INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2002; **55**:101–125 (DOI: 10.1002/nme.568)

A monolithic smoothing-gap algorithm for contact-impact based on the signed distance function

T. Belytschko[1,*,†], W. J. T. Daniel[2] and G. Ventura[3]

[1]*Department of Mechanical Engineering, Northwestern University, 2145 Sheridan, Evanston, lL, U.S.A.*
[2]*Department of Mechanical Engineering, University of Queensland, Brisbane, Australia*
[3]*Politecnico di Torino, Department of Structural and Geotechnical Engineering, Torino, Italy*

SUMMARY

A new algorithm has been developed for smoothing the surfaces in finite element formulations of contact-impact. A key feature of this method is that the smoothing is done implicitly by constructing smooth signed distance functions for the bodies. These functions are then employed for the computation of the gap and other variables needed for implementation of contact-impact. The smoothed signed distance functions are constructed by a moving least-squares approximation with a polynomial basis. Results show that when nodes are placed on a surface, the surface can be reproduced with an error of about one per cent or less with either a quadratic or a linear basis. With a quadratic basis, the method exactly reproduces a circle or a sphere even for coarse meshes. Results are presented for contact problems involving the contact of circular bodies. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: contact; finite elements; impact; smoothing

1. INTRODUCTION

This paper describes an algorithm that replaces a rough surface, such as generally found in finite element models of solids, with a smooth surface and at the same time computes the gap function that is needed for the implementation of contact-impact techniques. We call it monolithic because the smoothing operation is not separate from the gap computation; instead, during the gap computation, a smoothed surface is implicitly constructed.

The basic concept is to construct smoothed signed distance functions for the contacting bodies, with the surfaces of the bodies described by the zero isosurfaces of these functions. A moving least-squares approximation is used for the signed distance function so that the contact algorithm perceives the surface as continuously differentiable. In contrast to most previous smoothing algorithms, where a smooth approximation based on splines or other techniques

*Correspondence to: T. Belytschko, Department of Mechanical Engineering, Northwestern University, 2145 Sheridan, Evanston, IL 60208, U.S.A.
†E-mail: tedbelytschko@northwestern.edu

Contract/grant sponsor: Office of Naval Research.

Received 1 March 2002
Revised 15 March 2002

is constructed first, followed by an evaluation of the gap, in this method the gap of any point or node is computed directly. The smoothing is implicit in the computation of the gap, and there is no need to define surface normals; they emerge naturally from the algorithm (except for a sign). These attributes markedly decrease the complexity of smoothing contact-impact. Furthermore, the algorithm is readily applicable to arbitrary surface meshes, such as combinations of quadrilateral and triangular surface elements and can readily be applied to both two- and three-dimensional problems.

The algorithm may be made almost completely automatic for elements of any order. Smoothing can be of different orders: both linear and quadratic basis function smoothing is considered here. Moreover, features of the geometry, such as rounded corners and fillets, that are not explicitly modelled in the finite element model can be treated naturally within this method.

Smoothed contact algorithms have been described by Kikuchi [1] and Wriggers *et al.* [2]. Smooth interpolations have been given by Wriggers *et al.* [3] in two dimensions and by El-Abassi *et al.* [4] with cubic splines. Recently, Puso and Laursen [5] reported a method where Gregory patches, a form of spline interpolation, were used to smooth both frictionless and frictional contact in three dimensional problems. Other aspects of the contact problem that are relevant but not examined here are interface properties and stability, see McDevitt and Laursen [6], and References [7, 8]. Other relevant papers are Jones and Papadopoulos [9], who introduced pressure smoothing, Kloostermann *et al.* [10] who used a barrier method, Bajer and Demkowicz [11] and Stupkiewicz [12].

The outline of this paper is as follows. In Section 2 we summarize the notation to be used and the features of the contact problem relevant to this paper. In Section 3 we describe the surface smoothing procedure. Section 4 describes its implementation. In Section 5 we examine the accuracy of this method in fitting certain analytic shapes and describe the results of some contact problems involving smooth circular bodies modelled by piecewise linear elements.

2. NOTATION AND PRELIMINARIES

Notation and contact surface conditions. We first briefly review the contact-impact problem and the notation to be used in this paper, see References [13, 14] or [15]. This algorithm applies to an arbitrary number of bodies, but for purposes of simplicity, we limit our description to two bodies as illustrated in Figure 1. We denote the two bodies by Ω^A and Ω^B and the union of the two bodies by Ω. The boundaries of the bodies are denoted by Γ^A and Γ^B. The corresponding entities in the initial state, which corresponds to the reference state, are indicated by subscript noughts, e.g. Ω_0^A.

