Original Article

Institution of MECHANICAL ENGINEERS
Journal of MECHANICAL ENGINEERING SCIENCE

Proc IMechE Part C:
J Mechanical Engineering Science
0(0) 1–14
© IMechE 2013
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0954406213509086
pic.sagepub.com
SAGE

The use of static reduction in the finite element solution of two-dimensional frictional contact problems

A Thaitirarot[1], RC Flicek[1], DA Hills[1] and JR Barber[2]

Abstract

In this paper, detailed instructions are given for performing static reduction on a finite element description of an elastic contact problem, thus reducing the dimensionality of the problem to the set of contact nodes alone. This significantly reduces the computational time for the solution to evolutionary contact problems and also gives the user greater control over the detailed implementation of the contact and friction laws. The reduced stiffness matrix is also an essential ingredient in the determination of the critical coefficient of friction for the problem to be well posed, and it facilitates the determination of the conditions under which a frictional system may shake down under periodic loading.

Keywords

Contact problems, static reduction, shakedown, Coulomb friction, substructuring, finite element method

Date received: 18 April 2013; accepted: 17 September 2013

Introduction

Finite element solutions to quasi-static frictional elastic contact problems are generally constructed by linking finite element models of the contacting bodies through contact elements and utilizing either penalty methods or the augmented Lagrangian method to handle the discontinuous nature of the contact boundary conditions. The resulting models are often rather large and this can be a barrier to achieving good accuracy, particularly in cyclic loading problems where we would like to run a sufficient number of cycles to reach a steady state, or when multiple load cases are to be studied. Also, it can be difficult to decide whether unexpected features of the solution (e.g. contact traction distributions) reflect real features of the underlying continuum problem, or are mere artefacts of the regularization inherent in the contact elements.

An alternative approach that is particularly attractive in the case of two-dimensional problems is to use the equilibrium equations for the degrees of freedom at the non-contact nodes to reduce the problem to a contact problem defined on the N contact nodes alone. This procedure is known as *static reduction* or *substructuring*. In effect, the problem then becomes formally equivalent to a system of N massless rigid blocks connected by a general $N \times N$ stiffness matrix, with each block potentially in frictional contact with a rigid plane obstacle.

In this formulation, the evolutionary contact problem can be defined and solved without the necessity of contact elements, using iterative applications of the contact inequalities[1] or linear complementarity (LCP) methods.[2] Also, the reduced stiffness matrix provides a convenient vehicle for exploring other general features of the contact problem, such as the maximum amplitude of cyclic external loads below which the system is capable of shaking down.

The general idea of static reduction is, of course, well known. The purpose of the present paper is to define precisely the steps that need to be taken to convert a conventional finite element model to a reduced model, and to give access to a suite of files that can be downloaded for this purpose.

Single elastic body pressed against a rigid body

In explaining the procedure for extracting the reduced stiffness matrix from the full finite element stiffness matrix $\boldsymbol{K}$, it is convenient to start with the simplest

[1]Department of Engineering Science, University of Oxford, UK
[2]Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI USA

Corresponding author:
RC Flicek, Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, UK.
Email: robert.flicek@eng.ox.ac.uk

case where there is only a single elastic body that makes frictional contact with a rigid obstacle at a set of contact nodes. We shall find that the more general problem where two elastic bodies make contact with each other can be solved using a series of steps for the two bodies separately.

We discretize the body using the finite element method and characterize the resulting nodes as belonging to one of three complementary sets: the contact nodes $\mathcal{S}_C$, the externally loaded nodes $\mathcal{S}_E$ and the unloaded nodes $\mathcal{S}_I$. For the most general case, the set $\mathcal{S}_E$ should be further partitioned into $\mathcal{S}_U$, where non-zero nodal displacements are prescribed, and $\mathcal{S}_T$, where non-zero nodal forces are prescribed, but in most practical problems one of these sets will be null. In other words, we usually define the problem either under force control or under displacement control. In all cases we shall denote the nodal displacements by $\boldsymbol{u}$ and the nodal forces by $\boldsymbol{f}$ with appropriate subscripts or superscripts.

Sign convention

Since the problem is two-dimensional, the contact nodal force $\boldsymbol{f}_j^{\mathrm{C}}$ will have two components, which we denote by

$$\boldsymbol{f}_j^{\mathrm{C}} = \begin{Bmatrix} q_j \\ p_j \end{Bmatrix}, \tag{1}$$

where p_j is the force normal to the contact surface and is defined as positive when compressive and q_j is the component tangential to the surface with the sign convention of Figure 1(a).

The corresponding nodal displacements shown in Figure 1(b) are denoted by

$$\boldsymbol{u}_j^{\mathrm{C}} = \begin{Bmatrix} v_j \\ w_j \end{Bmatrix}, \tag{2}$$

where w_j is normal to the contact surface and hence contributes to a positive gap, whilst v_j is a tangential (slip) displacement.

We then construct contact nodal force and nodal displacement vectors

$$\boldsymbol{f}^{\mathrm{C}} = \left\{\boldsymbol{f}_1^{\mathrm{C}}, \boldsymbol{f}_2^{\mathrm{C}}, \boldsymbol{f}_3^{\mathrm{C}}, \ldots, \boldsymbol{f}_N^{\mathrm{C}}\right\}^T \tag{3a}$$

$$\boldsymbol{u}^{\mathrm{C}} = \left\{\boldsymbol{u}_1^{\mathrm{C}}, \boldsymbol{u}_2^{\mathrm{C}}, \boldsymbol{u}_3^{\mathrm{C}}, \ldots, \boldsymbol{u}_N^{\mathrm{C}}\right\}^T, \tag{3b}$$

where N is the number of contact nodes, and since the system is linear elastic, we can write

$$\boldsymbol{f}^{\mathrm{C}} = \boldsymbol{f}^{w} + \boldsymbol{K}^{\mathrm{C}}\boldsymbol{u}^{\mathrm{C}}, \tag{4}$$

where $\boldsymbol{K}^{\mathrm{C}}$ is a positive definite and symmetric *contact stiffness matrix* and $\boldsymbol{f}^{w}$ comprises the contact nodal forces that would be developed by the given external loading if the contact nodal displacements were all constrained to be zero (i.e. if $\boldsymbol{u}^{\mathrm{C}} = 0$).

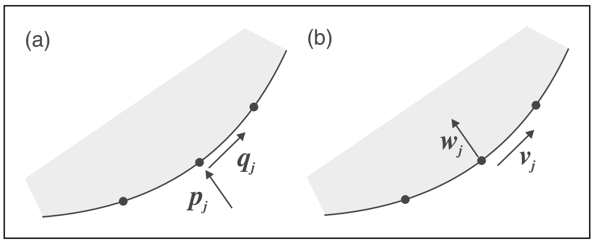

Figure 1. Sign convention for (a) nodal forces and (b) nodal displacements.

Eliminating the displacements at the unloaded nodes

Since $\mathcal{S}_C$ and $\mathcal{S}_E$ are sets of points on the boundary, whereas $\mathcal{S}_I$ includes all the nodes in the interior of the body (an area), most of the nodes will be in $\mathcal{S}_I$ and the dimensionality of the resulting matrices can be significantly reduced by using the condition $\boldsymbol{f}^{\mathrm{I}} = 0$ to eliminate these degrees of freedom. We first partition the full stiffness matrix such that

$$\boldsymbol{f} \equiv \begin{Bmatrix} \boldsymbol{f}^{\mathrm{I}} \\ \boldsymbol{f}^{\mathrm{E}} \\ \boldsymbol{f}^{\mathrm{C}} \end{Bmatrix} = \boldsymbol{K}\boldsymbol{u} \equiv \begin{bmatrix} \boldsymbol{K}^{\mathrm{II}} & \boldsymbol{K}^{\mathrm{IE}} & \boldsymbol{K}^{\mathrm{IC}} \\ \boldsymbol{K}^{\mathrm{EI}} & \boldsymbol{K}^{\mathrm{EE}} & \boldsymbol{K}^{\mathrm{EC}} \\ \boldsymbol{K}^{\mathrm{CI}} & \boldsymbol{K}^{\mathrm{CE}} & \boldsymbol{K}^{\mathrm{CC}} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}^{\mathrm{I}} \\ \boldsymbol{u}^{\mathrm{E}} \\ \boldsymbol{u}^{\mathrm{C}} \end{Bmatrix}, \tag{5}$$

from which we conclude that

$$\boldsymbol{f}^{\mathrm{I}} = \boldsymbol{K}^{\mathrm{II}}\boldsymbol{u}^{\mathrm{I}} + \boldsymbol{K}^{\mathrm{IE}}\boldsymbol{u}^{\mathrm{E}} + \boldsymbol{K}^{\mathrm{IC}}\boldsymbol{u}^{\mathrm{C}}, \tag{6}$$

and

$$\begin{Bmatrix} \boldsymbol{f}^{\mathrm{E}} \\ \boldsymbol{f}^{\mathrm{C}} \end{Bmatrix} = \begin{bmatrix} \boldsymbol{K}^{\mathrm{EI}} & \boldsymbol{K}^{\mathrm{EE}} & \boldsymbol{K}^{\mathrm{EC}} \\ \boldsymbol{K}^{\mathrm{CI}} & \boldsymbol{K}^{\mathrm{CE}} & \boldsymbol{K}^{\mathrm{CC}} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}^{\mathrm{I}} \\ \boldsymbol{u}^{\mathrm{E}} \\ \boldsymbol{u}^{\mathrm{C}} \end{Bmatrix}. \tag{7}$$

