 1 for weighted smoothing spline with 2nd order penalty, 2 for the weighted B-spline approach with 3rd order penalty.

`rparam(1) - sclfac` : scaling factor to convert input data to [mm], e.g. $10^3$ for data given in [m]. If `sclfac` $\leq 0$ (default), this will be obtained from the active unit convention (Section 7.2).

`rparam(2) - smooth` : smoothness parameter $\lambda$ as discussed in Section 3.3, with smoothing disabled if $\lambda \leq 0$ (default).

`rparam(3) - maxomit` : fraction; signal error if more than `maxomit` of the profile points are removed during profile cleanup. Default 0.5. Set to 1 to disable this error.

`rparam(4) - zigthrs` : angle threshold for zig-zag pattern detection. Default $\pi/2$ rad. Should be larger than $\pi/4$, set to $\geq \pi$ to disable zig-zag detection.

`rparam(5) - kinkhigh` : angle threshold $\delta\alpha_{high}$ for kink detection. Default $\pi/6$ rad. Set to $\geq \pi$ to disable kink detection.

`rparam(6) - kinklow` : angle threshold for neighbouring points at kink detection. Default value $\delta\alpha_{high}/5$.

`rparam(7) - kinkwid` : half-width of window used for kink detection. Default: using profile points within 2 mm on either side of each possible kink location.

Additional configuration parameters are read from (Simpack) `prr` and `prw` files. No further scaling should then be applied (`sclfac = 1`). Simpack's approximation smoothing value is disregarded, smoothing is implemented in a different way in CONTACT.

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`).

**cntc_getprofilevalues** Two different versions of subroutine `cntc_getprofilevalues` are provided with slightly different interfaces. Subroutine `cntc_getprofilevalues` provides the old interface that was used up to release 23.1 for backwards compatibility reasons. Subroutine `cntc_getprofilevalues_new` adds extensions for checking CONTACT's spline approximation.

```
!--original code, works with library versions 18.1 -- 24.2:
   ! set itask, iparam, ...
   call cntc_getprofilevalues(ire, itask, 1, iparam, lenarr, val)
```

This code may be updated as follows:

```
!--intermediate code, works with library versions 18.1 -- 24.1:
   ! set itask, iparam = (/ itype, isampl /), rparam = (/ ds_out /), ...
   if (ifcver.le.2319) then
      call cntc_getprofilevalues(ire, itask, 1, iparam, lenarr, val)
   else
      call cntc_getprofilevalues_new(ire, itask, 2, iparam, 1, rparam, lenarr, val)
   endif
```

In version 25.1, subroutine `cntc_getprofilevalues` will be updated according to the new interface. Using this version, the caller code can be updated as follows:

```
!--intermediate code, works with library versions 23.2 and newer:
   ! set itask, iparam = (/ itype, isampl /), rparam = (/ ds_out /), ...
   if (ifcver.le.2499) then
      call cntc_getprofilevalues_new(ire, itask, 2, iparam, 1, rparam, lenarr, val)
   else
      call cntc_getprofilevalues(ire, itask, 2, iparam, 1, rparam, lenarr, val)
   endif
```

Once version 24.1 is no longer needed, the switch may be removed to reach the final situation:

```
!--final code, works with library versions 25.1 and newer:
   ! set itask, iparam = (/ itype, isampl /), rparam = (/ ds_out /), ...
   call cntc_getprofilevalues(ire, itask, 2, iparam, 1, rparam, lenarr, val)
```

**cntc_settrackdimensions** Two different versions of subroutine `cntc_settrackdimensions` are provided with slightly different interfaces. Subroutine `cntc_settrackdimensions_old` provides the old interface that was used up to release 21.1 for backwards compatibility reasons. Subroutine `cntc_settrackdimensions_new` adds extensions for absolute rail placement.

```
!--original code, works with versions 18.1 -- 22.2:
   params = (/ gaugwd, gaught, cant, 0d0 /)
   call cntc_settrackdimensions(ire, 1, params, 4)
```

This code may be updated as follows:

```
!--intermediate code, works with versions 22.1 -- 24.1:
   if (ifcver.le.2199) then
      params = (/ gaugwd, gaught, cant, 0d0 /)
      call cntc_settrackdimensions_old(ire, 1, params, 4)
   else
      if (gaught.gt.0d0) then
         params = (/ gaught, gaugsq, gaugwd, cant, 0d0 /)
      else
         params = (/ gaught, raily0, railz0, cant, 0d0 /)
      endif
      call cntc_settrackdimensions_new(ire, 1, params, 5)
   endif
```

In version 23.1, subroutine `cntc_settrackdimensions` links to the new interface. Using this version, the caller code can be updated to the final situation:

```
!--final code, works with versions 23.1 and newer:
   if (gaught.gt.0d0) then
      params = (/ gaught, gaugsq, gaugwd, cant, 0d0 /)
   else
      params = (/ gaught, raily0, railz0, cant, 0d0 /)
   endif
   call cntc_settrackdimensions(ire, 1, params, 5)
```

**cntc_setwheelsetdimensions, position, velocity, flexibility** The specification of a wheelset needs geometry parameters for the proper placement of the wheel profile with respect to the wheelset center, position variables placing the wheelset in the track coordinate system, velocities as needed for the calculation of creepages, and optionally flexible wheelset deviations adding to position and velocity data. These aspects are governed by the $E_1$-digit of page 19 and the parameters defined in Section 3.4.

The parameters are configured using four `setwheelset*` routines that all take the $E_1$-digit as input. The same value can be used in successive calls, e.g. using $E_1 = 5$ in four successive calls, to set the position, velocity, dimensions and flexibilities. Each of the routines considers its own aspect only. That is, if you call `cntc_setwheelsetposition` using $E_1 = 1$, this does not clear flexibilities that were set previously by `cntc_setwheelsetflexibility` with $E_1 = 5$.

**cntc_setpotcontact** In wheel/rail contact, subroutine `cntc_setpotcontact` is re-used to set the grid sizes `DX`, `DS` and combination/separation parameters `A_SEP`, `D_SEP` and `D_COMB` (Section 3.6).

### 7.3.5 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.

### 7.3.6 Convencience function in Matlab

```
cntc_getcpresults             - get results of a single contact patch
                                in the form used by loadcase and plot3d
```

A function `cntc_getcpresults.m` is provided in Matlab that currently has no counterpart in Fortran, C and Python. This function will retrieve inputs and outputs from a wheel/rail contact problem from the CONTACT library and create a structure that can be used directly with `plot3d` (Section 6.2.2). The profiles `prr` and `prw` are included in the structure returned and can be omitted calling `plot3d`.

```
sol = cntc_getcpresults(iwhe, icp);

opt = plot3d;
opt.typplot  = 'rw_rear';
opt.field    = 'pn';
opt.rw_surfc = 'both';
plot3d(sol, opt);
```

### 7.3.7 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 and vertical coordinate directions. Forces and moments are provided relative to the track, wheelset, rail profile and wheel profile systems. Moments are computed at the rail or wheel profile markers.

Results can be retrieved for each contact patch separately (`icp` > 0), or summed over the contact patches (`icp` = −1). Forces and moments are defined *on* the output-body, by which the sign depends on the unit convention used (Section 7.2).

