Journal of Computational Physics 288 (2015) 86–100

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A fast nonlinear conjugate gradient based method for 3D concentrated frictional contact problems

Jing Zhao [a,*], Edwin A.H. Vollebregt [a,b], Cornelis W. Oosterlee [a,c]

[a] *Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628CD Delft, The Netherlands*
[b] *VORtech BV, 2600AG Delft, The Netherlands*
[c] *CWI – Center for Mathematics and Computer Science, 1090GB Amsterdam, The Netherlands*

ARTICLE INFO

Article history:
Received 25 June 2014
Received in revised form 14 January 2015
Accepted 13 February 2015
Available online 19 February 2015

Keywords:
Frictional contact problem
Azimuth angle variables
Active set strategy
Nonlinear conjugate gradient
Preconditioner
Fast Fourier transform

ABSTRACT

This paper presents a fast numerical solver for a nonlinear constrained optimization problem, arising from 3D concentrated frictional shift and rolling contact problems with dry Coulomb friction. The solver combines an active set strategy with a nonlinear conjugate gradient method. One novelty is to consider the tractions of each slip element in a polar coordinate system, using azimuth angles as variables instead of conventional traction variables. The new variables are scaled by the diagonal of the underlying Jacobian. The fast Fourier transform (FFT) technique accelerates all matrix–vector products encountered, exploiting the matrix' Toeplitz structure. Numerical tests demonstrate a significant reduction of the computational time compared to existing solvers for concentrated contact problems.

1. Introduction

1.1. Physics

The frictional contact problem has attracted interest from many researchers, due to various applications in the industry and engineering fields, e.g. rolling contact fatigue (RCF) [10], the fatigue life of machine elements [31], friction and wear [7,29,13]. This problem concerns two elastic bodies. When they are pressed together, the forces they obtain from each other result in elastic deformation. This yields a contact area where the surfaces of the two bodies coincide, and exert stresses on each other. These stresses are composed of normal stress (pressure), and the frictional stress (traction) acting in the tangential direction. When and where the frictional stress is small, the two bodies stick to each other. However, local sliding occurs where the frictional stress is large enough. The challenge is to find the distribution of the frictional stress, and the subdivision of the contact area: which part is an adhesion area and in which part does slip occur.

The model for frictional contact starts with a known contact area and pressure distribution. Then the frictional stress should satisfy:

1. In the adhesion area, the magnitude of the tractions does not exceed the traction bound, and there is no slip.
2. In the slip area, the traction bound is reached, and the resulting slip points in the opposite direction of the tractions.

[*] Corresponding author.
E-mail address: J.zhao-1@tudelft.nl (J. Zhao).

http://dx.doi.org/10.1016/j.jcp.2015.02.016
0021-9991/© 2015 Elsevier Inc. All rights reserved.

The traction bound comes from the frictional law that is used. For this one may take Coulomb's law locally, which states that the traction bound equals the product of the normal pressure and a friction coefficient. The magnitude of the tangential tractions should be less or equal to the traction bound. On the one hand, this gives rise to inequality constraints. On the other hand, when slip occurs, equality should hold, and the directions of the tangential tractions and the resulting slip should be opposite. This brings in nonlinearity.

1.2. Solution strategies

The classic solutions to frictional contact problems with partial sliding stem from the work by Cattaneo [4] and Mindlin [23]. In the last decades, other solution techniques have been studied, for example Johnson [14] and Kalker [17] contributed with fundamental work.

The numerical solution techniques employ variational inequalities [14,17,38]. They are generally divided into two classes. One is the class of finite element methods (FEM) [18,38,20,6,2,12], that are widely used, especially in the case of large deformations, and nonlinear elastic materials. These methods typically focus on overall behavior. Due to the discretization of the contacting bodies, this method can be computationally expensive. The other class is the boundary element methods (BEM) [17,1,22,19], that are well-suited for "concentrated contact" and efficient for homogeneous elastic problems. The boundary value problem is transformed to a boundary integral equation. The dimensionality of the problem decreases, i.e., the 3D contact problem is solved by considering 2D contact regions where only the boundary is discretized. Hence, this method reduces the computational time significantly.

When the contacting bodies are of different materials, the tangential tractions and normal displacements interact with each other. In this case, the normal and tangential problems cannot be easily separated. A straightforward way to process it is to solve a fully coupled formulation [37]. Another popular approach is via the so-called "Panagiotopoulos process" [25,9,17]. In each iteration, the normal problem is solved first followed by the tangential problem. When contacting bodies are of the same material, i.e. a so-called quasi-identity case, these two problems can be decoupled, and one iteration is sufficient [17].

1.3. Solution algorithms

Kalker's variational approach [17], which is a prominent method for the rolling contact problem, employs Green's function for the elastic half-space. This is a BEM, where Coulomb's law is applied. A fine discretization is used inside the contact area. Dense matrices need to be solved for elements in the contact area.

As a solution algorithm, the TANG algorithm was proposed in [16]. It applies an active set strategy [24], which leads to systems of nonlinear equations that are solved using Newton's method and Gauss-elimination (GE). This approach has $\mathcal{O}(n^{3.5})$ complexity, with n the number of contact elements. Another method is the ConvexGS method [32]. It reduces the global problem to a small-sized optimization problem on each element, and solves by a block Gauss–Seidel iteration. This method is incorporated into the software CONTACT [35]. However, the Gauss–Seidel process is also relatively slow for fine discretizations with a complexity of about $\mathcal{O}(n^{2.3})$ [33].

Different from BEM, FEM is based on a large number of elements covering whole contacting bodies, while much fewer elements are placed in the contact area. Sparse matrices are solved, but the size of the matrices is much larger than the dense matrices involved in BEM. Algorithms include the penalty approach, the augmented Lagrangian technique, etc. Comparing with the BEM methods, we encounter similar approaches for the nonlinear equations, like Newton-based methods [38], or a nonlinear Gauss–Seidel method [15] that is similar to ConvexGS.

1.4. A new solution method

The motivation of our work is to develop a fast solver for the 3D frictional contact problem, especially for the so-called shift problem, e.g. the Cattaneo shift [4]. It is a transient contact problem, and concerns one object pressed onto another, and shifted tangentially. It plays an important role in the study of rolling contact problems, since there can be generally a sequence of shift problems. In this paper we consider the tangential problem, with the solutions from the normal problem already available. It can be easily incorporated into the Panagiotopoulos process, to deal with more complicated contact.

Our new method contributes to the BEM solvers. We call it "TangCG", since it searches for the tangential tractions and is based on the nonlinear conjugate gradient method. The constraint that the magnitude of tractions on each slip element should equal the traction bound, inspires to place the traction vector at a circle in a polar coordinate system, with the radius being the traction bound. We use azimuth angles as variables in the slip area, which is a significant difference from conventional solvers.

The TangCG algorithm is a so-called bound-constrained conjugate gradient (BCCG) method, which was proposed for linear complementarity problems in normal contact [34]. The BCCG method uses an active set strategy, and employs the conjugate gradient (CG) method for the governing linear system. Differently, the governing system in frictional contact problems is mainly nonlinear, hence, we employ a nonlinear conjugate gradient (NLCG) method [28]. The TangCG algorithm is combined with a diagonal scaling preconditioner.

The most time consuming part of the solution procedure relates to a Boussinesq integral [14,17], which gives the relation between tractions and deformation. It results in a dense, symmetric and positive definite coefficient matrix. This matrix is block Toeplitz matrix with Toeplitz blocks (BTTB).[1] Such structure can be exploited by fast Fourier transform (FFT), which has been applied in the field of contact problems, e.g. [30,36,5]. The complexity is reduced by FFT to $\mathcal{O}(n\log n)$ with n the number of unknowns.

1.5. Structure of this paper

Section 2 gives the mathematical formulation of the 3D frictional contact problem and explains existing solvers. The TangCG method is illustrated and numerically analyzed in Section 3. Section 4 shows the numerical results of our new method, and also compares the efficiency with an existing method. Section 5 concludes.

2. Formulation of the 3D frictional contact problem

This section introduces the physical problem and the mathematical model of frictional contact. For more details, we refer to [17]. Subsequently, the discretization and existing methods for the model are given.

2.1. Notations

We specify some notations that will be used. The element index is I or J, which often appears as subscript. Sub- or super-scripts x and y indicate x- and y-directions, respectively. Letters in bold represent vectors. The tractions on element I are defined by $\mathbf{p}_I = [p_{Ix}, p_{Iy}]$, including two traction components in the x- and y-directions. Without the subscript I, e.g. $\mathbf{u}$ is defined by $\mathbf{u} = [\mathbf{u}_x, \mathbf{u}_y]$. Here, $\mathbf{u}_x = [u_{1x}, u_{2x}, \ldots]$ indicates the x-direction deformation of all elements, and a similar definition applies on $\mathbf{u}_y$.

Vectors $\bar{\mathbf{s}}_I$ and $\mathbf{s}_I$ are slip velocity and distance, respectively, on element I. For the use in Section 3, the *primary variables* on an adhesion element have two entries for x- and y-directions, but we use only one entry on each slip element, carrying the information about angles. They are denoted by subscript a, while subscript s denotes *auxiliary variables*, that have two entries for x- and y-directions on each adhesion and slip element. Superscript k is an iteration index.