Substituting $\boldsymbol{f}^{\mathrm{I}} = 0$ into equation (6), we have

$$\boldsymbol{K}^{\mathrm{II}}\boldsymbol{u}^{\mathrm{I}} = -\boldsymbol{K}^{\mathrm{IE}}\boldsymbol{u}^{\mathrm{E}} - \boldsymbol{K}^{\mathrm{IC}}\boldsymbol{u}^{\mathrm{C}},$$

and hence, solving for $\boldsymbol{u}^{\mathrm{I}}$

$$\boldsymbol{u}^{\mathrm{I}} = -\left[\boldsymbol{K}^{\mathrm{II}}\right]^{-1}\left[\boldsymbol{K}^{\mathrm{IE}} \quad \boldsymbol{K}^{\mathrm{IC}}\right]\begin{Bmatrix} \boldsymbol{u}^{\mathrm{E}} \\ \boldsymbol{u}^{\mathrm{C}} \end{Bmatrix}. \tag{8}$$

Using this result to eliminate $\boldsymbol{u}^{\mathrm{I}}$ in equation (7), we then obtain

$$\begin{Bmatrix} \boldsymbol{f}^{\mathrm{E}} \\ \boldsymbol{f}^{\mathrm{C}} \end{Bmatrix} = \left[-\begin{bmatrix} \boldsymbol{K}^{\mathrm{EI}} \\ \boldsymbol{K}^{\mathrm{CI}} \end{bmatrix}\left[\boldsymbol{K}^{\mathrm{II}}\right]^{-1}\left[\boldsymbol{K}^{\mathrm{IE}} \quad \boldsymbol{K}^{\mathrm{IC}}\right] + \begin{bmatrix} \boldsymbol{K}^{\mathrm{EE}} & \boldsymbol{K}^{\mathrm{EC}} \\ \boldsymbol{K}^{\mathrm{CE}} & \boldsymbol{K}^{\mathrm{CC}} \end{bmatrix}\right]\begin{Bmatrix} \boldsymbol{u}^{\mathrm{E}} \\ \boldsymbol{u}^{\mathrm{C}} \end{Bmatrix}. \tag{9}$$

At this point, it is convenient to use these relations to define a reduced matrix

$$\boldsymbol{L} = \left[-\begin{bmatrix} \boldsymbol{K}^{\mathrm{EI}} \\ \boldsymbol{K}^{\mathrm{CI}} \end{bmatrix} \left[\boldsymbol{K}^{\mathrm{II}}\right]^{-1} \left[\boldsymbol{K}^{\mathrm{IE}} \quad \boldsymbol{K}^{\mathrm{IC}}\right] + \begin{bmatrix} \boldsymbol{K}^{\mathrm{EE}} & \boldsymbol{K}^{\mathrm{EC}} \\ \boldsymbol{K}^{\mathrm{CE}} & \boldsymbol{K}^{\mathrm{CC}} \end{bmatrix} \right],$$

and to partition it such that

$$\begin{Bmatrix} \boldsymbol{f}^{\mathrm{E}} \\ \boldsymbol{f}^{\mathrm{C}} \end{Bmatrix} = \begin{bmatrix} \boldsymbol{L}^{\mathrm{EE}} & \boldsymbol{L}^{\mathrm{EC}} \\ \boldsymbol{L}^{\mathrm{CE}} & \boldsymbol{L}^{\mathrm{CC}} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}^{\mathrm{E}} \\ \boldsymbol{u}^{\mathrm{C}} \end{Bmatrix}. \tag{10}$$

The force-control problem. In this case, $\mathcal{S}_U$ is null and the nodal forces at the externally loaded nodes $\boldsymbol{f}^E$ in $\mathcal{S}_E$ are known functions of time t. Equation (10) can be written

$$\boldsymbol{f}^{\mathrm{E}} = \boldsymbol{L}^{\mathrm{EE}}\boldsymbol{u}^{\mathrm{E}} + \boldsymbol{L}^{\mathrm{EC}}\boldsymbol{u}^{\mathrm{C}} \tag{11a}$$

$$\boldsymbol{f}^{\mathrm{C}} = \boldsymbol{L}^{\mathrm{CE}}\boldsymbol{u}^{\mathrm{E}} + \boldsymbol{L}^{\mathrm{CC}}\boldsymbol{u}^{\mathrm{C}}, \tag{11b}$$

and since the external nodal force vector $\boldsymbol{f}^{\mathrm{E}}$ is known, we can solve (11a) for the vector of nodal displacements at the externally loaded nodes $\boldsymbol{u}^{\mathrm{E}}$ obtaining

$$\boldsymbol{u}^{\mathrm{E}} = \left[\boldsymbol{L}^{\mathrm{EE}}\right]^{-1}\left[\boldsymbol{f}^{\mathrm{E}} - \boldsymbol{L}^{\mathrm{EC}}\boldsymbol{u}^{\mathrm{C}}\right]. \tag{12}$$

Substituting this result in (11b), we obtain

$$\begin{aligned} \boldsymbol{f}^{\mathrm{C}} &= \boldsymbol{L}^{\mathrm{CE}}\left[\left[\boldsymbol{L}^{\mathrm{EE}}\right]^{-1}\left[\boldsymbol{f}^{\mathrm{E}} - \boldsymbol{L}^{\mathrm{EC}}\boldsymbol{u}^{\mathrm{C}}\right]\right] + \boldsymbol{L}^{\mathrm{CC}}\boldsymbol{u}^{\mathrm{C}} \\ &= \boldsymbol{K}^{\mathrm{E}}\boldsymbol{f}^{\mathrm{E}} + \boldsymbol{K}^{\mathrm{C}}\boldsymbol{u}^{\mathrm{C}}, \end{aligned} \tag{13}$$

where

$$\boldsymbol{K}^{\mathrm{E}} = \boldsymbol{L}^{\mathrm{CE}}\left[\boldsymbol{L}^{\mathrm{EE}}\right]^{-1}, \tag{14}$$

and

$$\boldsymbol{K}^{\mathrm{C}} = \left[\boldsymbol{L}^{\mathrm{CC}} - \boldsymbol{L}^{\mathrm{CE}}\left[\boldsymbol{L}^{\mathrm{EE}}\right]^{-1}\boldsymbol{L}^{\mathrm{EC}}\right] \tag{15}$$

is the contact stiffness matrix. Notice that equation (13) is of the same form as equation (4), with

$$\boldsymbol{f}^{w} = \boldsymbol{K}^{\mathrm{E}}\boldsymbol{f}^{\mathrm{E}}. \tag{16}$$

In other words, this procedure determines the reduced nodal forces $\boldsymbol{f}^w$ due to prescribed external forces $\boldsymbol{f}^{\mathrm{E}}$, as well as the contact stiffness matrix $\boldsymbol{K}^{\mathrm{C}}$, which is given by equation (15).

The displacement-control problem. If the nodal displacements at the externally loaded nodes $\boldsymbol{u}^E$ are prescribed functions of time, equation (11b) still applies, and hence

$$\boldsymbol{f}^{\mathrm{C}} = \boldsymbol{L}^{\mathrm{CE}}\boldsymbol{u}^{\mathrm{E}} + \boldsymbol{L}^{\mathrm{CC}}\boldsymbol{u}^{\mathrm{C}}.$$

Since in this case the nodal displacements $\boldsymbol{u}^E$ are known, we can immediately write the equations in the form of equation (4), with

$$\boldsymbol{f}^{w} = \boldsymbol{L}^{\mathrm{CE}}\boldsymbol{u}^{\mathrm{E}} \tag{17}$$

$$\boldsymbol{K}^{\mathrm{C}} = \boldsymbol{L}^{\mathrm{CC}}. \tag{18}$$

Notice that the contact stiffness matrix $\boldsymbol{K}^{\mathrm{C}}$ is affected by whether the externally loaded nodes are force or displacement controlled. This is because the contact stiffness matrix is the solution of a problem in which the prescribed loading conditions are replaced by equivalent homogeneous conditions, traction-free in the force controlled case and zero displacement (fixed) in the displacement controlled case.