In contact-impact formulations, it is often desirable to express the discretized equations in terms of one of the bodies, which is called the master; here, body A is designated as the master, body B as the slave. When we wish to distinguish field variables that are associated with a particular body, we append a superscript A or B; when neither of these superscripts appears, the field variable applies to the union of the two bodies. Thus the motion of the two bodies is given by

$$\mathbf{x}^A = \mathbf{x}(\mathbf{X}^A, t), \quad \mathbf{x}^B = \mathbf{x}(\mathbf{X}^B, t) \tag{1}$$

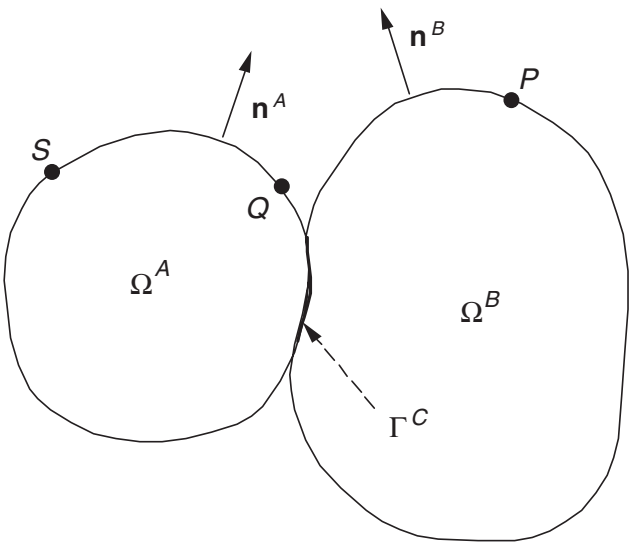

Figure 1. Nomenclature for contact-impact problem.

and the velocity field, $\mathbf{v}^B(\mathbf{X}^B,t) = \partial\mathbf{x}(\mathbf{X}^B,t)/\partial t$ refers to the velocity in body B, whereas $\mathbf{v}(\mathbf{X},t)$ refers to the velocity field in both bodies.

The contact interface Γ_c is the intersection of the surfaces of the two bodies, i.e.

$$\Gamma_c = \Gamma^A \cap \Gamma^B \tag{2}$$

The contact interface consists of the two surfaces of the two bodies that are in contact, but since they are theoretically coincident we refer to a single interface Γ_c, which is a function of time. In numerical solutions, the two surfaces will usually not be coincident. In those cases, Γ_c refers to the master surface. Moreover, when the two bodies are in contact on several disjoint interfaces, we designate their union by Γ_c.

The bodies are governed by the conservation of mass, momentum and energy and the constitutive equations; in addition a measure of strain, such as Green strain must be defined. Contact adds the following conditions: the bodies can not interpenetrate, called the impenetrability condition, and the tractions must satisfy momentum conservation on the interface. We assume here that the normal traction across the contact interface cannot be tensile, although it is not difficult to extend the methods to the situation where the normal tractions are bounded by an adhesive limit. The impenetrability condition is $\Omega^A \cap \Omega^B = 0$, i.e. the intersection of the two bodies is the null set. This condition can be written as

$$g(\mathbf{x}) = \min_{\bar{\mathbf{x}}\in\Gamma^A} \|\bar{\mathbf{x}} - \mathbf{x}\| \operatorname{sign}(\mathbf{n}^A \cdot (\bar{\mathbf{x}} - \mathbf{x})) \leqslant 0 \quad \forall \mathbf{x} \in \Gamma^B \tag{3}$$

where $\mathbf{n}^A$ is the unit outward normal to body A and $g(\mathbf{x})$ is the gap function (also called the interpenetration function); note that the sign of the above gap function differs from the common definition, but we use this definition since it avoids many minus signs in the subsequent development.

Penalty weak form. The formulation of contact-impact in this paper is limited to Lagrangian meshes. The implementation is independent of whether the formulation is total Lagrangian or updated Lagrangian or of the method for enforcing the impenetrability constraint. We have selected the penalty method as a model since it is the simplest and avoids inequality constraints. Moreover, because we are primarily interested in describing the use of this gap function, we will restrict the development to frictionless contact.

The weak form of the contact problem with the impenetrability condition imposed by penalty methods in terms of the displacements $\mathbf{u}(\mathbf{X},t)=\mathbf{x}-\mathbf{X}$ can be written as

For $\mathbf{u}(\mathbf{X},t)\in U$

$$0=\delta W(\mathbf{u})=\delta W^{\text{int}}(\mathbf{u})-\delta W^{\text{ext}}(\mathbf{u})+\int_{\Gamma_c}\beta g\,\frac{\partial g}{\partial \mathbf{u}}\,H(g(\mathbf{u}))\delta\mathbf{u}\,\mathrm{d}\Gamma,\quad \forall\delta\mathbf{u}(\mathbf{X})\in U_0 \tag{4}$$