2.2. Mathematical model of the frictional contact problem

First of all, we identify the geometry of two elastic bodies in undeformed state. A coordinate system $Oxyz$ is used on each particle of these two bodies to identify its position $\mathbf{x}$. This coordinate system is assumed to move with the contact region, with z pointing normally to the other body and x, y directing tangentially. Deformation occurs when the bodies are brought into contact. An elastic field including stress σ, strain ϵ and displacements $\mathbf{u}$ arises in the bodies and on their surfaces. The formulation of contact problems particularly focuses on the surface quantities: surface displacements $\mathbf{u}^{(i)}(\mathbf{x})$ of body i $(i = 1, 2)$ at position $\mathbf{x}$, and surface tractions $\mathbf{p}^{(i)}(\mathbf{x})$. The deformation at position $\mathbf{x}$ is defined by the displacement difference, i.e., $\mathbf{u}(\mathbf{x}) = \mathbf{u}^{(1)}(\mathbf{x}) - \mathbf{u}^{(2)}(\mathbf{x})$. Moreover, we solve for only tractions $\mathbf{p}^{(1)}$, since the relation $\mathbf{p}^{(1)} = -\mathbf{p}^{(2)}$ is valid for all particles on the surfaces.

The normal problem can be solved by Hertz theory [11] if the contacting surfaces are smooth and quadratically or by a numerical method (e.g. [35]) if the profile is non-Hertzian. The solution pressure is compressive in the contact area, and vanishes outside of it.

Tangential tractions result from the overall motion of the two bodies, causing a tendency of the surfaces to slip with respect to each other, called “rigid slip”. In rolling contact it is often characterized via “creepage”, which is the average relative velocity between the surfaces, and is given by the difference between the forward rolling velocity and the circumferential velocity.

The formulation concerns three aspects, i.e., slip, elastic deformation, and the friction between the two contacting bodies. First of all, the relation between slip, rigid slip and deformation is defined by [35]:

$$\bar{\mathbf{s}} := \bar{\mathbf{w}} + \frac{1}{V}\frac{D\mathbf{u}}{Dt}, \tag{1}$$

where, $\bar{\mathbf{s}}$ is the slip velocity of two opposing particles on the contacting surfaces with respect to each other. It is the summation of the relative rigid slip velocity $\bar{\mathbf{w}}$ and the change of deformation $\frac{D\mathbf{u}}{Dt}$ scaled by the rolling velocity V.

Concerning the effect of friction, the contact area C is divided into an adhesion area H and a slip area S, according to the contact conditions:

$$\begin{cases} \text{in the adhesion area} \quad H(t): \quad ||\bar{\mathbf{s}}(\mathbf{x},t)|| = 0, \quad ||\mathbf{p}(\mathbf{x},t)|| \le g(\mathbf{x},t), \\ \text{in the slip area} \quad S(t): \quad ||\bar{\mathbf{s}}(\mathbf{x},t)|| > 0, \quad \mathbf{p}(\mathbf{x},t) = -g(\mathbf{x},t)\frac{\bar{\mathbf{s}}(\mathbf{x},t)}{||\bar{\mathbf{s}}(\mathbf{x},t)||}. \end{cases} \tag{2}$$

[1] Toeplitz matrix is a matrix with constant diagonals.

The norm $||\cdot||$ is the Euclidean norm. These conditions state that the magnitude of the tangential tractions at position $\mathbf{x}$ cannot exceed the corresponding traction bound g in the adhesion area. When the traction bound is reached, local slip occurs with its direction opposite to the tractions.

The traction bound g in this model is obtained by applying Coulomb's frictional law locally

$$g(\mathbf{x},t)=\mu p_n(\mathbf{x},t), \tag{3}$$

where μ is a constant friction coefficient, and p_n is the normal pressure.

The deformation $\mathbf{u}$ in Eq. (1) is obtained by the *half-space* approach, which is based on four simplifying assumptions [17]. Firstly, the contacting bodies are made of homogeneous linear elastic material. Secondly, the contact area is very small compared with the contacting bodies and hence, it is *flat*. The geometry is assumed to be sufficiently smooth, and we ignore the effects of inertial force, which is a quasi-static approach.

Based on the classic solutions by Boussinesq and Cerruti (see Johnson [14] or Kalker [17]), the relation between the deformation $\mathbf{u}$ and traction $\mathbf{p}$ is given by:

$$\mathbf{u}(\mathbf{x},t)=\int\limits_{C(t)} A(\mathbf{x},\mathbf{x}')\mathbf{p}(\mathbf{x}',t)dx'. \tag{4}$$

This integral states that at a certain time instance t, the deformation at point $\mathbf{x}$ is influenced by the tractions on all points $\mathbf{x}'\in C$. Here, the kernel function $A(\mathbf{x},\mathbf{x}')$ gives the influence of a unit traction at $\mathbf{x}'$ on the deformation at $\mathbf{x}$. It can be written as:

$$A(\mathbf{x},\mathbf{x}')=A(\mathbf{x}-\mathbf{x}'), \tag{5}$$

which indicates that the influence coefficient depends on the relative positions of two surface points $\mathbf{x}$ and $\mathbf{x}'$. This property leads to a specially structured matrix after discretization.

2.3. Discretization

The discretization focuses on a rectangular potential contact area, which contains the true contact area. A rectangular mesh is placed on this 2D region, with $N=n_x\times n_y$ elements of size $\delta x\times\delta y$. The coordinates of the center of the element I are denoted by $[x_I, y_I]$.

Tractions $\mathbf{p}$ in the surface integral (4) are approximated by element-wise constant functions. The cell-centered discretization of the integral (4) yields, for each contacting element I:

$$u_{Ix}=\sum_{J=1}^{N}(A_{IJ}^{xx}p_{Jx}+A_{IJ}^{xy}p_{Jy}),\quad u_{Iy}=\sum_{J=1}^{N}(A_{IJ}^{yx}p_{Jx}+A_{IJ}^{yy}p_{Jy}), \tag{6}$$

where, u_{Ix} is the deformation in x-direction of element I. Influence coefficient A_{IJ}^{xy} represents the influence to the x-direction deformation on element I, caused by a unit y-direction traction on another element J. It is computed by integrating (4) over a single element J, with respect to an observation point at the center of element I (see [17] for the detailed formulas).

Writing (6) in matrix form, we obtain:

$$\mathbf{u}=A\mathbf{p},\quad A\in\mathbb{R}^{2n\times 2n},\quad \mathbf{u},\mathbf{p}\in\mathbb{R}^{2n}, \tag{7}$$

where, the global coefficient matrix A has 2×2 blocks:

$$A=\begin{pmatrix} A^{xx} & A^{xy}\\ A^{yx} & A^{yy}\end{pmatrix}, \tag{8}$$

with each block $A^{xx}, A^{xy}, A^{yx}, A^{yy}\in\mathbb{R}^{n\times n}$. These blocks are dense, symmetric, positive and definite (SPD). Moreover, due to property (5), these blocks are BTTB if the contact area is rectangular.[2]

To discretize slip Eq. (1), a sequence of time instances is used, with length $\delta t=t-t'$, where t and t' are the current and previous time points, respectively. Applying a backward Euler method for the derivative, we obtain:

$$\bar{\mathbf{s}}_I=\bar{\mathbf{w}}_I+\frac{\mathbf{u}_I-\mathbf{u}'_I}{\delta t V}. \tag{9}$$

Define the traversed distance per time step as $\delta q=V\cdot\delta t$, which is called time step for short. The shift on one element $\mathbf{s}_I:=\delta q\cdot\bar{\mathbf{s}}_I$ represents the aggregated slip distances over δq. Then, discretized Eq. (9) reads:

$$\mathbf{s}_I=\mathbf{w}_I+(\mathbf{u}_I-\mathbf{u}'_I),\quad 1\le I\le N, \tag{10}$$

[2] Otherwise, they are not BTTB anymore. An approach of applying FFTs in this case is given in [36].

where rigid shift $\mathbf{w}_I$ is defined as:

$$\mathbf{w}_I = [\xi - \phi y_I, \eta + \phi x_I]. \tag{11}$$

Here ξ, η are longitudinal and lateral shifts, respectively, and ϕ is the rotation shift [35]. Since the deformation at a previous time $\mathbf{u}'_I$ vanishes in a shift problem, Eq. (10) yields:

$$\mathbf{s}_I = \mathbf{w}_I + \mathbf{u}_I. \tag{12}$$

After discretization, condition (2) becomes:

$$\begin{cases} \text{in the adhesion area} \quad H\colon \quad ||\mathbf{s}_I|| = 0, \quad ||\mathbf{p}_I|| \le g_I, \\ \text{in the slip area} \quad S\colon \quad ||\mathbf{s}_I|| > 0, \quad \mathbf{p}_I = -g_I \cdot \mathbf{s}_I / ||\mathbf{s}_I||. \end{cases} \tag{13}$$

In the solution procedure, these two conditions are equivalent to solving the following equations:

- For adhesion element $I \in H$:

$$\mathbf{s}_I = \mathbf{0}. \tag{14}$$

- For slip element $I \in S$:

$$p_{Ix}^2 + p_{Iy}^2 = g_I, \tag{15}$$

$$p_{Ix}s_{Iy} - p_{Iy}s_{Ix} = 0, \tag{16}$$

where Eq. (16) is derived from the fact that slip $\mathbf{s}_I$ and traction $\mathbf{p}_I$ are in opposite directions, and hence, $\mathbf{p}_I^\perp \perp \mathbf{s}_I$ with $\mathbf{p}_I^\perp = [-p_{Iy}, p_{Ix}]$. Eqs. (15) and (16) give rise to nonlinearity.