Contact of two elastic bodies

If two elastic bodies make contact at a set of N contact nodes, the first stage is to develop separate finite element models of the two bodies, taking care to locate the contact nodes at the same points on the interface. We use the sign convention of Figure 1 *for each body separately,* which implies that, for example, p_j^1 and p_j^2 are both compressive nodal forces. With this sign convention, when the two bodies are placed in contact, Newton's third law demands that $p_j^1 = p_j^2, q_j^1 = q_j^2$, or equivalently $\boldsymbol{f}_1^{\mathrm{C}} = \boldsymbol{f}_2^{\mathrm{C}}$ so we can conveniently drop the subscripts on these terms. In geometric terms, we can enforce this sign convention by defining local right-handed x, y coordinate systems at each contact node in each body, with the y-axis directed into the body, taking care to number the contact nodes in the same sequence in the two bodies.

In most cases, one of the contacting bodies will be externally supported and the other will have rigid-body degrees of freedom. In such cases, we shall use the index '2' to denote the externally supported body and '1' for the free body. If both bodies are externally supported, the following procedure will work regardless of how the bodies are denoted.

Since the two elastic bodies make contact at a shared set of N contact nodes, we first use the procedure of section 'Single elastic body pressed against a rigid body' to determine the matrices $\boldsymbol{K}_1^{\mathrm{C}}, \boldsymbol{K}_2^{\mathrm{C}}$ and the loading vectors $\boldsymbol{f}_1^w, \boldsymbol{f}_2^w$ for each body separately, defined as in equation (4), and thus

$$\boldsymbol{f}^{\mathrm{C}} = \boldsymbol{f}_1^{w} + \boldsymbol{K}_1^{\mathrm{C}}\boldsymbol{u}_1^{\mathrm{C}} \tag{19a}$$

$$\boldsymbol{f}^{\mathrm{C}} = \boldsymbol{f}_2^{w} + \boldsymbol{K}_2^{\mathrm{C}}\boldsymbol{u}_2^{\mathrm{C}}. \tag{19b}$$

With the sign convention of Figure 1, the opening nodal displacement in the full contact problem is $w_j = w_j^1 + w_j^2$ and the relative tangential (i.e. slip) displacement is $v_j = v_j^1 + v_j^2$, therefore

$$\boldsymbol{u}^{\mathrm{C}} = \boldsymbol{u}_1^{\mathrm{C}} + \boldsymbol{u}_2^{\mathrm{C}}. \tag{20}$$

To create the reduced contact stiffness matrix $\boldsymbol{K}^{\mathrm{C}}$ for the two elastic bodies in contact, we first solve equation (19b) for $\boldsymbol{u}_2^{\mathrm{C}}$, obtaining

$$\left[\boldsymbol{K}_2^{\mathrm{C}}\right]^{-1}\left(\boldsymbol{f}^{\mathrm{C}} - \boldsymbol{f}_2^{w}\right) = \boldsymbol{u}_2^{\mathrm{C}}.$$

We next premultiply by $\boldsymbol{K}_1^{\mathrm{C}}$, obtaining

$$\boldsymbol{K}_1^{\mathrm{C}}\left[\boldsymbol{K}_2^{\mathrm{C}}\right]^{-1}\left(\boldsymbol{f}^{\mathrm{C}} - \boldsymbol{f}_2^{w}\right) = \boldsymbol{K}_1^{\mathrm{C}}\boldsymbol{u}_2^{\mathrm{C}}.$$

Finally, adding this to (19a) and using (20), we have

$$\left[\boldsymbol{K}_1^{\mathrm{C}}\left[\boldsymbol{K}_2^{\mathrm{C}}\right]^{-1} + \boldsymbol{I}\right]\boldsymbol{f}^{\mathrm{C}} = \boldsymbol{K}_1^{\mathrm{C}}\boldsymbol{u}^{\mathrm{C}} + \boldsymbol{f}_1^{w} + \boldsymbol{K}_1^{\mathrm{C}}\left[\boldsymbol{K}_2^{\mathrm{C}}\right]^{-1}\boldsymbol{f}_2^{w},$$

where $\boldsymbol{I}$ is the identity matrix. This equation can be inverted to yield a relation of the form of equation (4), with

$$\boldsymbol{K}^{\mathrm{C}} = \left[\boldsymbol{K}_1^{\mathrm{C}}\left[\boldsymbol{K}_2^{\mathrm{C}}\right]^{-1} + \boldsymbol{I}\right]^{-1}\boldsymbol{K}_1^{\mathrm{C}} \qquad (21)$$

$$\boldsymbol{f}^{w} = \left[\boldsymbol{K}_1^{\mathrm{C}}\left[\boldsymbol{K}_2^{\mathrm{C}}\right]^{-1} + \boldsymbol{I}\right]^{-1}\left(\boldsymbol{f}_1^{w} + \boldsymbol{K}_1^{\mathrm{C}}\left[\boldsymbol{K}_2^{\mathrm{C}}\right]^{-1}\boldsymbol{f}_2^{w}\right). \qquad (22)$$

Notice that if body 1 has rigid-body degrees of freedom, the matrix $\boldsymbol{K}_1^{\mathrm{C}}$ will be singular. However, neither of the inversions in equation (22) will be ill-defined, but the reduced stiffness matrix $\boldsymbol{K}^{\mathrm{C}}$ will also be singular. In fact, it will have the same rank deficiency as $\boldsymbol{K}_1^{\mathrm{C}}$, and the vectors defining the rigid-body modes of $\boldsymbol{K}^{\mathrm{C}}$ will be the same as those for $\boldsymbol{K}_1^{\mathrm{C}}$.

Partitioning the reduced stiffness matrix into normal and tangential components

The preceding operations will generate a $2N \times 2N$ stiffness matrix, whose components are ordered in nodal pairs. For example, the components of the nodal displacement vector $\boldsymbol{u}^{\mathrm{C}}$ will be ordered as $\{v_1, w_1, v_2, w_2, v_3, w_3, \ldots, v_N, w_N\}$. For many purposes it is beneficial to reorder the vectors as

$$\boldsymbol{u}^{\mathrm{C}} = \begin{Bmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{Bmatrix} \qquad (23a)$$

$$\boldsymbol{f}^{\mathrm{C}} = \begin{Bmatrix} \boldsymbol{q} \\ \boldsymbol{p} \end{Bmatrix}, \qquad (23b)$$

where $\boldsymbol{v} = \{v_1, v_2, v_3, \ldots, v_N\}$, $\boldsymbol{p} = \{p_1, p_2, p_3, \ldots, p_N\}$, etc. The stiffness matrix must then be partitioned into submatrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$, such that

$$\begin{Bmatrix} \boldsymbol{q} \\ \boldsymbol{p} \end{Bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{C} \end{bmatrix}\begin{Bmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{q}^{w} \\ \boldsymbol{p}^{w} \end{Bmatrix}. \qquad (24)$$

Notice that the complete reduced stiffness matrix

$$\boldsymbol{K}^{\mathrm{C}} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{C} \end{bmatrix} \qquad (25)$$

must be symmetric, so $\boldsymbol{A}, \boldsymbol{C}$ are symmetric, but $\boldsymbol{B}$ is not generally symmetric. A frictional elastic contact problem is described as 'uncoupled' if and only if the matrix $\boldsymbol{B} = 0$. The coupling implied when $\boldsymbol{B} \neq 0$ has important consequences for the behaviour of frictional elastic systems under periodic loading (see section 'Shakedown').

Algorithms for solving the reduced contact problem

Once the reduced stiffness matrix $\boldsymbol{K}^{\mathrm{C}}$ and the time-varying external nodal forces $\boldsymbol{f}^{w}(t)$ have been determined, the problem becomes equivalent to that of a set of N massless blocks in potential contact with a set of rigid plane obstacles. Various algorithms exist for the numerical solution. At any given point in the loading cycle, each node must be in one of four states which we designate by an integer state variable S_i. The conditions at each node are then defined by the relations

$S_i = 1$	Stick	$w_i = 0;$	$\dot{v}_i = 0;$	$p_i \geqslant 0;$	$\lvert q_i \rvert \leqslant fp_i$
$S_i = 2$	Separation	$p_i = 0;$	$q_i = 0;$	$w_i > 0$	
$S_i = 3$	Forward slip	$w_i = 0;$	$q_i = -fp_i;$	$\dot{v}_i > 0;$	$p_i \geqslant 0$
$S_i = 4$	Backward slip	$w_i = 0;$	$q_i = fp_i;$	$\dot{v}_i < 0;$	$p_i \geqslant 0.$

Notice that we have two equations at each node, so all the unknown nodal forces and displacements can be determined if the states S_i are assumed known.