In the above β is the penalty parameter and $H(g(\mathbf{u}))$ is the Heaviside step function defined by $H(x)=1$ for $x>0$; $H(x)=0$ if $x<0$. The scalars $W^{\text{int}}(\mathbf{u})$ and $W^{\text{ext}}(\mathbf{u})$ are the internal energy and the work of the external forces, respectively. For the penalty method the weak form is not an inequality; the signed character of the problem is taken care of by the presence of the Heaviside step function in the integral over the contact surface in the above. Note that the integral over the contact surface Γ_c consists of two surface integrals over Γ_c^A and Γ_c^B. The space U of trial functions consists of all functions which are continuous and piecewise continuously differentiable within each body and satisfy displacement boundary conditions; the space of trial functions need not be continuous across the interface between the two bodies. The space U_0 of test functions is the same as U except that the test functions vanish on the boundaries where the displacements are prescribed.

For the purpose of developing the discrete equations, we use a Lagrangian mesh and make the following approximation of the motion in terms of shape functions $N_I(\mathbf{X})$, and nodal values of the displacements $u_{iI}(t)$:

$$u_i(\mathbf{X},t)=N_I(\mathbf{X})u_{iI}(t) \tag{5}$$

where $u_{iI}(t)$ are the components of the displacements at node I. Throughout this paper, repeated upper case indices are summed over the nodes of the model or of an element; the range of lower case indices is 1 to n_{SD}, where n_{SD} is the number of space dimensions. The nodal displacements are sometimes also arranged in a column matrix $\mathbf{d}$, where $d_a=u_{iI},\ a=n_{\text{SD}}(I-1)+i$. The discrete weak form of the momentum equation can be written as

$$0=\delta W(\mathbf{d})=\frac{\partial W^{\text{int}}(\mathbf{d})}{\partial u_{iI}}\,\delta u_{iI}-\frac{\partial W^{\text{ext}}(\mathbf{d})}{\partial u_{iI}}\,\delta u_{iI}+\int_{\Gamma_c}\beta g\,\frac{\partial g}{\partial u_{iI}}\,H(g(\mathbf{d}))\,\mathrm{d}\Gamma\,\delta u_{iI} \tag{6}$$

We then invoke the standard definitions of nodal internal, external and contact forces:

$$f_{iI}^{\text{int}}=\frac{\partial W^{\text{int}}(\mathbf{d})}{\partial u_{iI}} \tag{7}$$

$$f_{iI}^{\text{ext}} = \frac{\partial W^{\text{ext}}(\mathbf{d})}{\partial u_{iI}} \tag{8}$$

$$f_{iI}^{\text{c}} = \int_{\Gamma^{\text{c}}} \beta g \frac{\partial g}{\partial u_{iI}} H(g(\mathbf{d}))\, \mathrm{d}\Gamma \tag{9}$$

The internal and external nodal forces can be computed either by a total Lagrangian formulation or an updated Lagrangian formulation; the corresponding equations are simply different expressions for the same nodal forces and the numerical values are identical. One can also use standard forms that do not arise from potentials, see Reference [13].

The equations corresponding to (6) can be obtained by invoking the arbitrariness of the nodal displacements on all surfaces that are not prescribed displacement surfaces, and some subtle steps on the contact surface which we will not repeat here, see Reference [13],

$$f_{iI}^{\text{int}} - f_{iI}^{\text{ext}} + f_{iI}^{\text{c}} = 0 \tag{10}$$

The above are the discrete equations for the contact impact problem in the penalty method. Note that the impenetrability constraint is completely taken care of by the contact forces arising from the penalty.

We can add the effects of inertia by adding a d'Alembert inertial term $f_{iI}^{\text{kin}} = M_{IJ}\ddot{u}_{iJ}$, where M_{IJ} is the mass matrix and superposed dots denote time derivatives. The discrete equations are then

$$M_{IJ}\ddot{u}_{iJ} + f_{iI}^{\text{int}} - f_{iI}^{\text{ext}} + f_{iI}^{\text{c}} = 0 \tag{11}$$

Remark

As can be seen from Equation (9), the contact forces do not depend on the shape functions but instead on the approximation of the surface. This is to be expected since the shape of the surface depends on the gap function approximation rather than the shape functions.

3. GAP ALGORITHM

The direct computation of a smoothed gap function (called the interpenetration function in Reference [13]) is described in this section. The essential idea is to associate with each body a smooth signed distance function expressed in terms of the surface nodes positions of the body. When a standard master–slave approach is used, it is only necessary to have a smoothed distance function for the master body. The interpenetration of any node on the slave body is then computed in terms of the gap function for the master body. However, we will first describe a general algorithm, where both bodies are treated equivalently.