The frictional contact problem now consists of Eqs. (7), (12) and (13). It was shown that this problem can also be considered as a variational inequality [17]:

$$\min_{\mathbf{p}} \; \frac{1}{2}\mathbf{p}^T A\mathbf{p} + \mathbf{w}^T\mathbf{p}, \quad \text{s.t.} \quad ||\mathbf{p}_I|| \le g_I \text{ for } I \in C. \tag{17}$$

This is a nonlinear convex optimization problem [3], for which, the corresponding Karush–Kuhn–Tucker (KKT) conditions [21] provide both sufficient and necessary conditions for the existence of a unique solution. The KKT conditions also result in Eqs. (7), (12), and (13).

2.4. Existing algorithms

There are essentially two existing methods based on the above formulation. One is Kalker's “TANG” algorithm [16,17]. It is an active set method and consists of inner and outer iterations. In each outer iteration it fixes the subdivision of the contact area, which determines the corresponding governing systems (14)–(16). The resulting solution is used to modify the subdivision according to contact condition (13). Newton's method is applied for the nonlinear system, and the linearized system in every inner iteration is solved by Gauss-elimination.

The limitation of this method shows when the problem size is large. Gauss-elimination, with complexity $\mathcal{O}(n^3)$, is time-consuming and memory-taking.

An alternative is the “ConvexGS” algorithm [32], which is a block Gauss–Seidel iteration. The elements in contact are processed one by one, with the recently updated tractions fixed. The subproblem defined on each element is derived based on the element-wise constraints in the convex optimization formulation (17). The corresponding nonlinear system (15)–(16) for the element is solved by Newton's method.

However, this method cannot benefit from an implementation using FFTs exploiting the BTTB structure.

3. New method: TangCG algorithm

We present our new algorithm TangCG for frictional contact in this section. Its main components are introduced in Section 3.1. Section 3.2 illustrates this method based on a small test, for additional insight. Section 3.3 gives the resulting algorithm.

3.1. Main components of the TangCG algorithm

The TangCG algorithm consists of six main components, shown in Fig. 1, where the adjacent pieces are related to each other.

The TangCG method adopts the framework of an active set strategy from the BCCG method [34]. In each outer iteration, the adhesion and slip areas are fixed and systems (14)–(16) are solved approximately; then we modify the subdivision

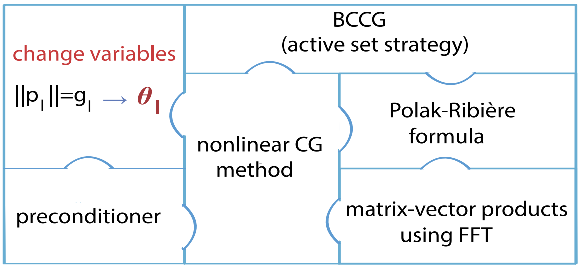

Fig. 1. The main components of the TangCG method. The adjacent pieces are related to each other.

of contact area, according to condition (13). A reduced system of equations is considered in inner iterations, where the constraints for elements in the slip area are automatically enforced.

The most pronounced component of the TangCG algorithm is as follows. Different from conventional methods that use traction variables, we change variables in the slip area, where the magnitude of tractions should equal the traction bound. On each slip element I, traction $\mathbf{p}_I$ lies at a circle with radius g_I. Therefore, we can define $\mathbf{p}_I$ in a polar coordinate system with its two components p_{Ix} and p_{Iy} along the horizontal and vertical axes, respectively. Since g_I is known, $\mathbf{p}_I$ can be uniquely determined by the azimuth angle θ_I as: $\mathbf{p}_I = g_I \cdot [\cos(\theta_I), \sin(\theta_I)]$. Thus, Eq. (15) is automatically satisfied. Only Eq. (16) is to be solved on each slip element.

For the resulting nonlinear system, TangCG employs a nonlinear conjugate gradient (NLCG) method. In each NLCG iteration, the system is linearized, and a CG process is applied. Concerning the construction of the search directions, the conjugacy may be lost as the iterations proceed. The Polak–Ribière formula [26] is in our case the preferable choice compared to the Fletcher–Reeves formula [8] to improve the search directions: the former directs the search direction automatically towards the steepest descent direction when the residual in the previous iteration is orthogonal to the new search direction, while the latter restarts the solution process in this case [27].

Preconditioning is done using the diagonal of the reduced system's Jacobian matrix. This preconditioner is necessary for the TangCG method, which employs different variables in adhesion and slip areas. These quantities are brought to the same measure by this preconditioner. Without preconditioner the solver may largely disregard the residual in either the adhesion or the slip area, depending on the material parameters.

Since the TangCG algorithm avoids generating Jacobian matrices, the matrix–vector products (MVPs) encountered only depend on the influence coefficient matrix A in (7), which includes submatrices being block Toeplitz with Toeplitz blocks (BTTB). The fast Fourier transform (FFT) technique is applied to accelerate such MVPs with complexity $\mathcal{O}(n \log n)$.

3.2. Illustration of the TangCG algorithm

This subsection aims at showing how the pieces shown in Fig. 1 are combined to result in an efficient algorithm. A very small test case is considered. It consists of only two elements in contact, with elements 1 and 2 in the adhesion and slip areas, respectively. The traction bounds g_I, $I = 1, 2$ are given.

3.2.1. Change of variables in slip area

Let's start with a prominent component of the method, which is the change of variables in the slip area. The following discussion is for the kth TangCG iteration, with the tractions $\mathbf{p}^{k-1}$ and slip $\mathbf{s}^{k-1}$ known from the previous iteration. For clarity of the equations, we avoid the iteration indices $k-1$ and k where possible.

When the slip area S is empty, i.e. in the full adhesion case, the governing system $\mathbf{s} = A\mathbf{p} + \mathbf{w} = \mathbf{0}$ is linear. When the slip area S is not empty, it becomes nonlinear. Considering the two-element test, the governing system reads:

$$\mathbf{F}_f = \mathbf{0}, \tag{18}$$

with

$$F_1 = s_{1x}, \tag{19}$$

$$F_2 = s_{1y}, \tag{20}$$

$$F_3 = p_{2x}^2 + p_{2y}^2 - g_2^2, \tag{21}$$

$$F_4 = p_{2x}s_{2y} - p_{2y}s_{2x}, \tag{22}$$

where the first two equations correspond to adhesion element 1. Eqs. (21) and (22) for slip element 2 bring in nonlinearity, and TangCG applies an NLCG method. The nonlinear system (18) is linearized by a truncated Taylor expansion, which yields:

$$J_f \cdot \delta\mathbf{p}_f = -\mathbf{F}_f(\mathbf{p}^{k-1}), \tag{23}$$

with Jacobian matrix $J_f = \nabla\mathbf{F}_f(\mathbf{p}^{k-1})$ having the following form:

$$J_f(\mathbf{p}^{k-1}) = \begin{pmatrix} A_{11}^{xx} & A_{11}^{xy} & A_{12}^{xx} & A_{12}^{xy} \\ A_{11}^{yx} & A_{11}^{yy} & A_{12}^{yx} & A_{12}^{yy} \\ 0 & 0 & p_{2x} & p_{2y} \\ a_1 & a_2 & a_3 & a_4 \end{pmatrix}, \tag{24}$$

and:

$$a_1 = p_{2x}A_{21}^{yx} - p_{2y}A_{21}^{xx}, \qquad a_2 = p_{2x}A_{21}^{yy} - p_{2y}A_{21}^{xy},$$
$$a_3 = p_{2x}A_{22}^{yx} - p_{2y}A_{22}^{xx} + s_{2y}, \quad a_4 = p_{2x}A_{22}^{yy} - p_{2y}A_{22}^{xy} - s_{2x}.$$

This Jacobian matrix corresponds to variables $\delta\mathbf{p}_f = [\delta p_{1x}, \delta p_{1y}, \delta p_{2x}, \delta p_{2y}]^T$. It is not symmetric and thus the CG method cannot be applied.

A change of variables in the slip area can overcome this difficulty. Since traction $\mathbf{p}_2$ is placed at a circle of radius g_2 (e.g. see Fig. 2(a)), the nonlinear Eq. (21) is satisfied automatically and can be eliminated. The linearized version of this equation gives:

$$\begin{aligned} p_{2x}\delta p_{2x} + p_{2y}\delta p_{2y} &= -(p_{2x}^2 + p_{2y}^2 - g_2^2) \\ &= 0 \text{ (since assuming } ||\mathbf{p}_2|| = g_2), \end{aligned} \tag{25}$$

which indicates $\delta\mathbf{p}_2 \perp \mathbf{p}_2$. It further yields:

$$\delta p_{2x} = -p_{2y}\delta\theta_2, \quad \delta p_{2y} = p_{2x}\delta\theta_2. \tag{26}$$

This relation satisfies $||\delta\mathbf{p}_2|| = ||\mathbf{p}_2||\delta\theta_2$, which is the definition of the arc length corresponding to the angle $\delta\theta_2$. This arc length is approximated by vector $\delta\mathbf{p}_2$ which is orthogonal to $\mathbf{p}_2$.

3.2.2. Reduced system

Eliminating Eq. (21) yields a reduced system written as:

$$\mathbf{F}_r = \mathbf{0}, \tag{27}$$

and the linearized system with relation (26) is given by:

$$J_r\delta\mathbf{p}_r = -\mathbf{F}_r, \tag{28}$$

with a 3 × 3 Jacobian matrix:

$$J_r = \begin{pmatrix} A_{11}^{xx} & A_{11}^{xy} & a_1 \\ A_{11}^{yx} & A_{11}^{yy} & a_2 \\ a_1 & a_2 & a_4 p_{2x} - a_3 p_{2y} \end{pmatrix}. \tag{29}$$

The corresponding variables are $\delta\mathbf{p}_r = [\delta p_{1x}, \delta p_{1y}, \delta\theta_2]^T$ satisfying

$$\delta\mathbf{p}_f = T\delta\mathbf{p}_r, \tag{30}$$

with transformation matrix T defined by:

$$T = \begin{pmatrix} 1 & & \\ & 1 & \\ & & -p_{2y} \\ & & p_{2x} \end{pmatrix}, \tag{31}$$

whose transpose can convert $\delta\mathbf{p}_f$ to $\delta\mathbf{p}_r$. The reduced Jacobian matrix (29) is symmetric and positive definite (SPD), so we can apply CG to it.