**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 38 for further information.

### 7.3.8 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 (4.3)
cntc_getcontactforces       - get normal/tangential forces (4.5, 4.7)
cntc_getpenetration         - get approach/penetration (4.5)
cntc_getcreepages           - get creepages (4.5)
cntc_getcontactpatchareas   - get size of contact area (-)
cntc_getmaximumpressure     - get maximum pressure (-)
cntc_getmaximumtraction     - get maximum shear stress (-)
cntc_getmaximumtemperature  - get maximum surface temperatures (-)
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 4.3), 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 (Section 4.5). The signs depend on the unit convention selected (Section 7.2), acting *on* the output-body.

The forces and moments are aggregated at the contact reference position of the patch. 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 4.5.2), in shifts (`T` = 1) they are shift distances (Section 4.5.1). `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 or elements in each state and multiplying by $\delta x \cdot \delta y$.

**cntc_getmaximumtraction** This subroutine determines the maximum tangential traction over all elements of the contact area, $\sqrt{p_x^2 + p_y^2}$.

**cntc_getsensitivities** Sensitivities are computed when the `S`2-digit '`SENS`' is set to 2 or 3. This uses small perturbations of the 'input variables', i.e. the approach $\delta$ and creepages $\xi, \eta, \phi$. 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.

### 7.3.9 Detailed outputs of the contact problem

Detailed outputs are presented by the following routines:

```
cntc_getelementdivision    - get elementwise adhesion/slip/plast areas (4.8)
cntc_getfielddata          - get elementwise output values (4.8)
cntc_gettractions          - get elementwise tractions (4.8)
cntc_getdisplacements      - get elementwise displacement differences (4.8)
cntc_getmicroslip          - get elementwise micro-slip velocity (4.8)
```

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 7.3.8).

**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_getfielddata** Provides various arrays selected via the `ifld` input, e.g. `CNTC_fld_pn` for `PN`, `CNTC_fld_temp1` for `TEMP1`, etc.

**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 7.2).

**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.

### 7.3.10 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
```

Subroutine `cntc_finalize` will automatically call `cntc_finalizelast` if there are no more result elements remaining. This closes the output file for the experiment. Making further calls after this, subroutine `cntc_initializefirst` will be called again, removing the previous contents of the out-file.

## 7.4 Calculation of subsurface stresses

The calculation of subsurface stresses works with 3D 'blocks' of $\mathtt{nx} \cdot \mathtt{ny} \cdot \mathtt{nz}$ points. One or more of such blocks can be defined, before or after the solution of surface contact problems (Section 7.3).