Each body is associated with a signed distance function, which for bodies A and B are denoted by $\phi^A(\mathbf{x}, t)$ and $\phi^B(\mathbf{x}, t)$, respectively, as shown in Figure 2. The signed distance function for body A is given by

$$\phi^A = \text{sign}(\mathbf{n}^A \cdot (\bar{\mathbf{x}} - \mathbf{x})) \min_{\bar{\mathbf{x}} \in \Gamma^A} \|\bar{\mathbf{x}} - \mathbf{x}\| \tag{12}$$

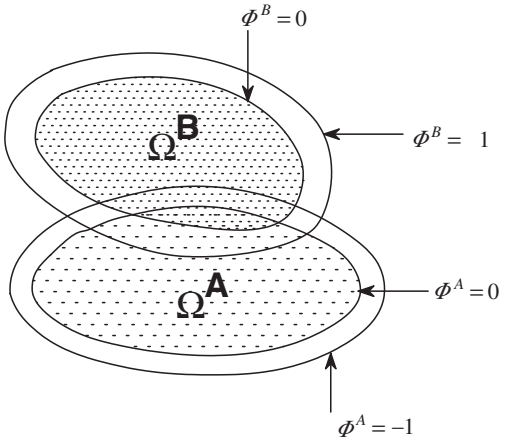

Figure 2. Two overlapping bodies and their signed distance functions.

The constraint that the two bodies do not interpenetrate is then given by

$$\phi^A(\mathbf{x},t) \leqslant 0 \quad \forall \mathbf{x} \in \Gamma^B \tag{13}$$

$$\phi^B(\mathbf{x},t) \leqslant 0 \quad \forall \mathbf{x} \in \Gamma^A \tag{14}$$

The impenetrability condition can also be expressed in terms of the gap function. For this purpose, we propose the following algorithmic definition of the gap function in terms of the signed distance functions:

$$g(\mathbf{x},t) = \max(\phi^A(\mathbf{x},t), \phi^B(\mathbf{x},t)) \leqslant 0 \tag{15}$$

where the inequality gives the impenetrability condition. Note that the above defines the gap function for any point in space: it is only necessary to compute the signed distance functions and then to choose the maximum of the two. This definition conforms with the standard definition of the gap function (3) but is advantageous when the gap function is constructed by an implicit smoothing as is done in the following.

For clarity, we describe the algorithm here for a master–slave approach to the contact. For a master–slave algorithm, the signed distance function is only constructed for the master body, which we denote by A. The rough finite element model is retained for body B. The gap function in this case is equivalent to $\phi^A(\mathbf{x},t)$, so the impenetrability condition can be written as

$$g(\mathbf{x},t) \equiv \phi^A(\mathbf{x},t) \leqslant 0 \quad \forall \mathbf{x} \in \Gamma^B \tag{16}$$

The gap function is constructed by a moving least square procedure in this paper, although other implicit function constructions can be employed. We describe the method only for the determination of $\phi^A(\mathbf{x},t)$; the determination of $\phi^B(\mathbf{x},t)$ is identical. The basic idea is to fit a smooth implicit surface function through the surface nodes of body A in the vicinity of any

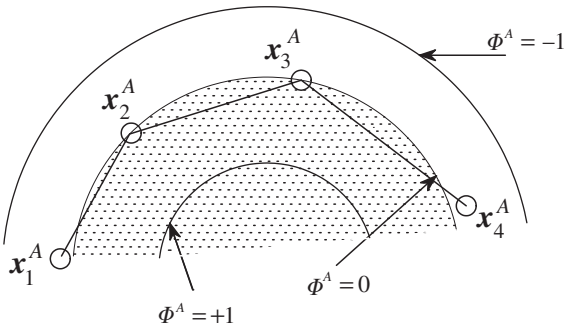

Figure 3. Example of a fit of a signed distance function to a set of nodes connected by linear element edges.

point where the gap is needed. The smooth surface is given by the zero isosurface of $\phi^A(\mathbf{x},t)$, i.e. $\phi^A(\mathbf{x},t)=0$. The implicit function is constructed by a moving least square function, see Reference [16]. As the function is constructed, we try to keep the norm of the gradient of $\phi^A(\mathbf{x},t)$ close to unity, so the function is approximately a signed distance function. A generic set of nodes for a quadrilateral mesh and the signed distance function $\phi^A(\mathbf{x},t)$ are illustrated in Figure 3. As can be seen, the smooth surface does not pass exactly through the nodes but it provides a least-squares fit.