Gauss–Seidel solution

Ahn and Barber[1] describe an algorithm in which the nodal displacements are initially assumed to have the same values as at the previous time step. The nodes are then examined one by one in a Gauss–Seidel sense such that the state and the nodal displacement *at the node under examination* are updated in accordance with the conditions specified in section 'Algorithms for solving the reduced contact problem'. The algorithm cycles through the entire set of nodes several times until the changes during one such cycle are less than some preset convergence criterion. This algorithm requires the reduced stiffness matrix to be configured as in equation (25).

LCP solution

An alternative approach is to recognize that as long as the states at all the nodes remain the same, the evolution of the discrete solution is defined by linear equations. If the load vector $\boldsymbol{f}^{w}(t)$ is also defined in piecewise linear terms, it is possible to use the above nodal conditions to solve for the time t_i at which an inequality at node i is first violated. If the minimum such value is t^* and if the node at which the resulting violation occurs is node j, we can then use the linear solution for that part of the loading cycle up to $t = t^*$, and the nature of the violation will tell us what state to expect at node j at times slightly larger than t^*. This approach is known as the *LCP* solution, and an algorithm suitable for two-dimensional frictional problems is defined by Bertocchi.[2]

Incomplete contact problems

The discussion so far has centred on 'complete' contact problems, where a defined set of nodes make contact when there are no external loads. Incomplete contact problems (also sometimes called advancing contact problems) arise when there is an initial gap between the bodies, the classical case being Hertzian contact between a quadratic surface and a plane. If the problem is to remain within the scope of linear elasticity, the magnitude of this gap must be small compared with other linear dimensions of the problem, notably the expected dimensions of the contact area.

If the contact is incomplete, the first stage is to define a *nominal contact area* comprising the set of points that might come into contact during the loading process. If s defines a spatial coordinate along the nominal contact area, the initial gap will be a function $g(s)$ and we also require that the derivative $g'(s)$ be small compared with unity. This in turn ensures that we can establish a set of nodes on each of the two surfaces such that the line joining corresponding nodes is approximately perpendicular to each surface, as shown in Figure 2. In the finite element discretization, the length of this line for node j will be denoted by g_j.

It then follows that the nodal gap after deformation is given by $g_i + w_i$ and hence the equations defining the contact algorithm of section 'Algorithms for solving the reduced contact problem' must be modified to read

$S_i=1$	Stick	$w_i+g_i=0;$	$\dot{v}_i=0;$	$p_i\geqslant 0;$	$\lvert q_i\rvert\leqslant fp_i$
$S_i=2$	Separation	$p_i=0;$	$q_i=0;$	$w_i+g_i>0$	
$S_i=3$	Forward slip	$w_i+g_i=0;$	$q_i=-fp_i;$	$\dot{v}_i>0;$	$p_i\geqslant 0$
$S_i=4$	Backward slip	$w_i+g_i=0;$	$q_i=fp_i;$	$\dot{v}_i<0;$	$p_i\geqslant 0.$

With this formulation, the contact stiffness matrix $\boldsymbol{K}^{\mathrm{C}}$ and the external loading vector $\boldsymbol{f}^{w}$ are the same as in the corresponding problem where there is no initial gap.

An alternative approach

An alternative approach, which does not require the contact algorithm to be modified, is to redefine the function $\boldsymbol{f}^{w}$ as the set of nodal forces needed to establish contact with no slip at all the nodes in the

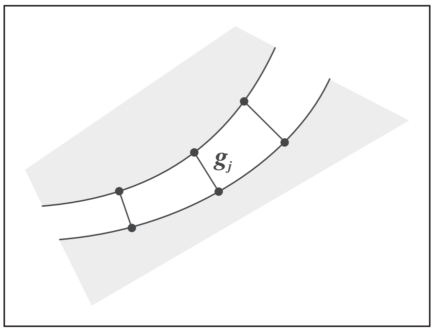

Figure 2. Definition of the nodal gap g_j.

nominal contact area – including closing the initial gap. The easiest way to determine these forces is to superpose the solution of two separate problems:

1. the forces $\boldsymbol{f}^{w0}$ that would be generated by the external loads if there had been no initial gap $\boldsymbol{g} = 0$, that is, if the contact had been complete, and
2. the forces $\boldsymbol{f}^{wg}$ that are required to close the initial gap in the absence of external loads. These are readily obtained using equation (24) as

$$\begin{Bmatrix} \boldsymbol{q}^{wg} \\ \boldsymbol{p}^{wg} \end{Bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{C} \end{bmatrix} \begin{Bmatrix} \boldsymbol{0} \\ -g \end{Bmatrix}. \tag{26}$$

The required reduced external forces are then

$$\boldsymbol{f}^{w} = \boldsymbol{f}^{w0} + \boldsymbol{f}^{wg}. \tag{27}$$

The two methods are equivalent. To establish this, define a new contact nodal displacement vector $\boldsymbol{u}^*$, such that

$$\boldsymbol{u}^* = \boldsymbol{u}^{\mathrm{C}} + \boldsymbol{u}^{g}, \tag{28}$$

where $\boldsymbol{u}^{\mathrm{C}}$ is the nodal displacement excluding the initial gap, and $\boldsymbol{u}^{g}$ is the nodal displacement corresponding to the initial gap, which is defined as

$$\boldsymbol{u}^{g} = \begin{Bmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{0} \\ -\boldsymbol{g} \end{Bmatrix}. \tag{29}$$

We then have

$$\boldsymbol{f}^{w} = \boldsymbol{f}^{w0} + \boldsymbol{K}^{\mathrm{C}}\boldsymbol{u}^{g}, \tag{30}$$

and hence

$$\boldsymbol{f}^{\mathrm{C}} = \boldsymbol{f}^{w} + \boldsymbol{K}^{\mathrm{C}}\boldsymbol{u}^{\mathrm{C}} = \boldsymbol{f}^{w0} + \boldsymbol{K}^{\mathrm{C}}(\boldsymbol{u}^{\mathrm{C}} + \boldsymbol{u}^{g}) = \boldsymbol{f}^{w0} + \boldsymbol{K}^{\mathrm{C}}\boldsymbol{u}^*. \tag{31}$$

The contact boundary condition at node i is $w_i = 0$, which therefore implies $w_i^* = -g_i$. Thus, we can use the original contact algorithm with the modified value of $\boldsymbol{f}^{w}$, or the modified algorithm (with contact boundary condition $w_i^* = -g_i$) and the original value $\boldsymbol{f}^{w} = \boldsymbol{f}^{w0}$.

Example problems

We will now look at three example problems. Each was solved in two ways: first, by using the commercial finite element programme ABAQUS, incorporating contact elements, and using the option of an artificial stiffness between sticking nodes, to help linearize the problem. Second, the same model was generated with ABAQUS and the stiffness matrix (or matrices if both contacting bodies have a finite stiffness) abstracted and processed in the way described in previous sections. In the latter case, the contact problem itself was solved within MATLAB, again as described, so that the Signorini conditions were incorporated precisely in the solution, using the Gauss–Seidel algorithm.

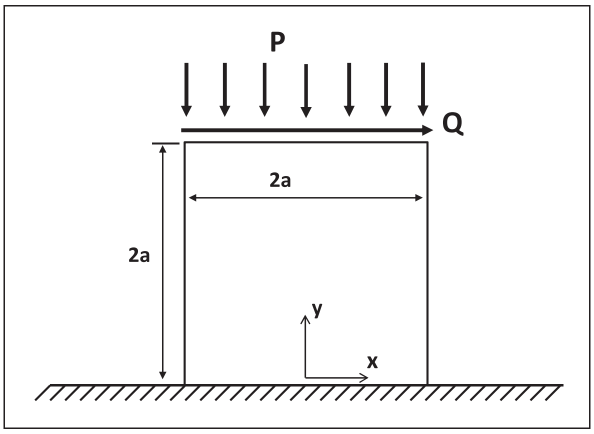

Figure 3. A diagram of the geometry considered in Case 1, which shows the applied loads and the *x, y* coordinate set that is used.

Case 1: Single elastic body pressed against a rigid counterface

Figure 3 shows a square elastic block, of side $2a$, pressed onto a flat surface, which is taken to be perfectly rigid. The block is first loaded with a total normal force P, which is distributed uniformly along the top surface of the block. A uniformly distributed (in space) shear traction, with a total resultant force of $Q(t)$, is subsequently applied to the top of the block, and its magnitude is monotonically increased in time. At the same time that the shear $Q(t)$ is applied, a further pressure distribution, varying linearly with position and proportional in magnitude to the shear traction, is applied to the top of the block, so that a moment is developed which renders the shear load statically equivalent to one applied along the plane of the contact. This device is employed so that the detailed way in which the loads are exerted does not significantly influence the traction distribution along the contact interface. The coefficient of friction f is initially taken as 0.2.