```
subs_addblock      - define a 3D grid for subsurface stress computation
subs_calculate     - perform actual subsurface stress calculation
subs_getblocksize  - get the number of points used in one block
subs_getresults    - get detailed outputs of subsurface stress calculation
```

**subs_addblock** Each contact problem (result element) can hold up to 10 blocks, numbered consecutively from 1 to `nblk`. These must be added in increasing order: $1, 2, \ldots$. Existing blocks are discarded when re-using a block number `iblk` $\leq$ `nblk`. That is, defining 'block 1' anew will clear the whole list and set new values for `iblk` = 1. Likewise, defining 'block 3' anew will keep the existing blocks 1–2 and discard all blocks thereafter.

In module 1, blocks may be defined for each contact patch separately (`icp` $> 0$), or for the wheel/rail contact problem as a whole (`icp` $= -1$). The two methods may be combined, with blocks defined for some contact patches (e.g. `icp` = 2) and for the wheel/rail problem as a whole. The latter option then serves as a fall-back for contact patches where no specific block data are given.

Blocks are defined via `ISUBS` = 1–9 as described in Section 4.9.2. The $xyz$-coordinates given are interpreted as $xsn$-coordinates, using the contact local reference system.

**subs_calculate** A separate calculation routine is provided to activate the subsurface stress calculation. This routine requires that `cntc_calculate` has been used first, to solve the contact problem.

- An 'on-line' way of working is to configure (`addblock`) the subsurface problem together with the surface contact problem, then calling `cntc_calculate` and `subs_calculate` directly after another.
- An 'off-line' way of working is to solve the surface problem first (`cntc_calculate`), then define (`addblock`) and compute the blocks (`subs_calculate`) on the basis of the surface results, or even re-define (iterate) the subsurface problem, for instance to zoom in on a region where the largest values are found.

**subs_getblocksize** Linked to the size of the potential contact area, the size of the blocks may be unknown in advance. This size can then be inquired with `subs_getblocksize`, to get the proper array-size for the subsurface results.

**subs_getresults** For each block of points, a table is computed with all the relevant output variables. The relevant columns of this table may be retrieved using `subs_getresults`. The columns are defined as follows:

- columns 1–3: $[xyz]$ coordinates of the points;
- columns 4–6: displacements $u_x, u_y, u_z$;

- columns 7–9: hydrostatic, von Mises and Tresca-stresses $\sigma_{hyd}$, $\sigma_{vm}$, $\sigma_{tr}$;
- columns 10–12: principal stresses $\sigma_1, \sigma_2, \sigma_3$;
- columns 13–21: components of the stress tensor: $\sigma_{xx}, \sigma_{yx}, \ldots, \sigma_{zz}$.

The precise meaning of these is specified in Section 4.9.3.

## 7.5 Examples for the CONTACT library

### 7.5.1 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 5.9.

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 5.13.

### 7.5.2 Calculation of wheel/rail contact

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 5.7.

### 7.5.3 Calculation of subsurface stresses

The calculation of subsurface stresses is shown in `matlab_subsurf.m`. This computes the same cases as in the `subsurf`-example of Section 5.6, creating the pictures of Figures 5.6 and 5.7.

# 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. Blanco-Lorenzo, J. Santamaria, E.G. Vadillo, and N. Correa. On the influence of conformity on wheel-rail rolling contact mechanics. *Tribology International*, 103:647–667, 2016.

[3] J. Blanco-Lorenzo, E.A.H. Vollebregt, J. Santamaria, and E.G. Vadillo. Approximating the influence coefficients of non-planar elastic solids for conformal contact analysis. *Tribology International*, 154:106671, 2021.

[4] A.F. Bower. *Applied Mechanics of Solids*. CRC Press, Taylor and Francis Group, Boca Raton, 2010. See `www.solidmechanics.org`.

[5] R.R. Craig Jr. and A.J. Kurdila. *Fundamentals of structural dynamics, 2nd edition*. John Wiley & Sons, Hoboken, New Jersey, 2006.

[6] 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.

[7] B. Engel, H.P. Beck, and J. Alders. Verschleißreduzierende Radschlupfregelung mit hoher Kraftschlussausnutzung. *Elektrische Bahnen*, 96:201–209, 1998.

[8] M. Ertz and K. Knothe. A comparison of analytical and numerical methods for the calculation of temperatures in wheel/rail contact. *Wear*, 253:498–508, 2002.

[9] R.C. Hibbeler. *Engineering Mechanics - Statics and Dynamics, 13th Edition*. Pearson Prentice Hall, Upper Saddle River, New Jersey, 2012.

[10] K. Hou, J. Kalousek, and E. Magel. Rheological model of solid layer in rolling contact. *Wear*, 211:134–140, 1997.

[11] K.L. Johnson. *Contact Mechanics*. Cambridge University Press, Cambridge (UK), 1985.

[12] J.J. Kalker. *On the rolling contact of two elastic bodies in the presence of dry friction*. PhD thesis, Delft University of Technology, 1967.

[13] J.J. Kalker. Simplified theory of rolling contact. *Delft Progress Report Series C1*, 1:1–10, 1973.

[14] J.J. Kalker. The computation of three-dimensional rolling contact with dry friction. *Int. Journ. for Numerical Methods in Engineering*, 14:1293–1307, 1979.

[15] J.J. Kalker. A fast algorithm for the simplified theory of rolling contact. *Vehicle System Dynamics*, 11:1–13, 1982.

[16] J.J. Kalker. Numerical calculation of the elastic field in a half-space. *Comm. Appl. Num. Meth.*, 2:401–410, 1986. Reprinted as Appendix C in [17].

[17] J.J. Kalker. *Three-Dimensional Elastic Bodies in Rolling Contact*, volume 2 of *Solid Mechanics and its Applications*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1990.

[18] J.J. Kalker. Book of tables for the Hertzian creep-force law. In Zobory [75], pages 11–20.

[19] 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.

[20] 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.

[21] W. Kik and J. Piotrowski. A fast approximate method to calculate normal load at contact between wheel and rail and creep forces during rolling. In Zobory [75].

[22] J. Kopp. Efficient numerical diagonalization of Hermitian 3×3 matrices. *International Journal of Modern Physics C*, 19:523–548, 2008.

[23] Zili Li. *Wheel-rail rolling contact and its application to wear simulation*. PhD thesis, Delft University of Technology, 2002.

[24] 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.

[25] M. Maglio. *Influence of railway wheel tread damage and track properties on wheelset durability – field tests and numerical simulation*. PhD thesis, Chalmers University of Technology, 2023.

[26] M. Maglio, T. Vernersson, J.C.O. Nielsen, A. Ekberg, and E. Kabo. Influence of railway wheel tread damage on wheel-rail impact loads and the durability of wheelsets. *Railway Engineering Science*, 2023.

[27] M. Malvezzi, E. Meli, S. Falomi, and A. Rindi. Determination of wheel-rail contact points with semianalytic methods. *Multibody System Dynamics*, 20:327–358, 2008.

[28] Klara Mattsson. Wheel-rail impact loads generated by wheel flats. Master's thesis, Chalmers University of Technology, 2023.

[29] S.Z. Meymand, A. Keylin, and M. Ahmadian. A survey of wheel-rail contact models for rail vehicles. *Vehicle System Dynamics*, 54:368–428, 2016.

[30] 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.

[31] P.M. Naghdi. *P.M. Naghdi's Notes on Continuum Mechanics*. University of California, Department of Mechanical Engineering, Berkeley, 2001.

[32] J.B. Nielsen and A. Theiler. Tangential contact problem with friction coefficients depending on sliding velocity. In Zobory [75], pages 44–51.

[33] J. Piotrowski and W. Kik. A simplified model of wheel/rail contact mechanics for non-Hertzian problems and its application in rail vehicle dynamics simulations. *Vehicle System Dynamics*, 46(1-2):27–48, 2008.

[34] 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.

[35] O. Polach. Creep forces in simulations of traction vehicles running on adhesion limit. *Wear*, 258:992–1000, 2005.

[36] V.L. Popov. *Contact Mechanics and Friction. Physical Principles and Applications*. Springer-Verlag, Berlin, 2010.

[37] D. Roylance. Engineering viscoelasticity. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 2001.

[38] A.A. Shabana. *Dynamics of Multibody Systems – Fourth Edition*. Cambridge University Press, New York, 2013.

[39] A.A. Shabana, K.E. Zaazaa, and H. Sugiyama. *Railroad Vehicle Dynamics: A Computational Approach*. CRC Press, Boca Raton, 2008.

[40] P. Shackleton and S.D. Iwnicki. Wheel-rail contact benchmark, version 3.0. Rail Technology Unit, Manchester Metropolitan University, 2006.

[41] 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.

[42] M.Sh. Sichani, R. Enblom, and M. Berg. A novel method to model wheel-rail normal contact in vehicle dynamics simulation. *Vehicle System Dynamics*, 52:1752–1764, 2014.

[43] 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.

[44] 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.

[45] 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.

[46] 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. University of Firenze. Open access.

[47] E.A.H. Vollebregt. User guide for CONTACT, J.J. Kalker's variational contact model. Technical Report TR09-03, v9.1, VORtech, 2009.

[48] 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*, pages 1–15, Stirlingshire, United Kingdom, 2010. Civil-Comp Press. DOI: 10.4203/ccp.93.17.

[49] 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.

[50] 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.

[51] E.A.H. Vollebregt. Numerical modeling of measured railway creep versus creep-force curves with CONTACT. *Wear*, 314:87–95, 2014.

[52] 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.

[53] 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. Open access.

[54] E.A.H. Vollebregt. Comments on "the Kalker book of tables for non-Hertzian contact of wheel and rail". *Vehicle System Dynamics*, 56(9):1451–1459, 2018. DOI: 10.1080/-00423114.2017.1421767.

[55] E.A.H. Vollebregt. Conformal contact: Corrections and new results. *Vehicle System Dynamics*, 56(10):1622–1632, 2018. DOI: 10.1080/00423114.2018.1424917.

[56] E.A.H. Vollebregt. Detailed wheel/rail geometry processing using the conformal contact approach. *Multibody System Dynamics*, 52:135–167, 2021. Open access.

[57] E.A.H. Vollebregt. Detailed wheel/rail geometry processing using the planar contact approach. *Vehicle System Dynamics*, 60(4):1253–1291, 2022. Open access.