The implicit distance function is

$$\phi^A(\mathbf{x},t)=a_i(\mathbf{x},t)\,p_i(\mathbf{x}) \tag{17}$$

where $a_j(\mathbf{x},t)$, for $j=1$ to m are the spatially varying parameters of the approximation to the surface and $p_j(\mathbf{x})$, for $j=1$ to m are the basis for the approximation; repeated lower case indices are summed over their range throughout this paper. In most cases, we will use a polynomial basis. For example, for a linear polynomial basis in two dimensions

$$[a_j]=[a_1\ \ a_2\ \ a_3] \tag{18}$$

$$[p_j]=[1\ \ x\ \ y] \tag{19}$$

For a quadratic polynomial basis in two dimensions

$$[a_j]=[a_1\ \ a_2\ \ a_3\ \ a_4\ \ a_5\ \ a_6] \tag{20}$$

$$[p_j]=[1\ \ x\ \ y\ \ x^2\ \ xy\ \ y^2] \tag{21}$$

Note that the function $\phi^A(\mathbf{x},t)$ is defined over the entire space, not just on the surface.

The coefficients $a_j(\mathbf{x},t)$ are obtained at any point in the domain of the problem by minimizing the weighted quadratic form

$$J(a_i,\mathbf{x},t)=\frac{1}{2}\sum_{I\in S_{\mathbf{x}}} w(\mathbf{x}-\mathbf{x}_I(t))(a_i(\mathbf{x},t)\,p_i(\mathbf{x}_I(t)))^2+\frac{\mu}{2}(\|\nabla\phi^A(\mathbf{x},t)\|^2-1) \tag{22}$$

where $w_I(\mathbf{x}-\mathbf{x}_I(t))$ is a weight function of compact support that is at least once continuously differentiable; μ is a Lagrange multiplier that enforces the normality of the gradient of $\phi^A(\mathbf{x},t)$. The purpose of the constraint associated with the Lagrange multiplier is to approximately endow $\phi^A(\mathbf{x},t)$ with the properties of a signed distance function. The support of the weight function w_I is denoted by Ω_I and $S_{\mathbf{x}}$ is the set of all nodes for which Ω_I includes $\mathbf{x}$. In some of our numerical implementations, we do not use the square of the norm in the last term of (22).

Note that the normality constraint is only applied at the point where the moving least square function is evaluated, yet remarkably, we find that the interpenetration functions constructed by this procedure are approximately signed distance functions in the vicinity of the surface of body A; this will be illustrated in the examples in Section 5. We have used cubic spline weight function in our studies

$$\bar{w}(r)=\begin{cases}\frac{2}{3}-4r^2+4r^3 & \text{for } 0\leqslant r\leqslant\frac{1}{2}\\ \frac{4}{3}-4r+4r^2-\frac{4}{3}r^3 & \text{for } \frac{1}{2}<r\leqslant 1\\ 0 & \text{for } 1<r\end{cases} \tag{23}$$

$$r=\frac{1}{d_{\inf}}\|\mathbf{x}-\mathbf{x}_I\| \tag{24}$$

We make the size of the domain of influence, $d_{\inf}$ a function of $\mathbf{x}$, so that it can be set on the run to include the minimum number of nodes sufficient to determine the minimum of $J(a_i,\mathbf{x},t)$. For a linear basis in two dimensions, at least three nodes should be included in the domain of influence, for a quadratic basis, six nodes are needed.

4. IMPLEMENTATION

To make the notation more compact, we let $w_I=w(\mathbf{x}-\mathbf{x}_I)$, $p_{iI}=p_i(\mathbf{x}_I)$. The gradient of the signed distance function is then given by

$$\phi_{,k}=a_{i,k}p_i+a_ip_{i,k} \tag{25}$$

The equations for coefficients $a_j(\mathbf{x},t)$, for $j=1$ to m, are obtained by finding the stationary points of J in Equation (22). This yields

$$0=\frac{\partial J}{\partial a_k}=\sum_{I\in S_{\mathbf{x}}} w_Ia_ip_{iI}p_{kI}+\mu(a_{i,r}p_i+a_ip_{i,r})p_{k,r} \tag{26}$$

$$0=\frac{\partial J}{\partial a_{j,k}}=\sum_{I\in S_{\mathbf{x}}}\mu(a_{i,k}p_i+a_ip_{i,k})p_j \tag{27}$$

$$0=\frac{\partial J}{\partial \mu}=\sum_{I\in S_{\mathbf{x}}}(a_{i,k}p_i+a_ip_{i,k})(a_{j,k}p_j+a_jp_{j,k}) \tag{28}$$

The above are non-linear equations in the unknowns $a_j(\mathbf{x},t)$, $a_{j,k}(\mathbf{x},t)$ and the Lagrange multiplier $\mu(\mathbf{x},t)$. For a polynomial basis, these equations can be simplified as shown below; as a matter of fact, closed form solutions are possible in two dimensions with certain assumptions.