A finite element model of the block was developed having 5000 nodes and with 101 nodes present along the line of the contact interface. Therefore, the full stiffness matrix for the problem $\boldsymbol{K}$ is of size $10,000 \times 10,000$, whilst the statically reduced matrix $\boldsymbol{K}^{C}$ is only 202×202. The results of the calculation are given in Figure 4. Only one set of figures is given both here and for all subsequent examples, because the results provided by ABAQUS and those derived from the statically reduced matrix are indistinguishable. Figure 4(a) shows the interfacial slip displacement $v(x)$ present when the normal load P alone is applied. As expected, the distribution of slip displacement $v(x)$ is antisymmetric and there is a significant

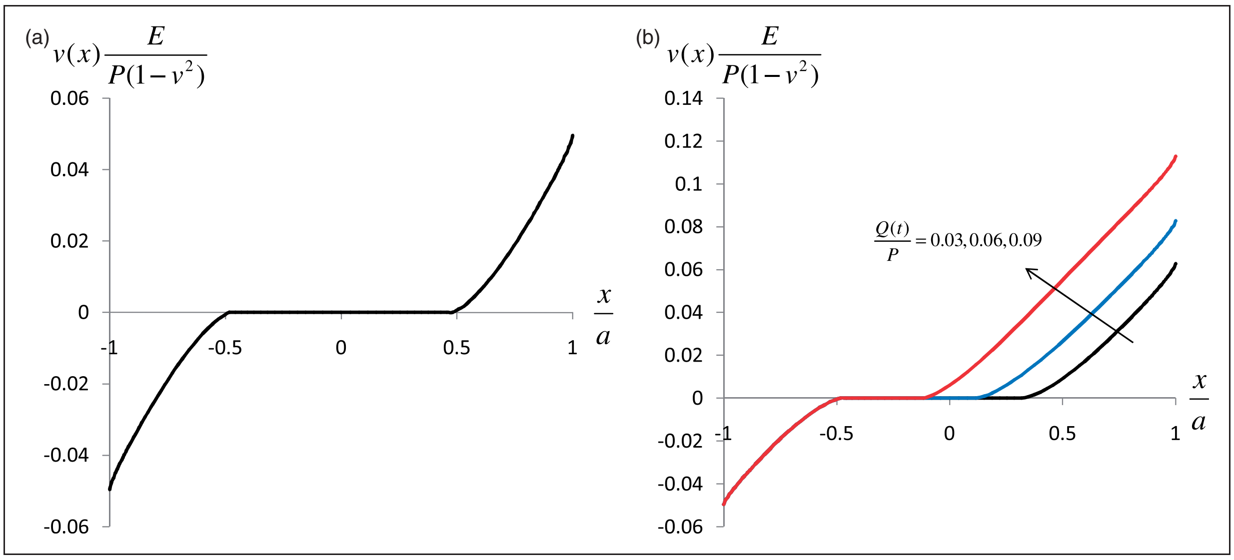

Figure 4. Plots of the (normalized) slip displacement *v*(*x*) along the contact interface for the case of an elastic block in frictional contact with a rigid plane obstacle, at a coefficient of friction *f* of 0.2, when (a) the normal load *P* is first applied and (b) at several levels of shear load *Q*(*t*).

central stick region. As the shear load $Q(t)$ is increased from zero, what is now the trailing edge sticks upon application of an infinitesimal load. This locks in the reverse slip displacement that is already present, and this behaviour continues as the shearing load $Q(t)$ is increased in value. At the same time, the forward slip zone attached to the leading edge increases in size as the shear load $Q(t)$ is raised, and this is shown in Figure 4(b). In Figure 4, the normalization of the displacement $v(x)$ is with respect to the 'plane strain modulus', where E is Young's modulus and ν is Poisson's ratio. On this scale, any very local contact corner oscillatory behaviour is not apparent.

Although this is, of course, very machine specific, the reduction in computer time between the conventional solution and that found from the reduced stiffness matrix was a factor of 60 here.

Case 2: Elastic square body pressed onto elastically similar half-plane

The problem depicted in Figure 3 is treated again, but this time the rigid plane obstacle is replaced by a half-plane with identical elastic properties to the other contacting body, and the coefficient of friction f is set to 0.4. The so-called half-plane is, of course, approximated in ABAQUS as a finite block, but larger than the contacting square by a factor of 200. The stiffness matrix for the square block was the same as above, and the 'half-plane' represented by a stiffness matrix $\boldsymbol{K}$ of size $17,000 \times 17,000$, but the reduced contact stiffness matrix $\boldsymbol{K}^{C}$ remained of the same size as in Case 1. The slip displacement $v(x)$ resulting from the application of a normal load P is given in Figure 5(a), and is qualitatively similar to that found in Case 1. Upon application of an infinitesimal shear load $Q(t)$, what is now the trailing edge of the contact again instantaneously sticks, and as the shear load $Q(t)$ is increased in value the leading edge slip zone increases monotonically in size, as shown in Figure 5(b). At the trailing edge a tiny region of separation might be expected,[3] but it is not apparent from the figure.

In this case, the computation time required to solve the evolutionary contact problem was reduced by a factor of 65 when the reduced contact stiffness matrix was used as compared with running the job in ABAQUS.

Case 3: Incomplete contact

In the two problems looked at so far, the size of the contact is known prior to the application of normal load, which facilitates the solution a great deal. We will now look at an incomplete contact, shown in Figure 6, where the front face of the elastic block is now in the form of a circular arc of radius R, which in this case is 100 a, with the coefficient of friction f still set to 0.4. Here, a normal load with a total resultant force of $P(t)$ is applied to the top of the block, and increased monotonically in time, up to a maximum value $P_{\max}$. The relative surface normal separation $w(x)$ of opposing contact nodes in the two bodies, including that of the undeformed bodies, is shown in Figure 7(a), as a function of applied normal load $P(t)$. Lastly, Figure 7(b) shows the contact pressure distribution $p(x)$ which is, of course, semi-elliptical in form, again as a function of normal load $P(t)$. It is again emphasized that the conventional finite element solution and a solution obtained from the reduce contact stiffness matrix gave results which were indistinguishable.

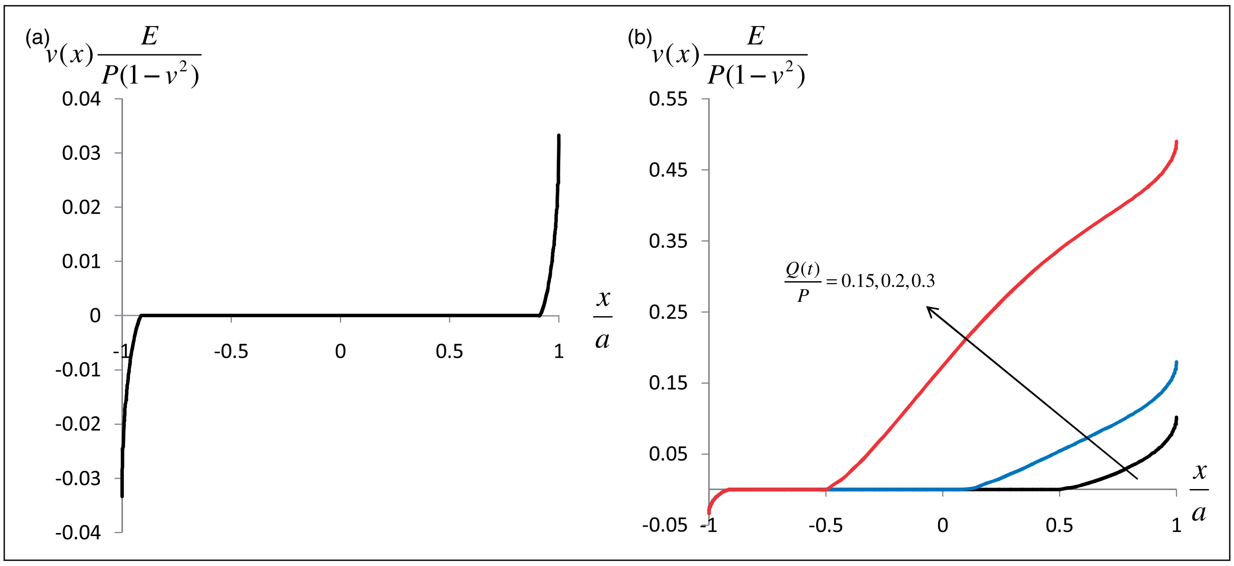

Figure 5. Plots of the (normalized) slip displacement $v(x)$ along the contact interface for the case of an elastic block in frictional contact with an elastically similar half-plane, at a coefficient of friction f of 0.4, when (a) the normal load P is first applied and (b) at several levels of shear load $Q(t)$.