[58] E.A.H. Vollebregt. License management for CONTACT version 22.1. Memo EV/M21.004, Vtech CMCC, August 2022. Open access.

[59] E.A.H. Vollebregt, A. Darbani, A. Ashtekar, and K. Oldknow. Smoothing procedures for detailed wheel/rail geometry processing. In P. Meehan and W. Yan et al., editors, *Proceedings of the 12th International Conference on Contact Mechanics and Wear of Rail/Wheel Systems*, pages 1–7, Australia, 2022. Monash University.

[60] 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.

[61] E.A.H. Vollebregt, J.J. Kalker, and G.Q. Wang. CONTACT'93 users manual. Technical report, Delft University of Technology, 1993.

[62] E.A.H. Vollebregt, P. Klauser, A. Keylin, P. Schreiber, D. Sammon, and N. Wilson. Extension of CONTACT for switches and crossings and demonstration for S&C benchmark cases. In W. Huang and M. Ahmadian, editors, *The 28th IAVSD Symposium on Dynamics of Vehicles on Roads and Tracks (IAVSD2023)*, Lecture Notes in Mechanical Engineering, page paper 236, Cham, 2023. Springer.

[63] 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.

[64] E.A.H. Vollebregt and A. Segal. Solving conformal wheel-rail rolling contact problems. *Vehicle System Dynamics*, 52(suppl. 1):455–468, 2014. DOI: 10.1080/00423114.2014.906634.

[65] E.A.H. Vollebregt, K. Six, and O. Polach. Challenges and progress in the understanding and modelling of the wheel–rail creep forces. *Vehicle System Dynamics*, 59(7):1026–1068, 2021. Open access.

[66] E.A.H. Vollebregt and C.D. van der Wekken. Advanced modeling of wheel-rail friction phenomena. Technical Report TR19-11, VORtech, November 2019. FRA project.

[67] E.A.H. Vollebregt and P. Voltr. Improved accuracy for FASTSIM using one or three flexibility values. *Vehicle System Dynamics*, 61(1):309–317, 2023. Open access.

[68] 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*, pages 1–6, Manchester, 2011. IAVSD. Open access.

[69] 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. Open access.

[70] 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.

[71] G. Wang and K. Knothe. Stress analysis for rolling contact between two viscoelastic cylinders. *Journal of Applied Mechanics; Transactions ASME*, 60:310–317, 1993.

[72] 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*, 24(11):3537–3553, 2019.

[73] 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 colloquim 578 on Rolling Contact Mechanics for Multibody System Dynamics*, pages 1–11, Lisbon, Portugal, 2017. IDMEC. Open access.

[74] 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.

[75] I. Zobory, editor. *Proceedings of the 2nd Mini Conference on Contact mechanics and Wear of Wheel/Rail systems*. Technical University of Budapest, Hungary, 1996.

# 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 the contact problem (A.2) and also for calculation of subsurface stresses (A.5);

`<profile>.<ext>` : input for rail (`prr`, `ban`) and wheel profiles (`prw`, `whl`) as discussed in Section 3.2;

`<varprof>.slcs` : input for variable rail profiles as used for switches and crossings (A.3);

`<inflcf>.txt` : input for numerical influence coefficients (A.4);

`<experim>.out` : output file;

`<experim>.<ncase>.mat` : tractions file, for communication with Matlab (A.6);

`<experim>.<ncase>.subs` : subsurface stresses file, for communication with Matlab (A.7).

## 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 2.1.1). 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;
- the maximum length of each line is 256 characters, with maximally 50 items per line. Empty lines are skipped, just as lines containing comments only;
- 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;
- newlines may also be used to separate values, i.e. groups of values may be entered using multiple lines;
- some data values are optional. These are indicated in curly braces `{}`. They must be entered on the same line as the preceding, mandatory value. Newlines cannot be used in this case;
- 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;
- some blocks are needed only in certain circumstances. They are enclosed in square brackets `[]`, preceded by a condition in the specification;
- some blocks can be repeated. They are enclosed in `[]` and are preceded by a stop-condition;
- 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` → `MX,MY,AA,BB,SCALE`), in order not to confuse yourself;
- This also applies for the `N`- (`PEN` or `FN` for `N` = 0 or 1) and `F`-digits (`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.

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 2.3:

CPBTNFS     (CONFIG, PVTIME, BOUND , TANG  , NORM  , FORCE , STRESS)
VLDCMZE     (VARFRC, FRCLAW, DISCNS, INFLCF, MATER , ZTRACK, EWHEEL)
HGIAOWR     (HEAT  , GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW  , RETURN)

% Parameters for the iterative solution algorithms (section 4.6):

if G<>1:      [ MAXGS   MAXIN   MAXNR   MAXOUT  EPS   {NPOT_MAX}   ]
if G=2,3:     [ OMEGAH  OMEGAS  INISLP  OMGSLP       ]
if G=4:       [ INISLP  OMGSLP                       ]

% Friction description (section 4.2):
%      L=0 --> Coulomb friction with constant coefficient of friction
%      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
%      L=6 --> temperature dependent friction with piecewise linear formula

if VARFRC=0:
      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=6: [ FREF   TREF   DFHEAT  DTHEAT               ]
if VARFRC=1:  [ NVF                                        ]
      if L=0: [ ALPHA  FSTAT  FKIN                         ], NVF times
      if L=2: [ ALPHA  FKIN   FLIN1  SABSH1  FLIN2  SABSH2 ], NVF times
      if L=3: [ ALPHA  FKIN   FRAT1  SABSH1  FRAT2  SABSH2 ], NVF times
      if L=4: [ ALPHA  FKIN   FEXP1  SABSH1  FEXP2  SABSH2 ], NVF times
      if L=6: [ ALPHA  FREF   TREF   DFHEAT  DTHEAT        ], NVF times
if L=2,3,4,6: [ MEMDST MEM_S0                              ]

% Information needed for influence coefficients,
% particularly the material parameters (section 4.1):

if INFLCF>=2:
      if INFLCF=4: [ IF_METH  VARIANT              ]
                     POISS1  POISS2  GG1     GG2
      if MATER =1: [ FG1     FG2       VT1     VT2  ]
      if MATER =2: [ FLX     KO_MF   ALFAMF  BETAMF ]
```

```
      if MATER =3: [         KO_MF    ALFAMF   BETAMF ]
      if MATER =4: [ GG3     LAYTHK   TAU_CO   K_TAU  ]

% Material parameters for temperature calculation (section 4.1):

if H=3:      [ BKTEMP1  HEATCP1  LAMBDA1  DENS1
               BKTEMP2  HEATCP2  LAMBDA2  DENS2          ]
if H=3, M=4: [ BETAPL                                    ]

% Information needed for the grid discretization (section 3.6):

if D>=2:     [ DX    DS    DQREL    A_SEP    D_SEP    D_COMB   {D_TURN} ]

% Information on the track geometry (section 3.3) & rail profile (section 3.2):

if Z=1,3:
      if CONFIG=0,1:  [  GAUGHT    GAUGSQ    GAUGWD    CANT      ]  if GAUGHT >  0
                 or:  [  GAUGHT    RAILY0    RAILZ0    CANT      ]  if GAUGHT <= 0
      if CONFIG=4,5:  [  GAUGHT    GAUGSQ    GAUGWD    NOMRADR   ]  if GAUGHT >  0
                 or:  [  GAUGHT    RAILY0    RAILZ0    NOMRADR   ]  if GAUGHT <= 0

if Z=3:        [ 'RFNAME'  MIRRORY  SCLFAC    SMOOTH  {MIRRORZ}  {PRFOPT} ]
if PRFOPT>=1:  [  ISMOOTH  ZIGTHRS  KINKHIGH  KINKLOW  KINKWID ]

if Z=2,3:      [  DYRAIL   DZRAIL   DROLLR    VYRAIL   VZRAIL   VROLLR  ]


% Information on wheelset geometry (section 3.4) & wheel profile (section 3.2):

if E=3,5:      [  FBDIST   FBPOS     NOMRADW   ]
if E=3,5:      [ 'WFNAME'  MIRRORY   SCLFAC    SMOOTH  {MIRRORZ}  {PRFOPT} ]
if PRFOPT>=1:  [  ISMOOTH  ZIGTHRS   KINKHIGH  KINKLOW  KINKWID ]


% Information on the wheelset state (section 3.4):

if CONFIG=0,1, E>=1: [ S_WS    Y_WS    Z_INP  ROLL    YAW    PITCH    ]
if CONFIG=4,5, E>=1: [ X_WS    Y_WS    Z_INP  ROLL    YAW    PITCH    ]
if CONFIG=0,1, E>=2: [ VS_WS   VY_WS   VZ_WS  VROLL   VYAW   V_INP    ]
if CONFIG=4,5, E>=2: [ RPITCH  VY_WS   VZ_WS  VROLL   VYAW   V_INP    ]
   with  N=0: Z_INP = Z_WS,   N=1: Z_INP = FZ_WS
         F=0: V_INP = VPITCH,

% Information on flexible wheelset deviations (section 3.4):
```

```
if E>=4:    [ DXWHL    DYWHL    DZWHL    DROLLW   DYAWW    DPITCHW
              VXWHL    VYWHL    VZWHL    VROLLW   VYAWW    VPITCHW    ]

% Subsurface stress calculation (section 4.9):

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 2.3:

 PBTNFS             (PVTIME, BOUND , TANG  , NORM  , FORCE , STRESS)
 LDCMZE             (FRCLAW, DISCNS, INFLCF, MATER , RZNORM, EXRHS )
HGIAOWR     (HEAT  , GAUSEI, IESTIM, MATFIL, OUTPUT, FLOW  , RETURN)

% Parameters for the iterative solution algorithms (section 4.6):

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 4.5):
% 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 4.2):
%      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
%      L=6 --> temperature dependent friction with piecewise linear formula

if VARFRC=0:
      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=6: [ FREF   TREF   DFHEAT  DTHEAT              ]
if L=2,3,4,6: [ MEMDST MEM_S0                             ]

% Information needed for influence coefficients,
% particularly the rolling direction and step size (section 4.5),
% and the geometry/material configuration (section 4.1):

if C>=2:
      if T>1: [ CHI  DQ   VELOC             ]
      if C=4: [ IF_METH   VARIANT   NN
                Y(I), ALPHA(I), I=1,NN      ]
      if C=9: [ 'CFNAME'                    ]
                POISS 1, 2   GG 1, 2
      if M=1: [ FG 1, 2   VT 1, 2           ]
      if M=2: [ FLX  K0_MF   ALFAMF  BETAMF ]
      if M=3: [      K0_MF   ALFAMF  BETAMF ]
      if M=4: [ GG3  LAYTHK  TAU_C0  K_TAU  ]

% Material parameters for temperature calculation (section 4.1):

if H=3:      [ BKTEMP1  HEATCP1  LAMBDA1  DENS1
               BKTEMP2  HEATCP2  LAMBDA2  DENS2 ]
if H=3, M=4: [ BETAPL                           ]

% Information needed for the grid discretization (section 4.3).
% Hertzian options when IPOTCN<0 (sections 4.3.1, 4.3.2, 4.3.3), direct
% specification of potential contact area when IPOTCN>0 (section 4.3.4).

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  A1  BB   SCALE ]
      if IPOTCN=-5: [ MX  MY  AA  BB   SCALE ]
      if IPOTCN=-6: [ 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 4.4):

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 4.5):

if E=9: [ EXRHS(I,2), EXRHS(I,3), I=1,NPOT    ]

% Subsurface stress calculation (section 4.9):

if S=2,3: [ subsurface points (see below) ]
```

