Vtech CMCC

Contact | Mechanics | Computing | Consulting

Bahialaan 100
3065 WC Rotterdam
The Netherlands

info@vtech-cmcc.nl

MEMO	EV/M22.002
Date	August 1, 2022
Author(s)	Edwin Vollebregt
Subject	Release-notes for CONTACT version 22.1

Summary

These release-notes document the changes in CONTACT version 22.1 with respect to version 21.1. The main changes are as follows:

- Added a ‘fast track’ through CONTACT using Kik-Piotrowski and Modified FASTSIM.
- Added new smoothing methods with better control over the level of smoothing;
- Reorganized and extended inputs for wheel and rail profiles and for profile placement.

This version has a number of compatibility breaks with respect to previous versions.

1 Fast and approximate contact solution

Background Over the years, the CONTACT library version for Matlab, Python, Fortran and C has grown into a powerful tool for wheel/rail contact in diverse programming situations. A complete

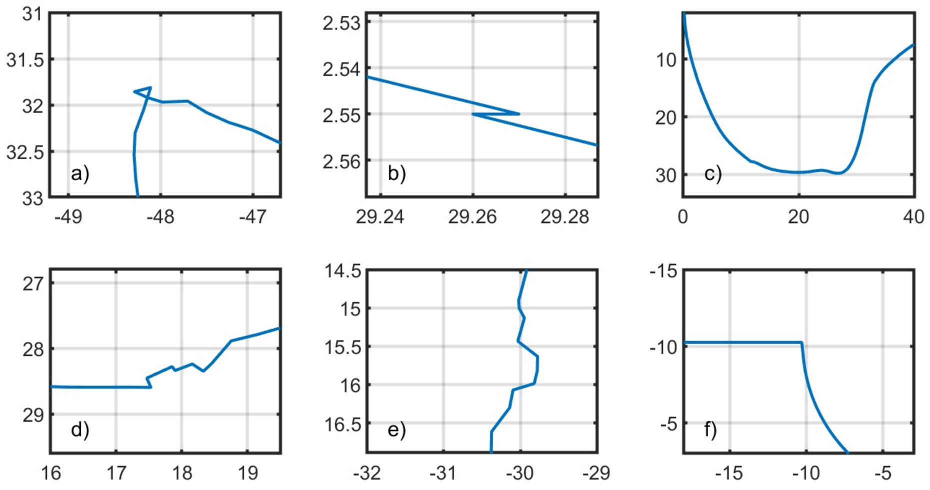

Figure 1: *Challenging features that may occur in wheel and rail input profiles: (a) self-crossing, (b) zig-zag pattern, (c) ‘dolphin nose’, (d) ‘roughness’, (e) ‘dent’, (f) ‘true kink’.*

API (Application Programming Interface) is provided that shields the user from the intricacies of the contact solution. Until now, however, this still required a separate effort from the user that needed to balance fast versus detailed computation, developing his own interfaces to simplified computations.

Two fast approximate methods are added to CONTACT for the computation of the contact shape and contact pressures. This completes the '**fast track**' through CONTACT, providing the option to use fast methods where possible and switch to detailed methods when needed, using the same library (interface) for all computations. This provides for convenient strategies such as quick turn-around time during model development, using fast methods in one's day-to-day work with occasional validation, or switching to the detailed solution in case of two-point contacts and other challenging configurations.

Fast non-Hertzian contact solution Wheel/rail contact analysis is generally split into three consecutive phases: (1) determining the contact location, (2) computing the contact patch shape and pressures, and (3) computing the tangential stresses arising from creepage [1, 2]. Little time is spent on the first of these stages, even when resolving the effects of the yaw angle in full [3]. The third stage can be made fast as well, switching to CONTACT's implementation of the Modified FASTSIM approach [4]. Our recent extensions concern fast algorithms for the second stage, removing the final bottle-neck for fast computation.

We have added two approximate non-Hertzian methods to CONTACT, that are sufficiently accurate most of the time, next to the existing methods that yield the detail needed in more complicated situations. Initially we aimed for the Extended Kik-Piotrowski method [5], that looked the most complete and fit for implementation in CONTACT. After extensive work we ended with two different methods, a new method called KPEC that builds on the Kik-Piotrowski approach [6] and a variant of the ANALYN method [7].

Both methods should be 'good enough' for the wheel/rail contact interaction as needed in multibody simulation. Particularly considering the uncertainties regarding the coefficient of friction, and the additional approximations introduced using FASTSIM for the tangential problem. Further tuning of the methods could be useful for wear and damage calculation.