In this case, the computation time required to solve the evolutionary contact problem was reduced by a factor of 40.

Other applications of the reduced stiffness matrix

The above examples show that frictional contact problems with time-varying external loads (so-called 'rate problems') can be solved using the reduced stiffness matrix with a considerable saving in computational effort. However, the method also enables us to formulate and solve other categories of problem that cannot conveniently be treated using conventional finite element methods.

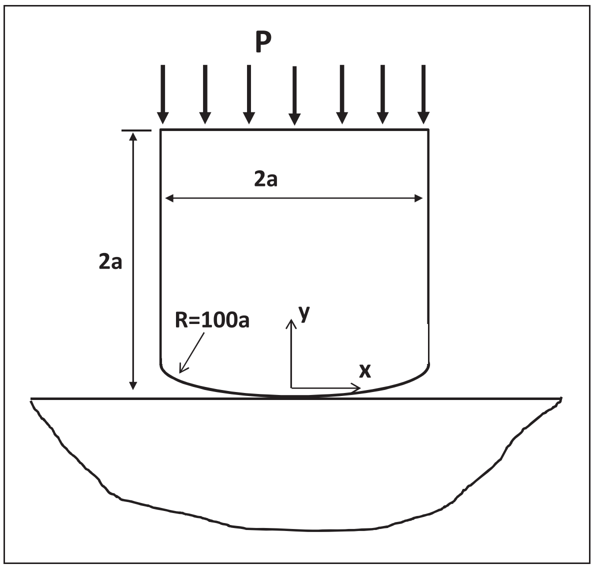

Figure 6. A diagram of the geometry considered in Case 3, which shows the applied load and the *x*, *y* coordinate set that is used.

The critical coefficient of friction

It is well known that the quasi-static discrete rate problem can be mathematically ill-posed if the coefficient of friction is sufficiently high.[4–6] In particular, there may occur steps in the loading trajectory where a strict interpretation of the friction law in the next time step permits multiple solutions, or (worse) no solution. In the latter case, simple examples suggest that the physical system will 'jump' to a new state with a discontinuous change in displacement, governed by the elastodynamic equations.[7,8] Klarbring[9] has shown that for two-dimensional problems, the rate problem is well posed if and only if all matrices of the form

$$\boldsymbol{A} + \Lambda \boldsymbol{B} \tag{32}$$

are P-matrices (i.e. all principal minors of the matrix are positive definite), where Λ is a diagonal matrix whose elements are $\pm f$, and the matrices $\boldsymbol{A}$, $\boldsymbol{B}$ are defined in equation (24) and require a previous determination of the reduced stiffness matrix $\boldsymbol{K}^{\mathrm{C}}$. Notice that if a conventional finite element solution is attempted above the critical coefficient of friction, a solution of doubtful meaning will probably be returned and no warning message will be provided.

At high coefficients of friction it is also possible for frictional elastic systems to become wedged, meaning that they remain in a state of stress with non-zero slip displacements when all external loads have been

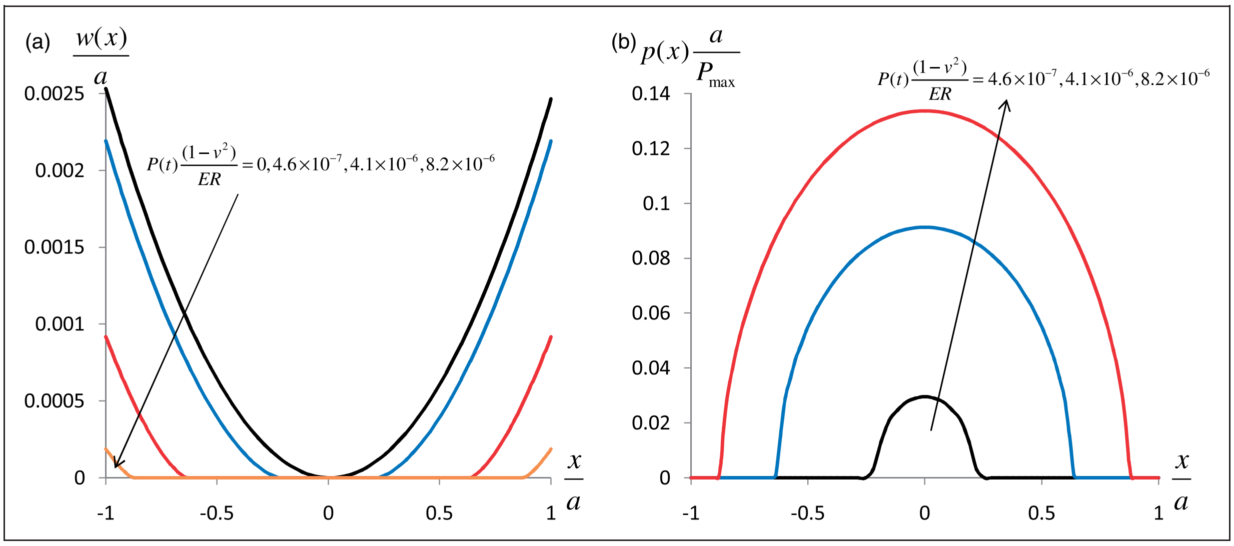

Figure 7. Plots of (a) the (normalized) extent of separation *w*(*x*) along the contact interface and (b) the (normalized) contact pressure distribution *p*(*x*), for an elastic block of radius $R = 100\,a$ in frictional contact with an elastically similar half-plane, at a coefficient of friction *f* of 0.4, as the normal load *P*(*t*) is raised monotonically, up to a maximum value P_{max}, which, once normalized by $ER / (1 - \nu^2)$, takes on the value 8.2×10^{-6} in this particular instance.

removed. Barber and Hild[10] discuss strategies for determining whether a system can become wedged, which also depend on a previous determination of the reduced contact stiffness matrix.

Shakedown

If the external loading $\boldsymbol{P}(t)$ is periodic in time, the slip displacements during the first few cycles modify the behaviour in subsequent cycles, and in some circumstances can lead to a state of *shakedown*, where no slip occurs in the steady state. A necessary condition for shakedown is that there exist a time-independent tangential displacement vector $\boldsymbol{v}^s$ such that $|q_i| < fp_i$ for all nodes i and all times t during the loading, and hence

$$-f(p_i^w(t) + B_{ij}v_j) < q_i^w(t) + A_{ij}v_j < f(p_i^w(t) + B_{ij}v_j) \tag{33}$$

for all i, t.

Klarbring et al.[11] have shown that this is a necessary and sufficient condition for shakedown if and only if the matrix $\boldsymbol{B} = 0$. It has been conjectured that this condition is also sufficient to guarantee that the time-varying components in the steady-state response of such a system should be unique, even above the shakedown limit.[12]

If the system is coupled ($\boldsymbol{B} \neq 0$), equation (33) defines a necessary but not sufficient condition for shakedown, so there will generally exist a class of periodic loading scenarios for which the occurrence of shakedown depends on the initial conditions. A strategy for defining the limits of this class is discussed by Ahn et al.[13] All of these procedures depend on a prior determination of the contact stiffness matrix.

Dislocation solutions

A popular method for the analytical solution of crack and contact problems is to write the perturbation in the stress field as a convolution integral on the solution for a dislocation at the interface, in which case the contact or separation conditions at the crack or the interface define an integral equation for the unknown dislocation distribution.[14] This method can be applied to any domain for which the solution for a concentrated dislocation is known, but its extension to more general problems requires that the dislocation solution be determined numerically.

The present procedure provides a simple way to obtain such solutions. We extend the crack line to the edge of the body, mesh the body such that this extended crack line forms a contact interface and find the corresponding reduced stiffness matrix $\boldsymbol{K}^{\mathrm{C}}$. To determine the solution for a unit dislocation at node i, we then impose the displacement conditions $\boldsymbol{u}_i^{\mathrm{C}} = \{1, 0\}^T$ or $\{0, 1\}^T$ for all contact nodes between i and the free end of the cut and $\boldsymbol{u}_i^{\mathrm{C}} = 0$ for contact nodes between i and the 'unextended' end of the crack.

Conclusions

In this paper, we have shown how to determine the reduced contact stiffness matrix, starting from the full stiffness matrix obtained from separate finite element models of the contacting bodies. Instructions for performing these operations in MATLAB are given in the appendix and a more detailed set of instructions and downloadable software can be accessed at the website www.eng.ox.ac.uk/stress.