## A.3 Specification of the file <varprof>.slcs/.slcw

The input of variable rail profiles in so-called 'slices files' (.slcs) starts with general configuration parameters. After this follows the list of filenames per slice, followed by the optional information on 'features' and 'parts' in lateral direction.

```
% General parameters and counters

S_OFFSET   S_SCALE
NSLC
NFEAT      NKINK      NACCEL
S_METHOD

% Slice positions and filenames per slice

[ S_SLC     RFNAME ],  NSLC times

% Feature information per slice

if (NKINK >0)  [ P_KINK(J),  J=1,NKINK    ]
if (NACCEL>0)  [ P_ACCEL(J), J=1,NACCEL   ]
if (NFEAT >1)  [ S_SLC  [ S_F(J), J=1,NFEAT ] ],  NSLC times
```

The positions of kinks and accelerations are entered each as one 'group of values'. They may be presented on a single line of input or may be separated by newlines as discussed in Section A.2.

The slices filenames and feature information use a group of values per slice. Newlines are optional within each slice and mandatory between consecutive slices.

The structure is the same for out-of-round wheels using 'wheel slices' (`.slcw`) files, using `TH_OFFSET` instead of `S_OFFSET`, `TH_SCALE` instead of `S_SCALE`, `TH_METHOD` instead of `S_METHOD`, and `TH_SLC` instead of `S_SLC`.

## A.4 Specification of the file of numerical influence coefficients

A file of numerical influence coefficients may be provided for conformal contact situations, with name `CFNAME`, using control digit $C_3 = 9$, as discussed in Section 4.1.6. This concerns the coefficients $C_{IiJj}$ of the tractions-displacements relation:

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

By expansion of the element numbers $I = (i_x, i_y)$, $J = (j_x, j_y)$, this gives an array $C(i_x, i_y, i, j_x, j_y, j)$ with six dimensions. This is reduced to five dimensions by assuming $C(i_x, :, :, j_x, :, :) = C'(i_x - j_x, :, :, :, :)$ for all pairs $i_x, j_x$. Another dimension may be removed in situations with (near) constant curvature, cf. `if_meth` = 0 in Section 4.1.6, assuming $C'(:, i_y, :, j_y, :) = C''(:, i_y - j_y, :, :)$.

There are two types of files, with a single 'matrix' $C''$ for all loadings, or $m_y$ separate matrices, slices $C'(:, :, :, j_y, :)$, for loadings at points $j_y = 1 \cdots m_y$.

```
nmatrix   iaconvex
lquasid   xyzorder
lsymmx    lsymmy
mxfile    myfile   dxfile  dyfile
% empty/comment line
% empty/comment line
for imatrix = 1 to nmatrix: [
   if nmatrix>1: [
      % empty/comment line
      imatrix  xcentr  ycentr
   ]
   for iyofs = -my+1 to my-1: [
      % empty/comment line
      % empty/comment line
      for ixofs = 0 to mx-1: [
         ixofs   xrel   Czz  Czx  Czy   Cxz  Cxx  Cxy   Cyz  Cyx  Cyy
      ]
   ]
]
```

The counters `ixofs`, `iyofs` stand for the offsets $i_x - j_x$ and $i_y - j_y$, with the loaded element placed at $j_x = j_y = 0$. Each matrix has the biggest values for `Cxx`, `Cyy`, `Czz` at `ixofs` = `iyofs` = 0.

The software requires non-quasi-identical input (9 coefficients $i, j$) in $zxy$-order. Data should be using symmetry in $x$ and not in $y$. The parameters `lquasid`, `xyzorder`, `lsymmx`, `lsymmy` are provided for future extension. The number of comment lines must be exactly as shown in the specification.

The grid employed in the file can have different numbers of rows and columns than the grid in CONTACT. If the file grid is smaller, the influence coefficients are padded with zeros. This creates inaccuracies in the simulation. If the file grid is larger than the grid used in the current case being computed, the additional values from the file will be discarded.