3.2.3. Solution method for the reduced system

The reduced system (27) is solved by an NLCG method. In each of its iterations, the CG method is applied to the linearized system (28), with initial iterate $\delta\mathbf{p}_r^0 = \mathbf{0}$.

First of all, we compute the residual of linearized system (28) by:

$$\begin{aligned} \mathbf{r}_a &= -\mathbf{F}_r - J_r \cdot \delta\mathbf{p}^0, \\ &= -\mathbf{F}_r, \quad \text{(because } \delta\mathbf{p}_r^0 = \mathbf{0}), \end{aligned} \tag{32}$$

which is equal to the residual of the nonlinear system (27). The search direction $\mathbf{v}_a$ adopts the steepest descent direction when the area subdivision changed. Otherwise, it employs the Polak–Ribière formula. The line search aims at determining a step length α, such that the residual resulting from the updated iterate is orthogonal to the current search direction $\mathbf{v}_a$. This yields a formula for α given by:

$$\alpha = \frac{(\mathbf{r}_a, \mathbf{v}_a)}{(\mathbf{v}_a, \mathbf{q}_a)}, \tag{33}$$

where $\mathbf{q}_a$ is the change of the residual caused by $\mathbf{v}_a$, defined by:

$$\mathbf{q}_a = J_r \mathbf{v}_a. \tag{34}$$

Subsequently, the iterate of linear system (28) is corrected by $\delta\mathbf{p}_r^1 = \alpha\mathbf{v}_a$. This further gives:

$$T\delta\mathbf{p}_r^1 = \alpha T\mathbf{v}_a \quad \Rightarrow \quad \delta\mathbf{p}_f^1 = \alpha \cdot \mathbf{v}_s, \tag{35}$$

where $\mathbf{v}_s$ is the auxiliary search direction defined by:

$$\mathbf{v}_s = T\mathbf{v}_a. \tag{36}$$

The new traction iterate is given by:

$$\mathbf{p}^k = \mathbf{p}^{k-1} + \delta\mathbf{p}_f^1 = \mathbf{p}^{k-1} + \alpha \cdot \mathbf{v}_s^k. \tag{37}$$

Moreover, the tractions on each slip element need to be projected onto the corresponding circle. This represents one NLCG iteration.

3.2.4. Matrix–vector products

To compute $\mathbf{q}_a$ in (34), TangCG does not directly implement this multiplication of Jacobian matrix J_r and vector $\mathbf{v}_a$, but takes a detour, which allows the use of FFTs. Vector $\mathbf{v}_a$ is transformed to the auxiliary search direction $\mathbf{v}_s$ by (36). Then its influence on slip $\delta\mathbf{s}$ can be computed by

$$\delta\mathbf{s} = A\mathbf{v}_s, \tag{38}$$

where the FFT technique can be applied due to the matrix A in (8). Eq. (34) with relation (26) yields:

$$\mathbf{q}_a = \begin{pmatrix} \delta s_{1x} \\ \delta s_{1y} \\ p_{2x}\delta s_{2y} - p_{2y}\delta s_{2x} + s_{2y}v_{2x} - s_{2x}v_{2y} \end{pmatrix}. \tag{39}$$

This indicates that on adhesion element 1 we have $\delta\mathbf{s}_1 = \mathbf{q}_{a1}$, and on slip element 2, $\delta\mathbf{s}_2$ also has its influence as seen in the last formula in (39).

3.2.5. Active set framework from BCCG(K)

An outer iteration is processed after every K inner iterations. It validates the contact conditions, and adjusts the subdivision of the contact area. The adhesion elements with $||\mathbf{p}_I|| > g_I$ are moved to the slip area, and are projected to $||\mathbf{p}_I|| = g_I$. Remember that tractions $\mathbf{p}_I$ of each slip element should have opposite directions with the slip $\mathbf{s}_I$. The elements are moved to the adhesion area if these two vectors lie in the same half plane. Finally, the tolerance of the solution is checked.

3.2.6. Illustration from view by angles

Since TangCG uses angle variables in the slip area, it can be explained from the view of angles as well. We define the anti-clockwise angle to be positive. Fig. 2 shows the kth TangCG iteration on slip element I. (We drop the superscripts $k-1$ and k in this figure.) In this discussion, we will use six angles $\tau_i, i \in \{1, 2, 3, 4, 5, 6\}$.

In Fig. 2(a), the traction $\mathbf{p}_I$ lies at the circle with radius g_I. It is expected that

$$\theta_p = \theta_s + \pi, \tag{40}$$

i.e., the azimuth angle of slip $\mathbf{s}_I$ plus π is equal to azimuth angle of $\mathbf{p}_I$. We notice that $\tau_2 = \theta_s + \pi - \theta_p$ is the residual of Eq. (40); τ_2 represents the angle along which $\mathbf{p}_I$ should move, in order to be located opposite to $\mathbf{s}_I$. However, a change in $\mathbf{p}_I$ also causes a change in $\mathbf{s}_I$. So τ_2 is not the exact angle that $\mathbf{p}_I$ needs to be changed with.

The angle residual computed by Eq. (32) is given by:

$$\begin{aligned} r_{aI} &= -(p_{Ix}s_{Iy} - p_{Iy}s_{Ix}) \\ &= ||-\mathbf{p}_I^{\perp}||\,||\mathbf{s}_I||\cos(\tau_1) \\ &= g_I||\mathbf{s}_I||\sin(\tau_2) \quad (\text{since } \tau_2 = \pi/2 - \tau_1) \\ &\approx g_I||\mathbf{s}_I||\tau_2, \quad \text{if } \tau_2 \to 0, \end{aligned} \tag{41}$$

where $-\mathbf{p}_I^{\perp} = [p_{Iy}, -p_{Ix}]$ is orthogonal to $\mathbf{p}_I$, and τ_1 is the angle between $-\mathbf{p}_I^{\perp}$ and $\mathbf{s}_I$. Residual r_{aI} can be regarded as the angle τ_2 scaled by $g_I||\mathbf{s}_I||$. It gives a preliminary angle by which the traction $\mathbf{p}_I^k$ should move.

In the first iteration, the search direction is equal to r_{aI}. As a result of linearization, the auxiliary search direction $\mathbf{v}_{sI}$ is orthogonal to $\mathbf{p}_I$, as seen in Fig. 2(b). It is a linear approximation of the arc length, corresponding to the angle r_{aI}.

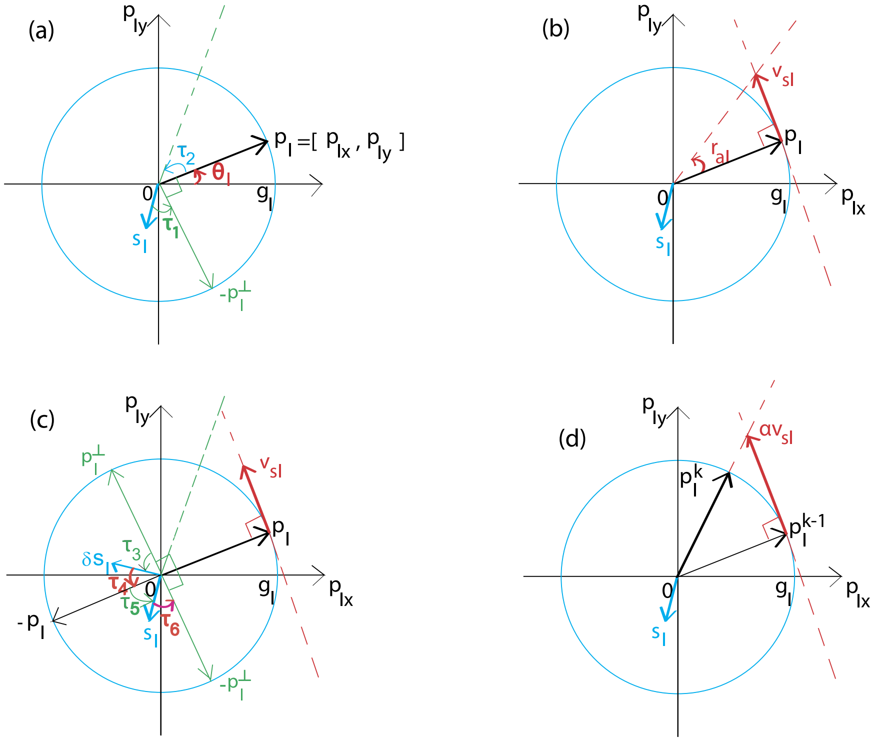

Fig. 2. First TangCG inner iteration on slip element I. (a) At the beginning, traction $\mathbf{p}_I$ lies at the circle with radius g_I. It should be opposite to slip $\mathbf{s}_I$. (b) Primary residual r_{aI} (equal to search direction v_{aI}) is computed. The auxiliary search direction $\mathbf{v}_{sI}$ is set to be orthogonal to $\mathbf{p}_I$, due to linearization. (c) Line search is performed to get a step length $\alpha = \frac{(\mathbf{v}_a, \mathbf{r}_a)}{(\mathbf{v}_a, \mathbf{q}_a)}$, where q_{aI} is obtained from the information of two angles τ_4, τ_6. (d) Update the traction by $\mathbf{p}_I^{k-1} + \alpha \mathbf{v}_{sI}^k$, which is then projected onto the circle, obtaining $\mathbf{p}_I^k$.