Improvement of FASTSIM The 'fast track' provided by CONTACT employs the FASTSIM algorithm for the tangential problem solution. This is based on simplification, using a linear relationship between surface tractions (p_x, p_y) and corresponding, elastic displacements (u_x, u_y). We simply set $u_i = L_i p_i$, for well-chosen flexiblities L_i.

Improvements have been made to the flexibility values used in the version of FASTSIM that is implemented in CONTACT. This mainly addresses situations in which the creepages are large, that had not been covered well by Kalker's equations [8]. A blending approach is implemented that automatically switches from three to one flexibilities with increasing creepage. This roughly halves the error in the total forces compared to CONTACT's full computation.

Changes to the input The KPEC and ANALYN methods are implemented in CONTACT. They are activated setting the BOUND control digit to `B` = 5 or `B` = 6, as discussed in the User guide in

Section 2.3.2.

2 Improvements for demanding wheel/rail configurations

Curation of anomolous profile data Over time, a large number of profiles have been collected, illustrated in Figure 1, that are demanding for contact mechanics calculations. Additional checks are done and modifications made to improve the handling of such special cases. Optional input parameters are provided for the fine-tuning of these modifications. The new input for a rail profile now looks as follows:

```
if Z=3:         [ 'RFNAME'  MIRRORY  SCLFAC    SMOOTH  {MIRRORZ}  {PRFOPT} ]
if PRFOPT>=1: [  ISMOOTH  ZIGTHRS  KINKHIGH  KINKLOW  KINKWID ]
```

The flag MIRRORZ had been added in version 21.1. Curly braces indicate that the value is optional; the default is 0 if the value is not provided. Another optional value PRFOPT is now added for the detailed profile configuration. If given, a new line of the input is used for the parameters for smoothing, zig-zag detection, and kink detection. These are discussed further in Section 3.2 of the User guide.

New smoothing methods The smoothing spline that was provided by CONTACT [3] appeared difficult to use, in particular regarding the choice of the smoothing parameter λ. This is tied to the spacing δs that can vary between different profiles and also within a single profile. Furthermore, even smooth initial profiles are distorted if too agressive smoothing is used [9].

In our quest for the perfect smoother, we implemented three new smoothing routines [9].

1. The weighted smoothing spline with penalty on curvature values;
2. The weighted B-spline approach with penalty on curvature variation (beta);
3. The arc-fit approach, fitting circles to profile points within a sliding window.

Currently, these methods all behave similarly on various measured profiles with different characteristics. The main benefit then lies in that the new methods are easier to use than the original method. They act like low-pass filters that attenuate fluctuations with shorter wavelengths L while leaving longer wavelengths (mostly) unaffected. This is reflected in the input parameter L_{filt} that replaces the smoothing parameter λ.

Currently, the old smoothing spline remains the default method, for backward compatibility reasons. It is advised to adopt the weighted spline where possible. The default behavior will change in a future version.

Focus on a half-wheelset model Conceptually, CONTACT was focused on 'wheelset-on-track' configurations, with two wheels in a wheelset, and two rails in a track. This scope was supported in a half-hearted way, with little facilities for the equilibrium of total forces. The focus is changed in the current version. From now on, a single case revolves around a '**half wheelset on half track**', skipping the dynamic interactions with the other side. The user guide is restructured accordingly, expanding Chapter 3 on the workings of module 1.

This change of focus leads to modifications that break the compatibility with the previous version of CONTACT, as discussed in Section 4.

3 Resolved problems and general improvements

A number of smaller issues have been resolved, in module 1, module 3, and in the plot programs for Matlab. The main ones are as follows.

Licensing The licensing mechanism is extended to support Docker instances for instance on Amazon Web Services, using virtualization.

In version v21.1, there have been some instabilities with the network floating license, with the connection to the server being lost from time to time. This is resolved using a number of retries with increasing time between them, before giving up and aborting a run.

Subsurface stresses The speed of the subsurface stress calculation is improved for options ISUBS 2 to 7. The inspection of maximum values is simplified using the option **slc=‘max’** in function `plotstrs.m`.

4 Compatibility w.r.t. previous versions

This version of CONTACT produces the same results as the previous one except when using FASTSIM (`M` = 3), where the flexibilities are modified.

A number of changes are needed to user input files to change from the previous to the current release:

- The option to compute alternatingly left and right wheels with different profiles is no longer supported. An alternative for this is to use separate input files for left and right sides.
- The options $\texttt{E}_1 = 4, 6, 8$ and $\texttt{Z}_1 = 4$ are no longer supported.
- The options $\texttt{E}_1 = 5$ and 7 need to be changed into $\texttt{E}_1 = 4$ and 5, respectively.
- The inputs need to be changed for $\texttt{Z}_1 = 1$ and 3, concerning rail placement:

```
if GAUGHT >  0 :   [  GAUGHT    n.y.a.   GAUGWD   CANT     ]
if GAUGHT <= 0 :   [  GAUGHT    RAILY0   RAILZ0   CANT     ]
```

Changes to the output-file concern printing of information on the wheel and rail profiles used, and printing of the maximum pressure per case, for quick inspection of the results.