We next illustrate the above equations for a two dimensional problem with a linear basis. We assume that the spatial derivatives of the parameters $a_{j,k}(\mathbf{x},t)$ can be neglected, which, as we shall see, does not detract much from the accuracy of the method. We immediately consider a master-slave form of the algorithm, so instead of the two signed distance functions, we need only compute one gap function (the more general procedure would just entail two repetitions of this step). The gap function for a linear basis in two dimensions is given by

$$g = a_1(\mathbf{x},t) + a_2(\mathbf{x},t)x + a_3(\mathbf{x},t)y \tag{29}$$

so

$$\mathbf{a} = [a_i] = [a_1 \;\; a_2 \;\; a_3] \text{ and } \mathbf{p} = [p_i] = [p_1 \;\; p_2 \;\; p_3] \tag{30}$$

If we neglect $a_{i,j}$, the gradient of the gap is given by

$$\nabla g = [g_{,i}] = [a_2 \;\; a_3] \tag{31}$$

The function J can be written as

$$J = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{A}\mathbf{a} + \frac{\mu}{2}(\mathbf{a}^{\mathrm{T}}\mathbf{B}\mathbf{a} - 1) \tag{32}$$

where

$$\mathbf{A} = \sum_I w_I \begin{bmatrix} 1 & x_I & y_I \\ & x_I^2 & x_I y_I \\ & & y_I^2 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ & 1 & 0 \\ & & 1 \end{bmatrix} \tag{33}$$

The equations for $\mathbf{a}$ are then

$$[\mathbf{A} + \mu\mathbf{B}]\mathbf{a} = \mathbf{0} \tag{34}$$

$$a_2^2 + a_3^2 = 1 \tag{35}$$

The above is a system of non-linear algebraic equations. In two dimensions the above system is easily solved since constraint (35) can automatically be met by letting

$$a_2 = \cos\alpha, \quad a_3 = \sin\alpha \tag{36}$$

The function J can then be given in terms of α:

$$J = \tfrac{1}{2}(c_1 - c_2 \sin^2\alpha + 2c_3 \sin\alpha\cos\alpha) \tag{37}$$

where

$$c_1 = A_{22} - \frac{A_{12}^2}{A_{11}} \tag{38}$$

$$c_2 = A_{22} - A_{33} - \frac{A_{12}^2}{A_{11}} + \frac{A_{13}^2}{A_{11}} \tag{39}$$

$$c_3 = A_{23} - \frac{A_{12}A_{13}}{A_{11}} \tag{40}$$

Minimizing J with respect to α by setting the first derivative to zero then gives

$$\alpha = \frac{1}{2} \tan^{-1} \frac{2c_3}{c_2} \tag{41}$$

The above has two solutions, so an estimate of the normal is necessary to select the correct solution. The correct angle is then the one that meets the condition $\mathbf{n}^A \cdot \nabla g \leqslant 0$; an estimate of the normal can easily be obtained by the assembled normal algorithm, [17], or by computing the normal to the nearest surface facet. This condition conforms to our convention that the gap function is negative inside the body and the positive normal points out. For three-dimensional problems, the determination of $\mathbf{a}$ can be posed as a minimization problem in two unknowns, but we have used a more general solver described later.

4.1. Computation of derivatives of gap function

As can be seen from Equation (9), the derivative of the gap function with respect to the nodal coordinates is needed to compute the contact forces in the penalty method. It is also needed in the Lagrange multiplier and augmented Lagrange methods.

We describe next how these derivatives are computed. For this purpose, we rewrite the normality condition as

$$\mathbf{a}^{\mathrm{T}}\mathbf{B}\mathbf{a} = 1 \tag{42}$$

We then take derivatives of Equations (32) and (33) with respect to x_{iI}. This yields

$$\mathbf{A}_{,x_{iI}}\mathbf{a} + \mathbf{A}\mathbf{a}_{,x_{iI}} + \mu_{,x_{iI}}\mathbf{B}\mathbf{a} + \mu\mathbf{B}_{,x_{iI}}a + \mu\mathbf{B}\mathbf{a}_{,x_{iI}} = 0 \tag{43}$$

$$\mathbf{a}^{\mathrm{T}}\mathbf{B}\mathbf{a}_{,x_{iI}} = 0 \tag{44}$$

where we have used the symmetry of $\mathbf{B}$ in developing the second equation. The above can be combined in a single matrix equation

$$\begin{bmatrix} \mathbf{A} + \mu\mathbf{B} & \mathbf{B}\mathbf{a} \\ \mathbf{a}^{\mathrm{T}}\mathbf{B} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{a}_{,x_{iI}} \\ \mu_{,x_{iI}} \end{Bmatrix} = \begin{Bmatrix} -\mathbf{A}_{,x_{iI}}\mathbf{a} - \mu\mathbf{B}_{,x_{iI}}\mathbf{a} \\ \mathbf{0} \end{Bmatrix} \tag{45}$$

Second derivatives, when needed for linearization of the equations, can be calculated similarly. It is only necessary to take derivatives of Equations (43) and (44) and put all terms which are known on the right-hand side. Note that part of the right-hand side of (45) is already triangulated when (34) is solved.