By the line search, the TangCG method computes the change of slip $\delta \mathbf{s}_I$, as shown in Fig. 2(c). It calculates the change of angle residual $\mathbf{q}_a$ as in (39):

$$\begin{aligned} q_{aI} &= p_{Ix} \cdot \delta s_{Iy} - p_{Iy} \cdot \delta s_{Ix} + s_{Iy} \cdot v_{Ix} - s_{Ix} \cdot v_{Iy} \\ &= ||\mathbf{p}_I^{\perp}||||\delta \mathbf{s}_I|| \cos(\tau_3) + ||\mathbf{v}_I^{\perp}||||\mathbf{s}_I|| \cos(\tau_5) \\ &= g_I ||\delta \mathbf{s}_I|| \sin(\tau_4) + ||\mathbf{v}_I^{\perp}||||\mathbf{s}_I|| \sin(\tau_6) \quad (\text{since } \tau_4 = \pi/2 - \tau_3, \tau_6 = \pi/2 - \tau_5) \\ &\approx g_I ||\delta \mathbf{s}_I|| \tau_4 + ||\mathbf{v}_I^{\perp}||||\mathbf{s}_I|| \tau_6, \quad \text{if } \tau_4, \tau_6 \to 0, \end{aligned} \tag{42}$$

where $\mathbf{v}_I^{\perp} = [-v_{Iy}, v_{Ix}]$. The two angles τ_4, τ_6 are presented in Fig. 2(c). They, together with the concrete scaling, give the angle change q_{aI}.

Then, the TangCG method computes the step length. The updating is shown in Fig. 2(d): after adding $\alpha \mathbf{v}_I$ to $\mathbf{p}_I^{k-1}$, the result is projected onto the circle again, which yields new iterate $\mathbf{p}_I^k$.

3.2.7. Preconditioning

The idea of preconditioning for Krylov subspace methods is to reduce the condition number of the coefficient matrix. In the TangCG algorithm, the preconditioner is applied to the reduced Jacobian matrix (29) in each NLCG iteration.

We use a diagonal matrix as the preconditioner, which scales the main diagonal of the Jacobian, to result in a matrix with constant diagonal entries. This constant can be chosen to be any diagonal component of Jacobian, corresponding to an adhesion element. The residuals in the adhesion and slip areas have different meanings: residuals are “slip” in the former and “angle” in the latter area. Hence, the preconditioner scales the quantities to the same measure, e.g. the measure in the adhesion area. Regarding computational time, the preconditioner should not cost much.

In the two element case, a preconditioner M for J_r in (29) is given by:

$$M = \begin{pmatrix} 1 & & \\ & A_{11}^{xx}/A_{11}^{yy} & \\ & & A_{11}^{xx}/(a_4 p_{2x}^k - a_3 p_{2y}^k) \end{pmatrix}. \tag{43}$$

With $M = S^2$, the preconditioned matrix $S J_r S$ has values of A_{11}^{xx} on its main diagonal.

3.3. The resulting algorithm

We detail the resulting TangCG algorithm with preconditioner M below. (The unpreconditioned algorithm can be obtained by setting $M = I$.)

0. Given normal pressure $\mathbf{p}_n$ and contact area C, we solve for the tangential tractions and the subdivision of the contact area. The initial adhesion and slip areas are set as: $H^0 = C$ and $S^0 = \emptyset$. We use zero initial tractions $\mathbf{p}^0 = \mathbf{0}$.
1. Start iterations $k = 1, 2, \ldots$ by computing slip $\mathbf{s}^{k-1} = \mathbf{w} + A\mathbf{p}^{k-1}$.
2. Compute corresponding primary residual $\mathbf{r}_a^{k-1}$, as Eq. (32):

$$\begin{cases} \mathbf{r}_{aI}^{k-1} = -\mathbf{s}_I^{k-1}, & I \in H^{k-1}, \\ r_{aI}^{k-1} = -(p_{Ix}^{k-1} s_{Iy}^{k-1} - p_{Iy}^{k-1} s_{Ix}^{k-1}), & I \in S^{k-1}. \end{cases} \tag{44}$$

3. Preconditioning by: $\mathbf{z}_a^{k-1} = M\mathbf{r}_a^{k-1}$.
4. Compute primary search direction $\mathbf{v}_a^k$:
   4.1 If the subdivision did not change, i.e. $H^{k-1} = H^{k-2}$, then choose $\mathbf{v}_a^k$ conjugate to $\mathbf{v}_a^{k-1}$ as:

$$\mathbf{v}_a^k = \mathbf{z}_a^{k-1} + \beta^k \cdot \mathbf{v}_a^{k-1}, \quad \text{with } \beta^k = \max(0, \beta_{PR}^k), \tag{45}$$

   where β_{PR}^k is the Polak–Ribière formula: $\beta_{PR}^k = \frac{(\mathbf{z}_a^{k-1}, \mathbf{r}_a^{k-1} - \mathbf{r}_a^{k-2})}{(\mathbf{z}_a^{k-2}, \mathbf{r}_a^{k-2})}$.

   4.2 Else, restart CG by using $\mathbf{v}_a^k = \mathbf{z}_a^{k-1}$, the steepest descent direction.
5. Compute auxiliary search direction $\mathbf{v}_s^k$ by Eq. (36):

$$\mathbf{v}_{sI}^k = \begin{cases} \mathbf{v}_{aI}^k, & I \in H^{k-1}, \\ v_{aI}^k \cdot [-p_{Iy}^{k-1}, p_{Ix}^{k-1}], & I \in S^{k-1}. \end{cases} \tag{46}$$

   Here, search direction $\mathbf{v}_{sI}^k$ can be regarded as the desired change of traction $\delta\mathbf{p}^k$. For a slip element, this corresponds to the desired change of angle v_{aI}^k.
6. Compute the change in the slip by Eq. (38), i.e.: $\delta\mathbf{s}^k = \mathbf{q}_s^k = A\mathbf{v}_s^k$.
7. Compute the change in primary residual $\mathbf{q}_a^k$ by Eq. (39):

$$\begin{cases} \mathbf{q}_{aI}^k = \mathbf{q}_{sI}^k, & I \in H^{k-1}, \\ q_{aI}^k = p_{Ix}^{k-1} \cdot \delta s_{Iy}^k - p_{Iy}^{k-1} \cdot \delta s_{Ix}^k + s_{Iy}^{k-1} \cdot v_{Ix}^k - s_{Ix}^{k-1} \cdot v_{Iy}^k, & I \in S^{k-1}. \end{cases} \tag{47}$$

8. Compute the step length by Eq. (33), i.e.: $\alpha^k = \frac{(\mathbf{r}_a^{k-1}, \mathbf{v}_a^k)}{(\mathbf{q}_a^k, \mathbf{v}_a^k)}$.
9. Update tractions: $\tilde{\mathbf{p}}^k = \mathbf{p}^{k-1} + \alpha^k \cdot \mathbf{v}_s^k$ for all elements. Then use:

$$\mathbf{p}_I^k = \begin{cases} \tilde{\mathbf{p}}_I^k, & I \in H^{k-1}, \\ g_I \cdot \tilde{\mathbf{p}}^k / ||\tilde{\mathbf{p}}^k||, & I \in S^{k-1}. \end{cases} \tag{48}$$

10. In the inner iteration, i.e. $mod(k, K) \neq 0$, the subdivision of the contact area is fixed, i.e., $H^k = H^{k-1}$, $S^k = S^{k-1}$, go to Step 1.
11. At the end of each Kth inner iteration, i.e. $mod(k, K) = 0$, do the following:
    11.1 Check elements $I \in H^{k-1}$. If $||\mathbf{p}_I^k|| > g_I$, then move I to the slip area, and set $\mathbf{p}_I^k = g_I \cdot \mathbf{p}_I^k / ||\mathbf{p}_I^k||$. This gives $\tilde{H}^k$ and $\tilde{S}^k$.
    11.2 Compute $\mathbf{s}^k = \mathbf{w} + A\mathbf{p}^k$, and check the elements $I \in \tilde{S}^k$. If $(\mathbf{p}_I^k, \mathbf{s}_I^k) > 0$, then move I to the adhesion area. This results in H^k and S^k.
    11.3 Check for convergence. If the subdivision of the contact area does not change, and the stopping criteria on the solution is satisfied, then we are done. Else go to Step 1.

Remark 1. Steps 1–10 present one inner iteration, where a nonlinear system is solved approximately by K iterations of NLCG. Line search is performed in Steps 5–8. Step 11 checks the contact conditions and convergence, according to the BCCG(K) algorithm. Moreover, Steps 9 and 11.1 insure that $||\mathbf{p}_I|| = g_I, I \in S$ is always satisfied.

Remark 2. In the above algorithm, there are K inner iterations before checking the conditions. It may happen that the tolerance is already reached within the K inner iterations. An improved version is to leave the inner iteration as soon as a tolerance is reached, and check the conditions in the outer iteration.