## A.5 Subsurface-stress input in the file `<experim>.inp`

The input of the subsurface points in `<experim>.inp` starts with the control digits `A` and `O`, governing the writing of the Matlab-file `<experim>.<case>.subs` and the level of output to the `out`-file (Section 4.9.1). 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` · `NY` · `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 4.9.2):

```
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.6 Specification of the file <experim>.<case>.mat

This file is written for each case with A-digit ≥ 1 and contains the detailed outputs 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 6.2).

Results 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. All lines lines have an equal number of columns. There are 11–15 columns, depending on the presence of plasticity and temperature in the calculations.

```
line  1: % comment line, describing wheel and rail markers
line  2:   Tim   Sws   Xw    Yw      ...                  ...   (dum)  Fmt
line  3: % comment line, describing contact reference position
line  4:   Xcp   Ycp   Zcp   Deltcp  ...
line  5: % comment line, describing grid discretisation variables
line  6:   Mx    My    Xl    Yl    Dx    Dy   Chi   Dq  ...
line  7: % comment line, describing the material parameters used
line  8:   Tdigit Mdigit Gg1    Gg2    Poiss1 Poiss2
line  9: % comment line, describing the friction law used
line 10:   Ldigit  (dum)  Veloc  (if L=0:) Fstat  Fkin
line 11: % 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) ...
                           [ TauCrt(i) UplsX(i) UplsY(i) ] [ Temp1(i) Temp2(i)]
where:
  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
  Mu(i)    = actual friction coefficient in element i
  Pn,Px,Py = normal/tangential tractions in element i
  Un,Ux,Uy = normal/tangential displacement differences in element i
  SRel(i)  = magnitude of the relative slip velocity in element i
  TauCrt   = plasticity bound in element i
  Uplsx,y  = tangential plastic displacements in element i
  Temp1,2  = optional surface temperature of bodies 1, 2 at element i
```

Note: in the computation of shifts (T = 1) instead of rolling (T ≥ 2), the shift distance Shft is stored instead of Srel.

## A.7 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 20. 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_hyd = I_1/3
SigVM    = von Mises stress sigma_vm = sqrt(3*J_2)
Sigxx,...,Sigzz = components of the stress tensor
```

Refer to Sections 4.8 and 4.9 for the description of these output quantities.

# Appendix B

# 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.

High-level functions are provided for wheel-rail contacts in 'module 1'. This is based on wheel and rail profiles and corresponding dimensions and states. In this case, CONTACT performs analysis of the contact geometry problem, identifying contact patches and computing the creepage [57, 56]. Low-level functions are provided in 'module 3' that are more general by nature but may be harder to use.

## B.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, with rail vehicle dynamics the most important exponent for 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

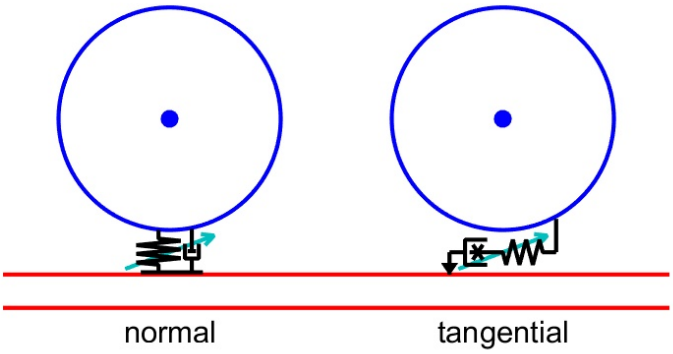



Figure B.1: *Schematization of the wheel-rail rolling contact forces: acting mainly as a variable spring in normal direction, Hertzian or non-Hertzian, and like a variable spring and dashpot tangentially, that breaks loose if a large a force is required.*

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's 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 B.1). In 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' centers 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 bodies' centers 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 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 some energy loss, i.e. this displacement is not undone upon releasing the force. This is indicated by the vertical arrow in Figure B.1, indicating the reference position against which the spring is pushing. Finaly, if the external force is large, compared to the normal load that presses the bodies together, then gross sliding occurs, amounting to further displacement of the vertical arrow.

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

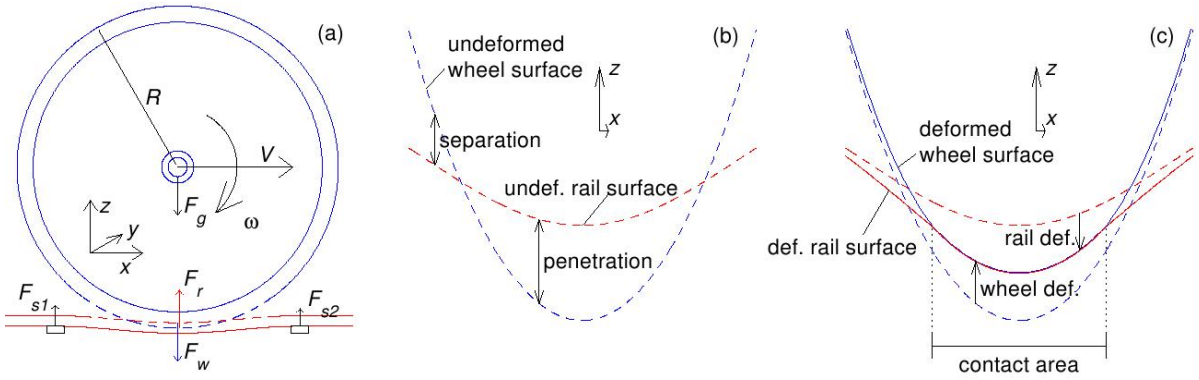



Figure B.2: *Illustration of the wheel-rail contact problem [63]. (a): overall geometry. (b): wheel and rail geometries in so-called undeformed states. (c): wheel and rail geometries in the deformed state and corresponding deformations and contact area. Note: graphs (b) and (c) are stretched vertically.*

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 creeping can be schematised using the dashpot shown in Figure B.1, *in series* with the tangential elasticity.

Creepage and tangential force are related to another in a complex, non-linear way. In steady state rolling and at low creepage, the force and percentual creepage are related according to the linear theory of Kalker [12, 17]. 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 5.13). 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 B.1. The tangential contact spring comes into play during non-steady scenarios, and may then lead to tangential *rocking* to occur [52, 53].