Once the contact stiffness matrix is determined, it can be used to develop a solution of the evolutionary contact problem using a direct implementation of the Signorini and Coulomb friction contact inequalities. This is significantly more efficient than a direct finite element solution of the same problem. For the examples treated in section 'Example problems', computational time is significantly reduced. However, another advantage of the proposed method is that the nonlinearities associated with the contact conditions are visible to and under the control of the user. This facilitates trouble shooting in complex problems and also permits the user to introduce alternative friction laws or experiment with different iterative strategies for the solution.

The reduced stiffness matrix is also a useful starting point for other more general investigations, such as the determination of the critical coefficient of friction for the problem to be well posed, or the range of alternating loads below which the system can shake down. With conventional time-marching finite element solutions, these problems can only be explored by running numerous particular cases and this is likely to be prohibitively computer intensive. However, when the number of remaining degrees of freedom is reduced to the number of contact nodes, it becomes practicable to use optimization codes to determine, for example, the optimal set of time-independent slip displacements that will permit the maximum amplitude of periodic loading without incurring further slip.

Funding

A Thaitirarot would like to thank The Royal Thai Government and Jesus College for their scholarship and financial support. RC Flicek would like to thank Rolls-Royce plc and the Technology Strategy Board for financial support under the programme SILOET-II.

Acknowledgements

We would like to thank Yong Hoon Jang and Young Ju Ahn for useful discussions. We wish to thank Rolls-Royce plc for granting permission to publish this work.

References

1. Ahn YJ and Barber JR. Response of frictional receding contact problems to cyclic loading. *Int J Mech Sci* 2008; 50: 1519–1525.

2. E. Bertocchi. *Selected topics on the plane elastic contact with friction*. PhD Dissertation, Department of Mechanical and Civil Engineering, University of Modena and Reggio Emilia, Italy, 2009, see Chapter 1.
3. Churchman CM and Hills DA. General results for complete contacts subject to oscillatory shear. *J Mech Phys Solids* 2006; 54(6): 1186–1205.
4. Kikuchi N and Oden JT. *Contact problems in elasticity: a study of variational inequalities and finite element methods*. Philadelphia: SIAM, 1988.
5. Klarbring A. Examples of non-uniqueness and non-existence of solutions to quasi-static contact problems with friction. *Ingenieur-Archiv* 1990; 60: 541–529.
6. Martins JAC, Montiero Marques MDP and Gastaldi F. On an example of non-existence of solution to a quasi-static frictional contact problem. *Eur J Mech A-Solid* 1994; 13: 133–113.
7. Cho H and Barber JR. Dynamic behavior and stability of simple frictional systems. *Math Comput Model* 1998; 28: 53–37.
8. Ahn YJ. Discontinuity of quasi-static solution in the two-node coulomb frictional system. *Int J Solids Struct* 2010; 47: 2866–2871.
9. Klarbring A. Contact, friction, discrete mechanical structures and discrete frictional systems and mathematical programming. In: P Wriggers and P Panagiotopoulos (eds) *New developments in contact problems*. Wien: Springer, 1999, pp.55–100.
10. Barber JR and Hild P. On wedged configurations with Coulomb friction. In: P Wriggers and U Nackenhorst (eds) *Analysis and simulation of contact problems*. Berlin: Springer-Verlag, 2006, pp.205–213.
11. Klarbring A, Ciavarella M and Barber JR. Shakedown in elastic contact problems with coulomb friction. *Int J Solids Struct* 2007; 44: 8355–8365.
12. Barber JR. Frictional systems subjected to oscillating loads. *Ann Solid Struct Mech* 2011; 2: 45–55.
13. Ahn YJ, Bertocchi E and Barber JR. Shakedown of coupled two-dimensional discrete frictional systems. *J Mech Phys Solids* 2008; 56: 3433–3440.
14. Hills DA, Kelly PA, Dai DN, et al. *Solution of crack problems – the distributed dislocation technique*. Dordrecht, The Netherlands: Kluwer Academic, 1996.

Appendix

In the interest of brevity, we assume that the reader is sufficiently familiar with ABAQUS to be able to create and mesh parts, create node sets, run jobs and most other basic operations. (If the reader is unfamiliar with ABAQUS, please consult the URL www.eng.ox.ac.uk/stress, which provides instructions detailing how to create a basic model and perform the required operations.)

The main pieces of information that are to be extracted from the ABAQUS model are as follows: (i) the global stiffness matrix $\boldsymbol{K}$ and (ii) several node sets, which can be used to group each degree of freedom in the finite element model into one of three categories; contact nodes, externally loaded nodes or internal nodes. These data are then imported into MATLAB where the static reduction and all subsequent analysis are performed.

Create node sets in ABAQUS

First, the reader must group all the nodes in the ABAQUS model into several node sets according to the model's features. These node sets will then appear in the input file **in the order in which they are created**, not in alphabetical order as they appear in ABAQUS. (It is helpful to name the first node set with a prefix, e.g. `aa-[node set name]`. This makes it easier to locate the node sets in the input file, e.g. by searching for the string `*Nset, nset=aa-`.) The nodes are to be grouped according to whether they: (i) lie along a contact region, (ii) are externally loaded, (iii) have some boundary condition applied to them in ABAQUS or (iv) do not fall into any of the above categories. If several independent loads are applied to different sets of nodes, or there are multiple contact regions, or there are different types of boundary conditions applied to different sets of nodes, etc., then each of these regions should be included in a **separate** node set, such that each node set contains only those nodes that can be treated in the same way, for example, all nodes are internal nodes, or all nodes have degree of freedom 1 fixed but degree of freedom 2 left free. These node sets can then be consolidated in MATLAB, and categorized as internal, externally loaded or contact nodes.

Export ABAQUS global stiffness matrix

To export the global stiffness matrix from an ABAQUS model, the reader must create a job file from the model that is to be reduced, and write an input file for this job. The input file must then be modified. To do this, search the input file for the string `** STEP:`. For ABAQUS version 6.11, the following code must be added directly above this line of code.

```
*STEP, name=exportmatrix
*MATRIX GENERATE, STIFFNESS
*MATRIX OUTPUT, STIFFNESS, FORMAT=MATRIX INPUT
*END STEP
**
```

For versions of ABAQUS prior to 6.11, the third line of the above code must be excluded, but the other four lines of code must still be added to the input file. The modified input file must then be run in ABAQUS, and when this is done the global stiffness matrix will be output in a .mtx file named [NAME_OF_INPUT_FILE]_STIF1.mtx.

Import node sets into MATLAB

To import the node sets into MATLAB, first, save a copy of the job input file under a different name, and, in this file, delete all information except for the node sets, such that the first line of the file begins with the string *Nset, nset=, and ends after the list of nodes numbers corresponding to the last node set that was created. This modified input file can be read with the MATLAB command read_input_file = fopen('NAME_OF_INPUT_FILE.inp',','), which stores the file that is being read in the variable read_input_file. The code below then imports the first node set into the MATLAB variable abaqus_node_set.

```
% read line containing the node set name starting with '*Nset, nset='
read_header = textscan(read_input_file, ' %s',1,'delimiter', '\n');
% check if the string ', generate' appears in the header line
nodeset_header = strfind(read_header{1},', generate');
% if string ', generate' does not appear
if numel(nodeset_header{1}) == 0
    temp_textscan_output = textscan(read_input_file,...
        '%d','delimiter',',');
abaqus_node_set = temp_textscan_output{1}';
% if string ', generate' does appear
elseif numel(nodeset_header{1}) == 1
    temp_textscan_output = textscan(read_input_file, ...'
        'delimiter',',');
abaqus_node_info = temp_textscan_output{1}';
abaqus_node_set = zeros(1,1+(abaqus_node_info(1,2)-...
        abaqus_node_info(1,1))/abaqus_node_info(1,3));
for i = 1:length(abaqus_node_set)
        abaqus_node_set(1,i) = abaqus_node_info(1,1) +...
            (i-1)*(abaqus_node_info(1,3));
    end
else
    error('Error: the string ", generate" appears more than once')
end
```

This code imports one node set at a time. To import all the node sets in the modified input file, this code can either be repeated several times, or used to create a MATLAB function that is then called several times. Once all the node sets have been imported from the input file, the input file can be closed using the command fclose(read_input_file).