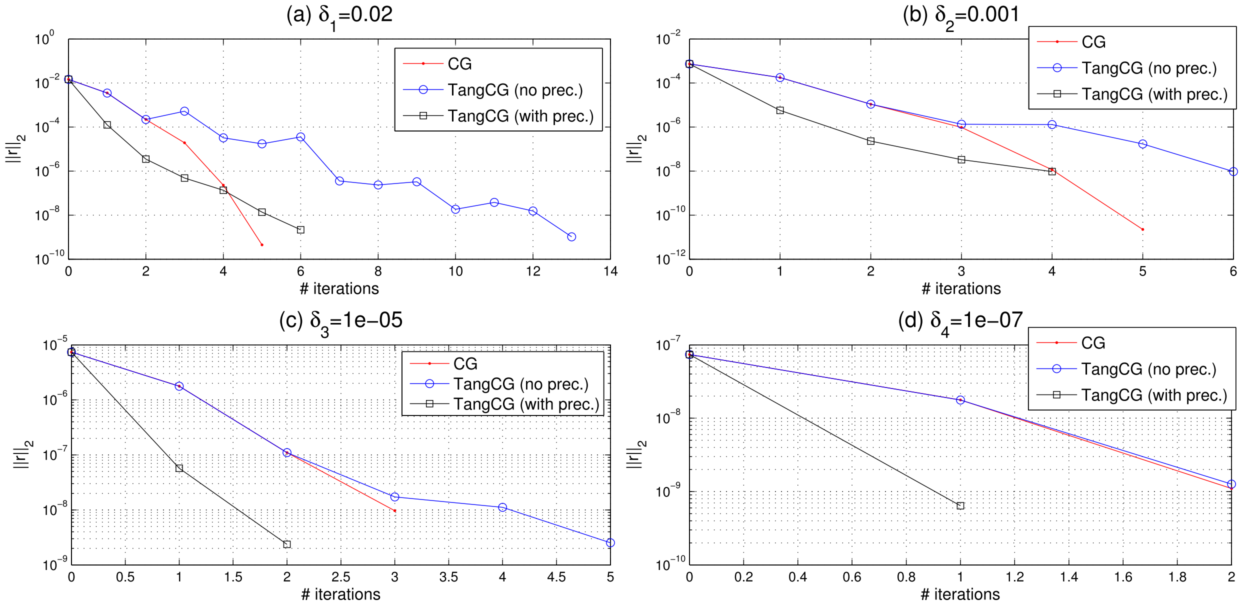

Fig. 3. The convergence behavior of TangCG with and without preconditioning, compared to CG, for four different initial points.

3.4. 2 × 2 test

To investigate the performance of NLCG within the preconditioned TangCG algorithm, we enlarge the test to a 2 × 2 grid, where Elements 1 and 2 are in adhesion and 3 and 4 are in slip. The contact area is $[-1,1] \times [-1,1]$ mm^2. The material parameters are: $G = 200$ N/mm^2, $\nu = 0.42$. The traction bound is defined as $\mathbf{g} = [0.4, 0.8, 0.282843, 0.5]$. The true solution reads: $\mathbf{p}_1^* = [0.2, 0.1]$; $\mathbf{p}_2^* = [0.1, 0.2]$; $\mathbf{p}_3^* = [0.2, 0.2]$; $\mathbf{p}_4^* = [0.3, 0.4]$. The true slip is $\mathbf{s}_1^* = [0, 0]$; $\mathbf{s}_2^* = [0, 0]$; $\mathbf{s}_3^* = [-0.1, -0.1]$; $\mathbf{s}_4^* = [-0.3, -0.4]$.

We linearize the nonlinear system at true solution $\mathbf{p}^*$. By setting $\delta p_{3x} = -\beta_3 \cdot p_{3y}^*$, $\delta_{3y} = \beta_3 \cdot p_{3x}^*$, and $\delta p_{4x} = -\beta_4 \cdot p_{4y}^*$, $\delta_{4y} = \beta_4 \cdot p_{4x}^*$, the Jacobian matrix is reduced to a 6 × 6 matrix:

$$J_r^* = \begin{pmatrix} A_{11}^{xx} & A_{11}^{xy} & A_{12}^{xx} & A_{12}^{xy} & a_{31}^{x} & a_{41}^{x} \\ A_{11}^{yx} & A_{11}^{yy} & A_{12}^{yx} & A_{12}^{yy} & a_{31}^{y} & a_{41}^{y} \\ A_{21}^{xx} & A_{21}^{xy} & A_{22}^{xx} & A_{22}^{xy} & a_{32}^{x} & a_{42}^{x} \\ A_{21}^{yx} & A_{21}^{yy} & A_{22}^{yx} & A_{22}^{yy} & a_{32}^{y} & a_{42}^{y} \\ a_{31}^{x} & a_{31}^{y} & a_{32}^{x} & a_{32}^{y} & a_{33}^{y} p_{3x}^* - a_{33}^{x} p_{3y}^* & a_{34}^{y} p_{4x}^* + a_{34}^{x} p_{4y}^* \\ a_{41}^{x} & a_{41}^{y} & a_{42}^{x} & a_{42}^{y} & a_{43}^{y} p_{3x}^* - a_{43}^{x} p_{3y}^* & a_{44}^{y} p_{4x}^* + a_{44}^{x} p_{4y}^* \end{pmatrix}. \tag{49}$$

Here, for elements $I \in S$ and the elements in contact $J \in C$:

$$\text{if } I \neq J: \quad a_{IJ}^{x} = A_{IJ}^{yx} \cdot p_{Ix}^* - A_{IJ}^{xx} \cdot p_{Iy}^*, \qquad a_{IJ}^{y} = A_{IJ}^{yy} \cdot p_{Ix}^* - A_{IJ}^{yx} \cdot p_{Iy}^*,$$

$$\text{if } I = J: \quad a_{II}^{x} = A_{II}^{yx} \cdot p_{Ix}^* - A_{II}^{xx} \cdot p_{Iy}^* + s_{Iy}^*, \qquad a_{II}^{y} = A_{II}^{yy} \cdot p_{Ix}^* - A_{II}^{yx} \cdot p_{Iy}^* - s_{Ix}^*.$$

The Jacobian matrix (49) is SPD. Its condition number is found to be 88.

The TangCG algorithm employs an NLCG process, whose convergence should be similar to the linear CG algorithm when the iterates are close to true solution $\mathbf{p}^*$. We test whether this is also true for the TangCG algorithm. We also check whether the preconditioner improves the convergence near $\mathbf{p}^*$.

As a test, four different initial points are implemented, respectively, with their absolute distances to $\mathbf{p}^*$ being $\delta_i = 0.02, 0.001, 1 \times 10^{-5}, 1 \times 10^{-7}$ $(i = 1, 2, 3, 4)$. CG is applied to the linearized system $J_r^* \delta \mathbf{p}_r^* = \mathbf{0}$, with initial $\delta \mathbf{p}_r^0 = T^T(\mathbf{p}^* - \mathbf{p}^0)$. The iterate is converted to traction by (30) and further the nonlinear residual is compared with those by the unpreconditioned and preconditioned TangCG algorithms.

These results are shown in Fig. 3. It can be seen that the closer the initial point is at the true solution, the more similar the convergence by TangCG is to the CG convergence. As the iterate approaches the true solution, the Jacobian matrix obtained by linearization in each TangCG iteration is close to the exact Jacobian J_r^*. In this case, the NLCG method within TangCG is the same as CG, as seen in Fig. 3(d). Moreover, we find that the preconditioner indeed improves the convergence of TangCG.

Table 1
Test setting: seven cases with different percentages of slip. Longitudinal shift ξ, lateral shift η and rotation shift ϕ are prescribed. They are used to compute rigid shift $\mathbf{w}$ (see Eq. (11)).

	Subdivision of contact area	ξ	η	ϕ
Case 1	full adhesion	0	0	5×10^{-7}
Case 2	0.6% slip	1.5×10^{-5}	0	1×10^{-5}
Case 3	20% slip	0	0.0015	0.0012
Case 4	40% slip	0.0021	0.0010	0.003
Case 5	60% slip	0.0037	0.0048	0.004
Case 6	80% slip	0	0.005	0.009
Case 7	full slip	0.0044	0	0.08

Table 2
Unpreconditioned TangCG(K): the total number of inner iterations with different values of K for the seven slip cases, using a 120×100 grid (with $5732 \times 2 = 11464$ unknowns).

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7
$K = 1$	30	46	135	100	148	85	48
$K = 2$	30	52	66	94	119	81	48
$K = 3$	30	54	86	83	105	77	65
$K = 4$	30	50	94	104	117	95	57
$K = 5$	30	53	78	104	165	87	37

4. Numerical results

Various Cattaneo shift problems are tested here with different patterns of slip. A sphere is pressed onto a plane, and is then shifted tangentially. The sphere and plane are of the same material, with shear modulus $G = 200$ N/mm^2, and Poisson's ratio $\nu = 0.42$. The potential contact area is $[-a, a] \times [-b, b]$ where $a = b = 1.2857$ mm. The total normal force is $F_n = 9.1954$ N. The radii of the sphere are $R_x = R_y = 50$ mm, and the profile of the sphere is $h = \frac{1}{2R_x}x^2 + \frac{1}{2R_y}y^2$. Coulomb's law is applied with friction coefficient $\mu = 0.4$.

Since the two contacting bodies are quasi-identical, the solution process is decoupled to first solve the normal problem, followed by the tangential problem. The former is solved by Hertz theory [14] or a numerical method [35], which results in a contact area C and normal pressure $\mathbf{p}_n$. Applying Coulomb's law, we thus obtain the traction bound $\mathbf{g} = \mu \mathbf{p}_n$. Below we focus on solving the tangential problems, with seven different slip cases, as specified in Table 1. The stopping criterion on the accuracy of the tractions is given by

$$\frac{||\mathbf{p}^k - \mathbf{p}^{k-1}||}{||\mathbf{p}^k||} < \varepsilon, \tag{50}$$

where, if not specified differently, we use $\varepsilon = 10^{-5}$. The norm used in this section is the "root-mean-square" norm, defined as $\| \mathbf{x} \|_{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$. This stopping criterion is chosen since engineers pay most attention to tractions, rather than to the slip, which is the residual of nonlinear system (27). Moreover, when (50) is satisfied, the residual of this nonlinear system is smaller than 10^{-8} in our numerical tests.