## B.2 Overall problem versus the contact problem

The geometry of the wheel-rail contact problem is introduced in Figure B.2. In graph (a), the overall geometry is shown: the rolling wheel, with radius $R$, forward velocity $V$ and angular velocity $\omega$, 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 B.2 (b), stretched vertically. The distance by which the wheel is shifted down equals the maximum penetration and is called the *approach*, $\delta_n$. 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

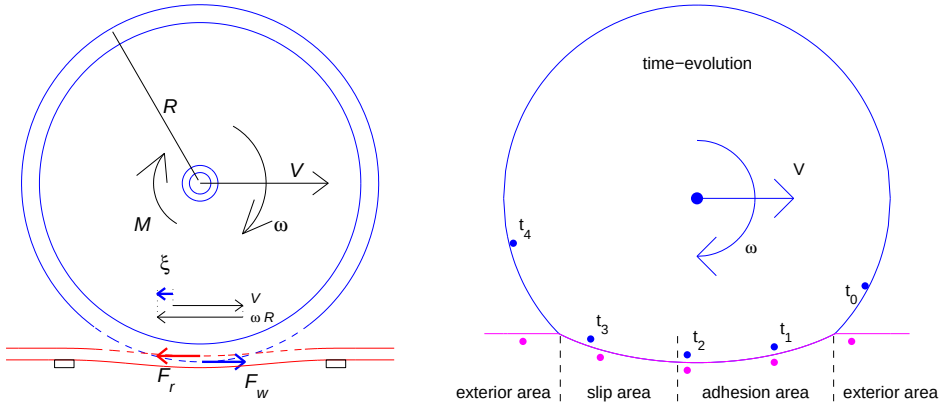



Figure B.3: *Left: illustration of the creepage $\xi$, i.e. the relative velocity difference $(V + \omega R)/V$. Right: particles that move through the contact area (from right to left) tend to adhere to the opposite surface first, are strained more and more until local sliding (micro-slip) sets in.*

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 B.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 reaction forces in the contact patch.

The elastic deformations and resulting contact patch are illustrated in Figure B.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 $10 \times 10$ 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 [17].

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 B.3, left. When a tractive force (torque $M$) is aplied on a railway wheel, a small difference arises between the overall forward velocity $V$ and circumferential velocity $\omega R$ (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.

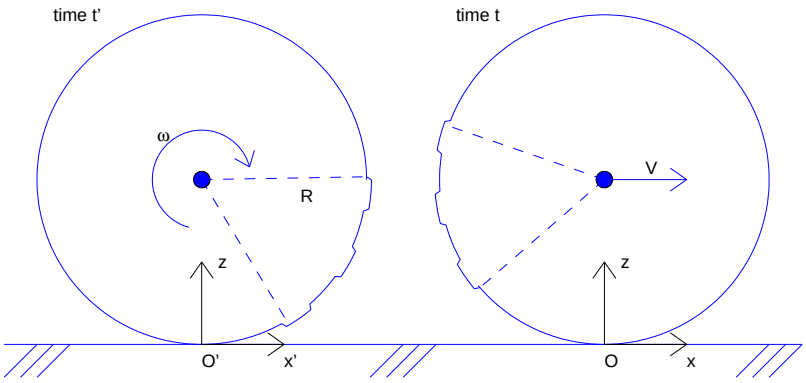



Figure B.4: *Schematic example illustrating non-stationary rolling of a wheel with some irregularities on a rail.*

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 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 $\eta$ 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 $\phi$. These combined creepages lead to combined forces $F_x, F_y$ and the spin moment $M_z$.

Figure B.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.

## B.3 Contact-fixed and world-fixed coordinate systems

A typical geometrical configuration is presented in Figure B.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 small, at most a few

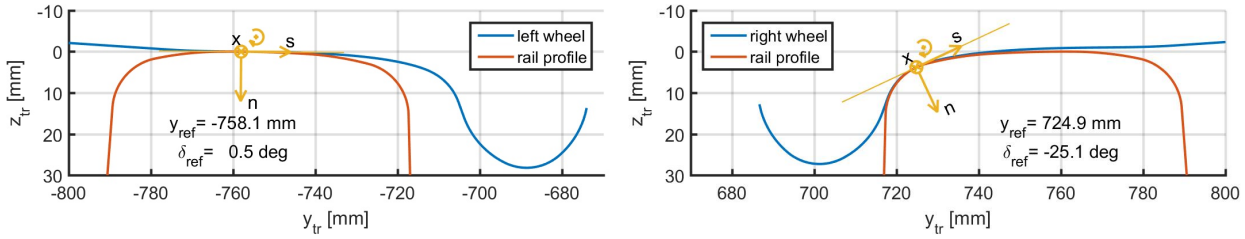



Figure B.5: *Initial contact points for left and right wheels at wheelset lateral displacement* $y_{ws} = 6$ mm*, yaw* $\psi_{ws} = 0°$ *(Manchester contact benchmark, see Section 5.7).*

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 B.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 [17].

## B.4 Using local coordinates

The consequence of anisotropy (Figure B.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 $\delta$ that arises when the contact moves from the wheel tread towards the flange, as illustrated in Figure B.5. Note that $z_{tr}$ and $n$ are defined positive downwards, which makes the rail the upper body.

The planar contact approach used in CONTACT relies on a so-called 'contact reference point' [57]. This serves as the origin for the local coordinate system, and as the 'spin center' for spin creepage [54]. The contact reference point can be set to the geometric point of contact (initial contact position).

Alternatives are to use the center of gravity of the interpenetration area [68], or the pressure center of gravity [54].

The steps for solving a contact problem are then:

1. Locate a contact patch, for instance using the initial contact position, and define the '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 (so-called rigid slip velocity) relative to the contact plane;
4. Solve the normal and tangential contact problems;
5. Convert results to the global coordinates.

Planar contact coordinates are designated as $[\bar{x}, \bar{s}, \bar{n}]^T$, or as $[x, s, n]^T$ or $[x, y, z]^T$ if there's no ambiguity on the interpretation.

All the steps are automated for wheel-rail contact analysis as discussed in Chapter 3. 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.

## B.5 Conformal contact

Conformal contact deals with curved contact patches, where the surface normal direction changes from one side of the contact patch to the other, see Figure 5.14. This arises when two conditions are met [55]:

1. the lateral radius of curvature $R_y$ is small in the contact area, such that the normal direction changes rapidly along the profile, and
2. the shapes of the two bodies are conforming, such that the contact is wide enough to pick up this change in normal direction.

In Figure 5.14, 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 if the flange root of the wheel has a radius $R_{wy} > -10.2$ or $> -10.5$ mm, depending on the total load on the contact [55].

The conformal contact approach works similarly to the planar approach described above:

1. Determine a suitable 'contact reference position', to be used a.o. for presenting aggregate outputs;

2. Determine the curved contact surface, and the corresponding normal and tangent directions, varying along the contact surface;
3. Determine the contact geometry (esp. the undeformed distance function) and kinematics (rigid slip velocity), relative to the curved contact surface;
4. Solve the normal and tangential contact problems, with elastic deformations computed according to the conformal shapes;
5. Present the results in terms of a global coordinate reference.

This conformal contact approach is described elegantly using generalised curvilinear coordinates $[\tilde{x}, \tilde{s}, \tilde{n}]^T$, with $\tilde{n} = 0$ in the contact plane.

All the steps are automated for wheel-rail contact analysis as discussed in Chapter 3. 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.

## B.6 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).

### B.6.1 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

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

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 $\mathsf{C}^{(a)}$ is a fourth-order stiffness tensor. The first equality of (B.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 $\partial\Omega^{(a)}$.