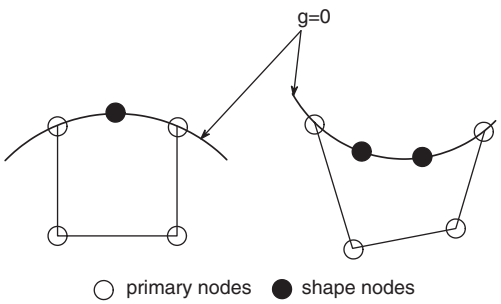

Figure 4. Applications of shape nodes to obtain a better description of a curved surface.

4.2. Numerical implementation

For quadratic or higher order basis in both two and three dimensions, a general solver is necessary. In our implementation, the supports of the weight functions are chosen so that the minimum number of nodes required to determine the moving least square approximation are used at each point that the gap is computed. For example, for a linear basis, since there are only three constants, it suffices that the supports of three nodes on the contact surface include the point considered; we denote the number of nodes selected to fit the function at $\mathbf{x}$ by $m(\mathbf{x})$. The algorithm is as follows:

1. The $m(\mathbf{x})$ closest nodes on surface A to $\mathbf{x}$ are found; this set of nodes is called $S(\mathbf{x})$ and $d_{\inf}(\mathbf{x})$ in (24) is set by

$$d_{\inf}(\mathbf{x}) = \max_{I \in S} 2\|\mathbf{x} - \mathbf{x}_I\| \tag{46}$$

2. Only the nodes in the set $S(\mathbf{x})$ are used to compute the gap function at x.

The weight function is then a function of the spatial coordinate, i.e. $w = w(\mathbf{x}, \mathbf{x} - \mathbf{x}_I)$. In computing the derivatives of w with respect to spatial coordinates, we neglect its dependence on the first of these variables.

For bases that are higher order than quadratic, the solver must be able to deal with the fact that whenever the nodes that define the contacting surface are coplanar or nearly coplanar, the matrix $\mathbf{A}$ becomes singular or near singular. This invariably occurs as meshes are refined. In Appendix A, we describe a method which is able to deal with these difficulties.

4.3. Shape nodes

In some cases, the actual nodes of the finite element mesh do no suffice to give a good description of the shape of the contact surfaces. These details can be modeled by defining additional shape nodes as shown in Figure 4; these can be called secondary or slave nodes, but we do not use the latter nomenclature because it is already used for master and slave contact surface nodes. Shape nodes are also useful in the algorithm because of the number of nodes needed to define the surface is often greater than the number of nodes in the contacted element. For example, with a linear basis in two dimensions, three nodes are needed to define the moving least square approximation.

Therefore, some of the nodes that are assigned contact forces according to (9) are outside the contact area. Similar situations arise for linear bases in three dimensions and quadratic bases in two or three dimensions, though the disparity between the number of nodes on a surface element and that required to define the smoothed function decreases in three dimensions. The shape nodes do not add unknowns to the system and serve strictly to describe the shape and deformation of the contact surface.

We assume that the motion of the shape nodes on the surface are given by the same interpolant that governs the motion within the element, so

$$\mathbf{x}^{\mathrm{S}}(\xi, t) = \mathbf{x}_I(t) N_I(\xi^{\mathrm{S}}) \tag{47}$$

where the range of the repeated upper case index is over the nodes of the element whose edge contains the slave node $\mathbf{x}^{\mathrm{S}}$ and ξ^{S} is the parent element co-ordinate of this shape node. Note that the parent element co-ordinate ξ^{S} will be outside the standard domain if the surface is convex with respect to the interior of the element, inside the parent element if the surface is concave.

The additional nodal contact forces dues to the presence of the shape nodes are then obtained by evaluating Equation (9) by the chain rule and using Equation (47), which gives

$$f_{iI}^{\mathrm{c}} = \int_{\Gamma_{\mathrm{c}}} \beta g \frac{\partial g}{\partial u_{iJ}^{\mathrm{S}}} N_J(\xi^{\mathrm{S}}) H(g(\mathbf{d}))\, \mathrm{d}\Gamma \tag{48}$$

where u_{iJ}^{S} are the displacements of the displacements of the shape nodes of the element. Note that the contact forces will now appear on all nodes of the element, including the nodes one layer beneath the contact surface. However, the nodal forces on the layer of nodes beneath the contact surface will be quite small.