- The default behavior is changed for `parse_out3.m`, returning structs instead of arrays.

The format of the mat-file is unchanged with respect to the previous release.

A number of changes are needed to user programs calling the CONTACT library version.

- Four functions have been removed and may no longer be called, concerning separate profiles on left and right sides:

```
cntc_setrailprofilefname     ->  cntc_setprofileinputfname
cntc_setrailprofilevalues    ->  cntc_setprofileinputvalues
cntc_setwheelprofilefname    ->  cntc_setprofileinputfname
cntc_setwheelprofilevalues   ->  cntc_setprofileinputvalues
```

- For profile inputs, additional configuration parameters may be given (`iparam`, `rparam`):

```
cntc_setprofileinputfname   ->   ismooth, zigthrs, kinkhgh, etc.
cntc_setprofileinputvalues  ->       "       "        "
```

- The contents of `iparam` and `rparam` is changed in function `cntc_settrackdimensions`:

```
cntc_settrackdimensions   ->   cntc_settrackdimensions_old (old param)
                          ->   cntc_settrackdimensions_new (new param)
```

- The input `ewheel` is changed, removing 4, 6, 8 and shifting 5, 7:

```
cntc_setwheelsetdimensions   4 -> 3,  5 & 6 -> 4,  7 & 8 -> 5
cntc_setwheelposition          "            "            "
cntc_setwheelvelocity          "            "            "
cntc_setwheelflexibilities     "            "            "
```

Further changes are needed specifically in user programs developed in Matlab (matlab_intfc) or Python (python_intfc):

- Redundant input arguments have been deleted, such as the length of arrays being input, and the number of elements used. In Matlab, the calling signatures now look as follows:

```
function [ ifcver, ierror ] = cntc_initialize(ire, imodul, outpath, idebug)
function [ ] = cntc_setglobalflags         (params, values)
function [ ] = cntc_setflags               (ire, icp, params, values)
function [ ] = cntc_setmetadata            (ire, icp, params, values)
function [ ] = cntc_setsolverflags         (ire, icp, gdigit, iparam, rparam)
function [ ] = cntc_setmaterialparameters  (ire, icp, mdigit, rparam)
function [ ] = cntc_settemperaturedata     (ire, icp, imeth, params)
function [ ] = cntc_setfrictionmethod      (ire, icp, imeth, params)
function [ ] = cntc_setinterfaciallayer    (ire, icp, imeth, params)
function [ ] = cntc_settrackdimensions     (ire, ztrack, params)
function [ ] = cntc_setwheelsetdimensions  (ire, ewheel, params)
function [ ] = cntc_setwheelsetflexibility (ire, ewheel, params)
function [ ] = cntc_setwheelsetposition    (ire, ewheel, params)
function [ ] = cntc_setwheelsetvelocity    (ire, ewheel, params)
function [ ] = cntc_sethertzcontact        (ire, icp, ipotcn, params)
function [ ] = cntc_setpotcontact          (ire, icp, ipotcn, params)
function [ ] = cntc_setundeformeddistc     (ire, icp, ibase, prmudf)
function [ ] = cntc_setextrarigidslip      (ire, icp, wx, wy)
function [ eldiv      ] = cntc_getelementdivision (ire, icp)
function [ fld        ] = cntc_getfielddata       (ire, icp, ifld)
function [ sx, sy     ] = cntc_getmicroslip       (ire, icp)
function [ un, ux, uy ] = cntc_getdisplacements   (ire, icp)
function [ pn, px, py ] = cntc_gettractions       (ire, icp)
```

5 Known problems and restrictions

- plotting on rail/wheel surfaces doesn't account properly for $\Delta\phi_{rr}$;
- subsurface stresses are computed using elastic influence functions, also when the problem uses the viscoelastic material model.

References

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

[2] 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.

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

[4] 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.

[5] B. Liu, S. Bruni, and E.A.H. Vollebregt. A non-Hertzian method for solving wheel-rail normal contact problem taking into account the effect of yaw. *Vehicle System Dynamics*, 54(9):1226–1246, 2016.

[6] 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.

[7] 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.

[8] E.A.H. Vollebregt and P. Voltr. Improved accuracy for FASTSIM using one or three flexibility values. *Vehicle System Dynamics*, 2022. Open access.

[9] 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*, Australia, 2022. Monash University. Open access.