Convert 'ABAQUS nodes' to 'MATLAB nodes'

In ABAQUS, each node has two degrees of freedom. However, in MATLAB we define a new convention in which each *Mnode* has only **one degree of freedom**, which we define as

$$MnodeDOF1 = 2(Anode - 1) + 1 \tag{34a}$$

$$MnodeDOF2 = 2(Anode - 1) + 2 \tag{34b}$$

where *Anode* is a shorthand for the node number given by ABAQUS, and where *MnodeDOF1* and *MnodeDOF2* are the 'MATLAB node numbers', or *Mnodes*, for the degree of freedom in direction 1 and direction 2, respectively. If both degrees of freedom from a set of *Anodes* are to be included in the corresponding set of *Mnodes*, then the conversion is performed by the following code, which stores the *Mnodes* in the MATLAB variable matlab_node_set.

```
matlab_node_set = zeros(1,2*length(abaqus_node_set));
for j = 1:length(abaqus_node_set)
```

```
    matlab_node_set(1, 2*(j-1)+1) = 2.*(abaqus_node_set(1,j)-1)+1;
    matlab_node_set(1, 2*(j-1)+2) = 2.*(abaqus_node_set(1,j)-1)+2;
end
```

If only one of the degrees of freedom from a set of *Anodes* is to be included in the corresponding set of *Mnodes*, then the following code must be used, where the string XXXXDOFXXXX must be replaced either by the value 1 or 2, according to which degree of freedom is to be retained. (An example situation in which this would be required is if a boundary condition is applied in ABAQUS, such that, for some set of nodes, say, degree of freedom 1 is fixed, but degree of freedom 2 is left free. In this case, degree of freedom 1 should be left out of the reduction, and only degree of freedom 2 should be retained, and should be included in the set of internal *Mnodes*.)

```
matlab_node_set = zeros(1,length(abaqus_node_set));
for j = 1:length(abaqus_node_set)
    matlab_node_set(1,j) = 2.*(abaqus_node_set(1,j)-1)+ XXXXDOFXXXX;
end
```

Import global stiffness matrix into MATLAB

The code provided below will import the global stiffness matrix from an .mtx file into MATLAB. Note that, the string NAME_OF_STIFFNESS_MATRIX must be replaced by the name of the .mtx file, and this file must be located in MATLAB's working directory.

```
% Read the abaqus .mtx file into a [n x 5] sparse matrix in matlab
abaqus_stiffness_matrix = dlmread('NAME_OF_STIFFNESS_MATRIX.mtx');
%merge columns 1 and 2, and turn into Mnodes
matlab_nodes(:,1) = 2*(abaqus_stiffness_matrix(:,1)-1)+...
    abaqus_stiffness_matrix(:,2);
%merge columns 3 and 4, and turn into Mnodes
matlab_nodes(:,2) = 2*(abaqus_stiffness_matrix(:,3)-1)+...
    abaqus_stiffness_matrix(:,4);
% extract stiffness values, and store in a double length vector
stiffness_values = [abaqus_stiffness_matrix(:,5);...
    abaqus_stiffness_matrix(:,5)];
% compile the stiffness matrix using the new node numbering convention
[matlab_matrix_indices, abaqus_stiffness_value_index] = unique(...
    [matlab_nodes; matlab_nodes(:,2) matlab_nodes(:,1)], 'rows');
K = accumarray(matlab_matrix_indices,...
    stiffness_values(abaqus_stiffness_value_index), [], @max, [], true);
```

The global stiffness matrix is output to the variable K in sparse matrix format. The stiffness values are stored as element values in the matrix, and are arranged such that the row and column numbers represent the numbers of the two *Mnodes* that the stiffness value connects.

Static reduction

The static reduction procedure begins by partitioning the ***K*** matrix and creating and partitioning the ***L*** matrix. To do this, the various sets of *Mnodes* must be grouped into three categories: (i) internal, (ii) externally loaded and (iii) contact nodes. Once the node sets are consolidated in this way, and stored in the MATLAB variables internal_nodes, ext_loaded_nodes, contact_nodes, the following code will create the ***K*** and ***L*** submatrices.

```
KII = K(internal_nodes, internal_nodes);
KIE = K(internal_nodes, ext_loaded_nodes);
KIC = K(internal_nodes, contact_nodes);
KEE = K(ext_loaded_nodes, ext_loaded_nodes);
KEC = K(ext_loaded_nodes, contact_nodes);
KCC = K(contact_nodes, contact_nodes);
KEI = KIE'; KCI = KIC'; KCE = KEC';
L = full([KEE,KEC;KCE,KCC]-[KEI;KCI]*(KII\[KIE,KIC]));
LEE = L(1:length(KEE(:,1)), 1:length(KEE(:,1)));
```

```
LEC = L(1:length(KEE(:,1)), length(KEE(:,1))+1:length(L(:,1)));
LCE = LEC';
LCC = L(length(KEE(:,1))+1:length(L(:,1)),...
    length(KEE(:,1))+1:length(L(:,1)));
```

If the external loads are applied in force control, then $\boldsymbol{K}^C$ and $\boldsymbol{K}^E$ can be computed using the code below.

```
KC = LCC-LCE*(LEE\LEC);
KE = LCE/LEE;
```

If the external loads are applied in displacement control, then $\boldsymbol{K}^C$ and $\boldsymbol{K}^E$ are found using the following code.

```
KC = LCC;
KE = LCE;
```

Create the $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ *matrices from* $\boldsymbol{K}^C$

The following MATLAB code will repartition $\boldsymbol{K}^C$ and create the $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ matrices.

```
degree_of_freedom_one = zeros(1,length(KC)/2);
degree_of_freedom_two = zeros(1,length(KC)/2);
for j = 1:length(KC)/2
    degree_of_freedom_one(1,j) = 2*(j-1)+1;
    degree_of_freedom_two(1,j) = 2*(j-1)+2;
end
A = KC(degree_of_freedom_one, degree_of_freedom_one);
B = KC(degree_of_freedom_two, degree_of_freedom_one);
C = KC(degree_of_freedom_two, degree_of_freedom_two);
```

Create $\boldsymbol{f}^w$ *for a single body problem*

To create $\boldsymbol{f}^w$, first, the load distribution must be determined. MATLAB code for some common load distributions is provided below. It does not matter whether the problem is force or displacement controlled, this procedure remains the same.

```
load_zero = zeros(length(KE(1,:))/2,1);
load_constant = ones(length(KE(1,:))/2,1);
load_linear = zeros(length(KE(1,:))/2,1); %preallocate
for i = 0:length(load_linear)-1
    load_linear(i+1,1) = -1 + 2*(i/(length(load_linear)-1));
end
```

The load distributions above are half the length of the $\boldsymbol{K}^E$ matrix. Half of the $\boldsymbol{K}^E$ matrix corresponds to degree of freedom 1, and the other half to degree of freedom 2. Thus, a load distribution must be selected for the set of nodes comprising each degree of freedom, and saved as the variables `load_distribution_DOF_one` and `load_distribution_DOF_two`. The following code will then form a vector containing the full load distribution, and create $\boldsymbol{f}^w$.

```
load_distribution = zeros(2*length(load_distribution_DOF_one),1);
for j = 1:(length(KE(1,:))/2)
    load_distribution(2*(j-1)+1, 1) = load_distribution_DOF_one(j,1);
    load_distribution(2*(j-1)+2, 1) = load_distribution_DOF_two(j,1);
end
fw = KE*load_distribution;
```

Reorganize $\boldsymbol{f}^w$ *to match the* $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ *matrices*

The following MATLAB code will reorganize $\boldsymbol{f}^w$ to be compatible with the $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ matrices, and output the result in the variable `load_vector`.

```
load_vector = zeros(length(fw), 1);
for j = 1:length(load_vector)/2
  load_vector(j, 1) = fw(2*(j-1)+1, 1);
  load_vector(length(load_vector)/2+j, 1) = fw(2*(j-1)+2, 1);
end
```

Create $\boldsymbol{K}^C$ for contact between two elastic bodies

The following code will merge the $\boldsymbol{K}^C$ matrices of two elastic bodies, where the $\boldsymbol{K}^C$ matrices are named `body_1_KC` and `body_2_KC`. The result is output in the variable `KC_two_body`.

```
KC_two_body = (body_1_KC/body_2_KC + eye(size(body_2_KC)))\body_1_KC;
```

Create $\boldsymbol{f}^w$ for contact between two elastic bodies

To create $\boldsymbol{f}^w$ for a contact problem between two elastic bodies, the $\boldsymbol{K}^C$ matrices are to be named as in the previous section, and the $\boldsymbol{f}^w$ matrices are to be named `body_1_fw` and `body_2_fw`. If the loads corresponding to each individual $\boldsymbol{f}^w$ vector are to be independently applied, then the vectors should be created by running the following code with only one vector input as either `body_1_fw` or `body_2_fw`, and the other vector should be input as a null vector.

```
load_vector_two_body = (body_1_KC/body_2_KC +...
    eye(size(body_2_KC))) \(body_1_fw + body_1_KC*(body_2_KC\body_2_fw));
```