The tangential solvers are TangCG(K) algorithms with and without preconditioning, using the improved version mentioned in Remark 2. Remember that K is the number of NLCG iterations for one nonlinear system before checking the contact conditions. We implemented these two algorithms for the seven problem cases, in order to find an optimal value for K. They are then compared with each other, yielding the most efficient algorithm. It is also compared with the ConvexGS method [32], from the perspective of iteration numbers and CPU time.

4.1. Optimal K

First we use a 120×100 grid, which involves 5732 contacting elements with $5732 \times 2 = 11\,464$ traction unknowns. We apply $K \in \{1, 2, 3, 4, 5\}$ for the seven cases of Table 1.

Table 2 gives the total number of inner NLCG iterations by TangCG(K) iterations. For Case 1, the full adhesion problem, the results are independent of the values of K. Remember that the TangCG(K) method starts with the subdivision $H^0 = C$, and $S^0 = \emptyset$. This is already the correct subdivision in the full adhesion case. The governing system is linear, and TangCG is actually equivalent to the CG method.

When slip occurs, different K values show different results. $K = 1$ is the fastest in the slight slip Case 2. It gets worse when more slip is included, as seen in Cases 3–6, where $K = 3$ seems a proper choice. The reason is that the slip area has to be found from initial area $S^0 = \emptyset$. If $K = 1$, then only one inner iteration is performed and the traction iterates may not be accurate enough to perform a useful adjustment of the subdivision of the contact area: in Step 11 of the TangCG algorithm (see Section 3.3), many elements that are first moved from the adhesion to slip area, may be immediately moved

Table 3
Preconditioned TangCG(K): the total number of inner iterations with different values of K for the seven slip cases, using a 120×100 grid (with $5732 \times 2 = 11464$ unknowns).

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7
$K = 1$	30	43	52	59	66	52	27
$K = 2$	30	50	47	52	77	60	32
$K = 3$	30	48	49	55	49	46	37
$K = 4$	30	45	56	52	56	50	34
$K = 5$	30	54	52	57	56	47	22

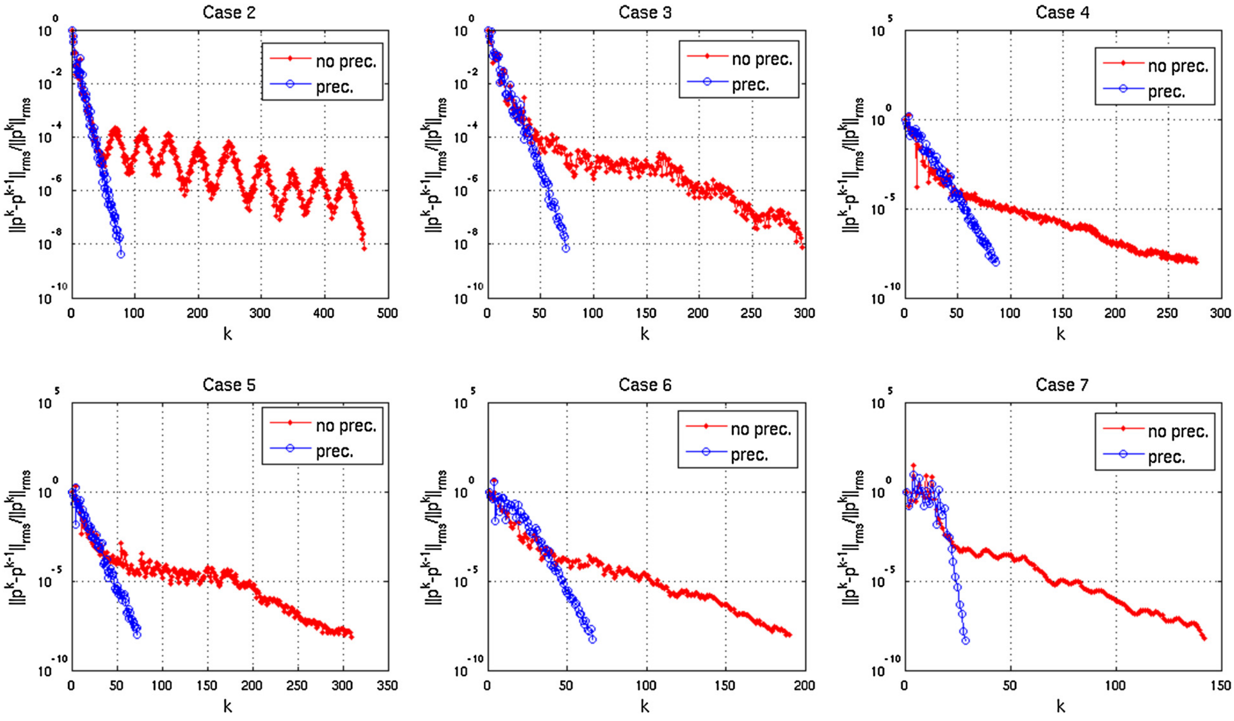

Fig. 4. Convergence of TangCG(3) with and without preconditioning for Cases 2–7, with $\varepsilon = 10^{-8}$ in the stopping criterion (50). The horizontal axis is number of inner iterations, and the vertical axis is the relative difference between two iterates.

back. Using a larger value for K can improve this situation. However, if K is too large, there would be unnecessarily many inner iterations.

In the full slip Case 7, $K = 5$ results in the fastest convergence. The initial slip area S^0 tends to be $S = C$. More precise solutions can move more elements into the slip area at once, which accelerates the convergence. Moreover, the results of other values for K are also acceptable.

From the above discussion, we find $K = 3$ is an optimal choice for the unpreconditioned TangCG(K) method in different cases. The same conclusion can be found for TangCG(K) with preconditioning, from the results in Table 3. Therefore, we will use $K = 3$ in the following tests.

4.2. Comparison between TangCG(3) with and without preconditioning

Comparing these two tables, we find that the preconditioner helps to accelerate the convergence, depending on the different slip situations. One exception is the full adhesion Case 1, where the main diagonal entries of the Jacobian are almost the same. Hence, the condition number after preconditioning hardly reduces.

Fig. 4 displays the convergence behavior of the two methods for Cases 2–7, that include slip. In this figure, we use $\varepsilon = 10^{-8}$ in the stopping criterion (50), to examine the behavior of these two methods near the true solution. We find that TangCG(3) with preconditioning greatly speeds up the convergence and makes the method much more robust. It gives an almost linear reduction of the difference between two iterates, when approaching the true solution. TangCG(3) without preconditioning, on the other hand, displays a slower convergence speed, and even an oscillate reduction in Case 2.

Table 4
Numbers of inner iterations by preconditioned TangCG(3), and numbers of iterations by ConvexGS (shown in brackets).

Discretization	30 × 25	60 × 50	120 × 100	240 × 200
# traction unknowns	728	2880	11 464	45 736
Case 1	14 (23)	23 (32)	30 (50)	49 (77)
Case 2	20 (26)	33 (32)	48 (48)	62 (72)
Case 3	33 (31)	39 (37)	49 (56)	65 (88)
Case 4	33 (32)	41 (33)	55 (48)	76 (73)
Case 5	33 (31)	42 (32)	49 (45)	70 (69)
Case 6	34 (29)	34 (30)	46 (35)	73 (53)
Case 7	14 (18)	12 (18)	37 (18)	25 (18)

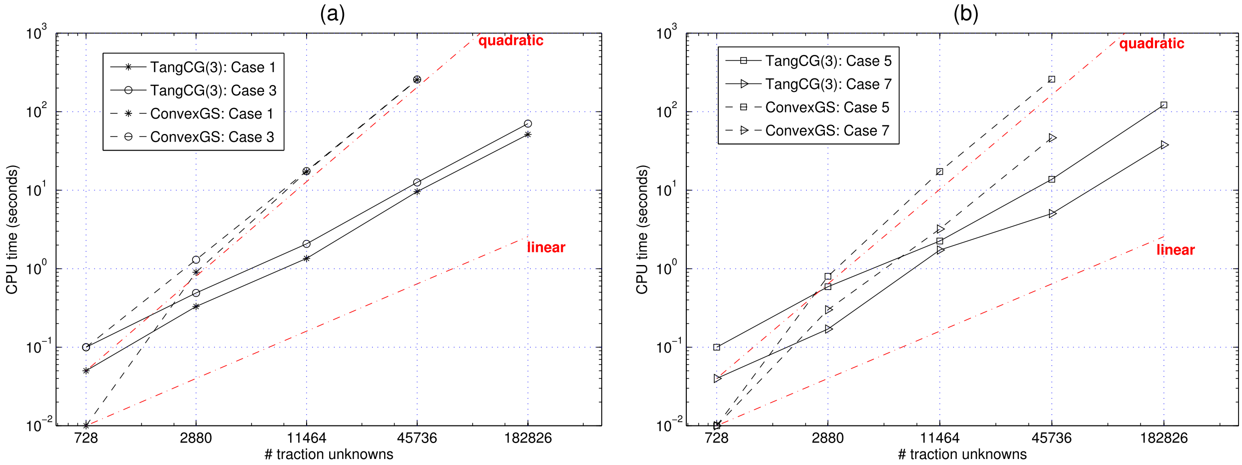

Fig. 5. CPU time in seconds by TangCG(3) with preconditioning and ConvexGS for (a) Cases 1 and 3, (b) Cases 5 and 7.