### B.6.2 Surface quantitities

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

$$\mathbf{p}^{(a)}(\mathbf{x}) = \sigma(\mathbf{x}) \cdot \mathbf{n}^{(a)}(\mathbf{x}), \tag{B.2}$$

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 $\mathbf{x}$:

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}^{(1)}(\mathbf{x}) - \mathbf{u}^{(2)}(\mathbf{x}). \tag{B.3}$$

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 $Oxy$,[1] such that $\mathbf{n}^{(a)} = [0, 0, (-1)^a]^T$, and concentrate on the surface points $\mathbf{x} = [x, y, 0]^T$. 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

$$p_n(x, y) = \mathbf{p}(\mathbf{x})^T \mathbf{n}, \quad \mathbf{p}_t(\mathbf{x}) = [p_x(x, y), p_y(x, y)]^T. \tag{B.4}$$

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

### B.6.3 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*.[1] The elasticity equations (B.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 displacements $\mathbf{u}$:

the tractions-displacement relation:

$$\mathbf{u}(\mathbf{x}, t) = \int_{\mathbf{x}' \in C} \mathbf{A}(\mathbf{x}, \mathbf{x}')\, \mathbf{p}(\mathbf{x}', t)\, dC \tag{B.5}$$

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 [17]:

[1] For simplicity we do not consider conformal contact situations here. For information on this consult the paper [64] that describes the extensions with respect to the half-space approach.

- 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.

### B.6.4 The contact conditions

Two important quantities of the contact problem are:

$$\text{normal problem :}$$
$$\text{deformed distance} \quad e := h + u_n \tag{B.6}$$
$$\text{tangential problem :}$$
$$\text{relative slip velocity} \quad \mathbf{s}_t := \mathbf{w}_t + \dot{\mathbf{u}}_t/V \tag{B.7}$$

The function $h(x, y)$ describes the profiles of the two bodies as well as the approach denoted by $\delta_n$. 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:

$$\text{normal problem :}$$
$$\text{in exterior } E: \quad e > 0,\ p_n = 0 \tag{B.8}$$
$$\text{in contact } C = H \cup S: \quad e = 0,\ p_n \geq 0 \tag{B.9}$$
$$\text{tangential problem :}$$
$$\text{in exterior } E: \quad \mathbf{s}_t \text{ free},\ \mathbf{p}_t = \mathbf{0}, \tag{B.10}$$
$$\text{in adhesion } H: \quad \|\mathbf{s}_t\| = 0,\ \|\mathbf{p}_t\| \leq g, \tag{B.11}$$
$$\text{in slip } S: \quad \|\mathbf{s}_t\| > 0,\ \mathbf{p}_t = -g\,\mathbf{s}_t/\|\mathbf{s}_t\|, \tag{B.12}$$
$$\text{Coulomb friction}: \quad g(\mathbf{x}, t) = \mu\, p_n(\mathbf{x}, t)\ . \tag{B.13}$$

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 (B.1) for bodies $a = 1, 2$. The deformations in the two bodies' interiors cannot be solved independently but are connected, via their overall position and motion $(\delta_n, h, \mathbf{w}_t)$, the deformation ($\mathbf{p}, \mathbf{u}$, equation (B.5)) and the precise conditions used at the mutual interface (equations (B.6)–(B.13)).

## B.7 Discretisation of the problem

In CONTACT, a calculation starts by defining a potential contact area that encompasses the true contact area. This potential contact area is discretised into $N = mx \cdot my$ rectangular elements of size $\delta x \cdot \delta y$ (see Figure 4.9). The surface tractions are approximated by piecewise constant functions per element.[2] This leads to

$$u_i(\mathbf{x}_I) = \sum_{j\in\{n,x,y\}} \sum_{J=1}^{N} A_{IiJj}(\mathbf{x}_I, \mathbf{x}_J)\, p_j(\mathbf{x}_J), \quad \text{for } I \in \{1 \cdots N\},\ i \in \{n, x, y\}. \tag{B.14}$$

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 (B.5) over a single element $J$ with respect to an observation point at $\mathbf{x}_I$, which can be done analytically [17]. 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. $\delta q$ is also called the 'time step' for brevity. The displacements at the previous time instance are denoted by $\mathbf{u}'$.

[2] Bilinear elements are provided also, with little benefit over a piecewise constant discretization [46, 72].

| | |
|---|---|
| $\mathbf{x}, \mathbf{y}$ | 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); |
| $\alpha$ | tangential coordinate direction: 2, 3 or $x$, $y$; |
| $\mathbf{p}(\mathbf{x})$ | surface traction (vector) acting on body 1 at position $\mathbf{x}$; |
| $\mathbf{A}(\mathbf{x}, \mathbf{x}')$ | $3 \times 3$ 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 (4.17)); |
| $\mathbf{u}_I$ | 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 $\alpha$ ($x$ or $y$) at center of element $I$; |
| $t, t', \delta t$ | time instances, time step; |
| $V$ | in rolling problems: the rolling speed, in shifts $V = 1$; |
| $\delta q$ | geometrical time step size: distance traversed per time step; |

Table B.1: *Overview of notations and conventions used in this document.*

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 $t$.

$$\mathbf{S}_{It} = \mathbf{W}_{It} + \mathbf{u}_{It} - \mathbf{u}'_{It}, \quad \text{for } I \in \{1 \cdots N\}. \tag{B.15}$$

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

## B.8 Fast or detailed solution

The full algorithms of CONTACT are relatively slow due to the dense matrix-vector products of Equation (B.14). This is cumbersome in case many time steps need to be solved, such as in a vehicle dynamics simulation. Approximate algorithms are provided for this situation.

For the normal contact problem, CONTACT provides the KPEC-method (Kik-Piotrowski with Ellipse Correction), a variant of the algorithm of Kik and Piotrowski [21, 33]. The shape of the contact area is approximated using the virtual interpenetration function. Different from Kik and Piotrowski's scaling, $\epsilon_{kp} = 0.55$, we use variable scaling $\epsilon_{kpec}(y)$ based on Hertz' theory, with exact agreement for elliptic contact patches. This provides a shape correction in a way that fits with the discretization used in CONTACT. The pressure distribution is approximated the same as in the KP-method, and is solved in the same way. An implementation of the ANALYN method is also provided [42]. For the tangential contact problem, CONTACT provides an implementation of Modified FASTSIM [43].

Using these fast algorithms, one may benefit from CONTACT's detailed contact search algorithms without jeopardizing the speed of the simulation. One can use fast algorithms during model building and one's day-to-day work, in regular cases. The full algorithms may be used once in a while, to check the validity of simplifications, for the final simulation, and in the most demanding situations.

## B.9 Specification of a case

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

- a coordinate system with positive $z$ ($n$) 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 B.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 $mx, my$ (Figure 4.9);
- in rolling problems: the rolling velocity $V$, direction $\chi$ ('`chi`') and 'time step size' $\delta q = V\,\delta t$. Note: recent modifications of the program more or less require that $\chi = 0°$ 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 $\delta_n$;
- either the approach $\delta_n$ 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 $\xi$ (`ksi`) and $\eta$ (`eta`) in $x$- and $y$-directions respectively, and the spin creepage $\phi$ (`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 switching to KPEC and Modified FASTSIM, the required solution accuracy, number of iterations required, and so on.

These quantities are easily recognized in the variables that are specified in the user input, which are described in detail in Chapter 4.

Some of these inputs are hard to obtain, like an appropriate contact reference plane in wheel/rail contact situations, and the corresponding undeformed distance and creepage situation. This is alleviated by the extensions of Chapter 3 for wheel/rail contact.