4.3. Comparison with ConvexGS method

We compare TangCG(3) with preconditioning to the performance of the ConvexGS method in the software CONTACT. The codes of the former were written in Matlab 7.13 (R2011b). The latter was programmed in Fortran. Both methods are implemented on a Linux desktop PC (Intel Core I5-2400 processor, quad-core, 3.1 GHz). First of all, iteration numbers by these two methods are compared in Table 4, where the iteration numbers by ConvexGS are given in brackets. As can be seen, iteration numbers of TangCG(3) are of $\mathcal{O}(n^{0.4})$, with n the number of unknowns.[3] Moreover, TangCG(3) requires fewer iterations than ConvexGS for the test cases with a small percentage of slip. With more slip, the iteration numbers are comparable.

Fig. 5 shows the CPU time (in seconds) for some of the test cases, for TangCG(3) with preconditioning (denoted by solid lines), and ConvexGS (denoted by dashed lines). For the smallest problem, both methods require approximately the same computing time. The superiority of TangCG(3) shows when more unknowns are involved. The *speedup factor* is defined as the ratio of time by ConvexGS and by TangCG(3). For around 1×10^4 unknowns, the speedup factors for Cases 1, 3, 5, 7 are about 13, 8, 8, 2, respectively. They grow to 27, 20, 19, 9 when the number of unknowns is multiplied by a factor of 4. Fig. 5 also displays curves of quadratic complexity $\mathcal{O}(n^2)$ and linear complexity $\mathcal{O}(n)$ (denoted by the dash–dot lines). The CPU time curves for TangCG(3) remain between these two complexity lines. We find that the complexity of TangCG(3) is $\mathcal{O}(n^{1.7})$, which is faster than ConvexGS with $\mathcal{O}(n^{2.3})$. Note that ConvexGS cannot benefit from the favorable Toeplitz matrix structure.

5. Conclusion

In this paper we propose the TangCG algorithm to solve a nonlinear constrained optimization formulation, arising from the frictional contact problem. The corresponding KKT conditions provide the governing equations and contact conditions. The TangCG algorithm applies an active set strategy. The subdivision of the contact area is fixed and the resulting governing system is solved approximately. The resulting solution is used to modify the subdivision based on these contact conditions.

Because the magnitude of the tractions on a slip element equals the traction bound, we can change the conventional traction variables to azimuth angles, when placing the traction vector of each slip element in a polar coordinate system.

[3] There is no good estimate for Case 7, since the corresponding iteration numbers do not gradually increase but oscillate as unknown n grows.

This changes the constraints to simple bounds such that the BCCG strategy can be used. The resulting nonlinear equations are solved by several NLCG iterations, which is based on linearization and applies a CG method in each iteration. A diagonal scaling preconditioner is necessary to bring the residuals of adhesion and slip areas to the same scale, and all matrix–vector products in this algorithm are speeded up by FFTs.

The TangCG algorithm is tested for Cattaneo shift problems, with different amounts of slip. The preconditioner is found to improve robustness and accelerate convergence when slip occurs. The computational time is reduced dramatically compared to ConvexGS, a state-of-the-art method for concentrated frictional contact problems. The corresponding speedup factor grows as problem size increases. This confirms the efficiency of our new method. Moreover, since rolling contact problems can be transformed into a sequence of shift problems, our TangCG method can also be applied in this case.

Acknowledgements

Jing Zhao is grateful for the scholarship by China Scholarship Council (CSC).

References

[1] T. Andersson, The boundary element method applied to two-dimensional contact problems with friction, in: Proceedings of the 3rd International Seminar on Recent Advances in Boundary Element Methods, Springer, Berlin, 1981.
[2] M. Barboteu, P. Alart, M. Vidrascu, A domain decomposition strategy for nonclassical frictional multi-contact problems, Comput. Methods Appl. Mech. Eng. 190 (2001) 4785–4803.
[3] S.P. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, UK, 2004.
[4] C. Cattaneo, Sul contatto di due corpi elastici: distribuzione locale degli sforzi, Rend. Accad. Naz. Lincei 27 (1938) 434–436.
[5] W.W. Chen, Q.J. Wang, A numerical model for the point contact of dissimilar materials considering tangential tractions, Mech. Mater. 40 (2008) 936–948.
[6] Z. Dostál, T. Kozubek, A. Markopoulos, T. Brzobohatỳ, V. Vondrák, P. Horyl, A theoretically supported scalable TFETI algorithm for the solution of multibody 3D contact problems with friction, Comput. Methods Appl. Mech. Eng. 205 (2012) 110–120.
[7] R. Enblom, Deterioration mechanisms in the wheel–rail interface with focus on wear prediction: a literature review, Veh. Syst. Dyn. 47 (2009) 661–700.
[8] R. Fletcher, C.M. Reeves, Function minimization by conjugate gradients, Comput. J. 7 (1964) 149–154.
[9] J.Á. González, R. Abascal, Solving 2D rolling problems using the NORM–TANG iteration and mathematical programming, Comput. Struct. 78 (2000) 149–160.
[10] S. Grassie, Rolling contact fatigue on the British railway system: treatment, Wear 258 (2005) 1310–1318.
[11] H. Hertz, Über die Berührung fester elastischer Körper, J. Reine Angew. Math. 92 (1882).
[12] S. Hüeber, G. Stadler, B.I. Wohlmuth, A primal–dual active set algorithm for three-dimensional contact problems with Coulomb friction, SIAM J. Sci. Comput. 30 (2008) 572–596.
[13] M. Ignesti, A. Innocenti, L. Marini, E. Meli, A. Rindi, A numerical procedure for the wheel profile optimisation on railway vehicles, Proc. Inst. Mech. Eng., Part J J. Eng. Tribol. 228 (2014) 206–222.
[14] K.L. Johnson, Contact Mechanics, Cambridge University Press, UK, 1987.
[15] F. Jourdan, P. Alart, M. Jean, A Gauss–Seidel like algorithm to solve frictional contact problems, Comput. Methods Appl. Mech. Eng. 155 (1998) 31–47.
[16] J.J. Kalker, Contact mechanical algorithms, Commun. Appl. Numer. Methods 4 (1988) 25–32.
[17] J.J. Kalker, Three-Dimensional Elastic Bodies in Rolling Contact, Kluwer Academic Publisher, The Netherlands, 1990.
[18] N. Kikuchi, J.T. Oden, Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods, SIAM, Philadelphia, USA, 1987.
[19] F. Kosior, N. Guyot, G. Maurice, Analysis of frictional contact problem using boundary element method and domain decomposition method, Int. J. Numer. Methods Eng. 46 (1999) 65–82.
[20] R. Krause, A nonsmooth multiscale method for solving frictional two-body contact problems in 2D and 3D with multigrid efficiency, SIAM J. Sci. Comput. 31 (2009) 1399–1423.
[21] H.W. Kuhn, Nonlinear programming: a historical view, in: Traces and Emergence of Nonlinear Programming, 2014, pp. 393–414.
[22] K.W. Man, Contact Mechanics Using Boundary Elements, Computational Mechanics Publisher, Boston, MA, 1994.
[23] R.D. Mindlin, Compliance of elastic bodies in contact, J. Appl. Mech. 16 (2013).
[24] K.G. Murty, Linear Complementarity, Linear and Nonlinear Programming, Heldermann, Berlin, 1988.
[25] I.P. Panagiotopoulos, A nonlinear programming approach to the unilateral contact, and friction-boundary value problem in the theory of elasticity, Ing.-Arch. 44 (1975) 421–432.
[26] E. Polak, G. Ribière, Note sur la convergence de méthodes de directions conjuguées, ESAIM: Math. Model. Numer. Anal. (Modél. Math. Anal. Numér.) 3 (1969) 35–43.
[27] M.J.D. Powell, Restart procedures for the conjugate gradient method, Math. Program. 12 (1977) 241–254.
[28] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in Fortran 77: The Art of Scientific Computing, 1992.
[29] E. Rabinowicz, Friction and Wear of Materials, vol. 2, Wiley, 1995.
[30] H.M. Stanley, T. Kato, An FFT-based method for rough surface contact, J. Tribol. 119 (1997) 481–485.
[31] A. Van Beek, Advanced Engineering Design: Lifetime Performance and Reliability, Delft University of Technology, Mechanical Engineering, Delft, The Netherlands, 2006.
[32] E.A.H. Vollebregt, A Gauss–Seidel type solver for special convex programs, with application to frictional contact mechanics, J. Optim. Theory Appl. 87 (1995) 47–67.
[33] E.A.H. Vollebregt, Refinement of Kalker's rolling contact model, in: Proceedings of the 8th International Conference on Contact Mechanics and Wear of Rail/Wheel Systems, Firenze, 2009, pp. 149–156.
[34] E.A.H. Vollebregt, The bound-constrained conjugate gradient method for non-negative matrices, J. Optim. Theory Appl. (2013) 1–23.
[35] E.A.H. Vollebregt, User's guide for CONTACT, Vollebregt & Kalker's rolling and sliding contact model, Technical Report TR09-03, version 13.1, VORtech, 2013.
[36] E.A.H. Vollebregt, A new solver for the elastic normal contact problem using conjugate gradients, deflation, and an FFT-based preconditioner, J. Comput. Phys. 257 (2014) 333–351.
[37] K. Willner, Fully coupled frictional contact using elastic halfspace theory, J. Tribol. 130 (2008) 031405.
[38] P. Wriggers, Computational Contact Mechanics, Springer-Verlag, Berlin, Heidelberg, 2006.