4.4. Features

In most finite element models, small scale features such as fillets and rounds on corners are neglected because modeling them would entail the use of too many elements. Instead, finite element models often have sharp corners at intersections of surfaces. However, sharp corners are often inappropriate models for the contact phenomena that occur there. In this algorithm, it is very easy to include features such as rounded corners and fillets. We illustrate the concept in Figure 5. The basic idea is to associate a feature in the model with a specific gap function and to define an appropriate hierarchy of gap functions. For a rounded corner, the gap functions is

$$g = r_0 - \sqrt{(x - x^f)^2 + (y - y^f)^2} \equiv r_0 - \|\mathbf{x} - \mathbf{x}^f\| \tag{49}$$

where $\mathbf{x}^f = [x^f \quad y^f]$ is the position of the centre of the round and r_0 is the radius of the round. This gap function is then used to compute the gap in the domain of dominance for the round: see Figure 5.

Since the gap in the above is not expressed in terms of the nodes of the system, it is necessary to transfer nodal contact forces to the primary nodes, as for the shape nodes. The relationship is obtained by using the chain rule with Equations (49) and (47):

$$f_{iI}^{\mathrm{c}} = \int_{\Gamma_{\mathrm{c}}} \beta g N_I(\xi^f) \frac{x_i - x_i^f}{\|x_i - x_i^f\|}\, \mathrm{d}\Gamma \tag{50}$$

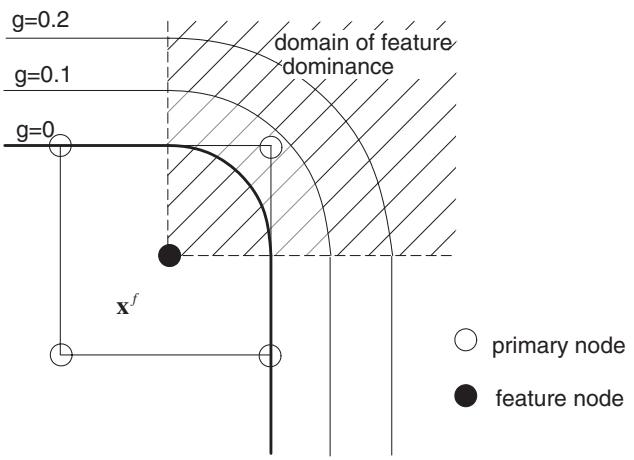

Figure 5. Depiction of a feature: a rounded corner; the domain of feature dominance is also shown.

The philosophy here differs markedly from Pandolfi *et al.* [18], who deal explicitly with corners. Here, the objective is to avoid corners.

A crease can be introduced in the contact surface by adding the absolute signed distance function to the gap function, see Reference [19] or [20] for the construction of functions with discontinuous derivatives.

4.5. Other matters

For a master–slave routine, an estimate of the normal is always needed to obtain the correct sign on the solutions for **a**. One approach is to use an assembled surface normal algorithm, see Reference [17] or [21]. Alternatively, the normal can be estimated by finding the normal to the closest facet of the master.

The contact-impact algorithm, like any other contact-impact routine, requires a bin sort to isolate sets of nodes that are likely to be in contact in order to achieve reasonable efficiency. These algorithms are well known and widely used in contact-impact methods so we will not discuss them further.

5. TESTS AND EXAMPLES

In the following we describe some tests of the algorithm. Our intent is not to demonstrate the solution of large problems, but to list some tests that should be made by anyone who wishes to implement this algorithm and to indicate the level of accuracy. We also use these examples to illustrate some of the advantages and shortcomings of this method. In each of the tests, the nodes were placed on a surface defined by an analytic function, such as a sphere. The gap function and its gradient were then computed in a subdomain about the surface. Three measures are used to quantify the error of the

algorithm:

1. *Error in the surface definition.* This corresponds to the deviation of the $g=0$ isosurface from the analytic surface on which the nodes were placed, given by

$$\mathrm{err}_\Gamma = \frac{1}{S_\Gamma}\int_\Gamma |g(\mathbf{x})|\,\mathrm{d}\Gamma \tag{51}$$

where S_Γ is the area of the surface considered. Note that the integral is evaluated over the exact surface. For a circle in two dimensions, this definition of error can be approximated by

$$\mathrm{err}_\Gamma = \frac{1}{2\pi r_{\mathrm{exact}}}\int_0^{2\pi} |r - r_{\mathrm{exact}}| r_{\mathrm{exact}}\,\mathrm{d}\theta \tag{52}$$

where r is the distance from a point on the computed zero isoline to the origin of the circle.

2. *Error in penetration.* This is a measure of the error in the penetration function $g(\mathbf{x})$ in a subdomain about the impact surface. It is given by

$$\mathrm{err}_g = \frac{I(g-g_{\mathrm{exact}})}{I(g)}, \quad \text{where } I(g) = \int_\Omega |g|\,\mathrm{d}\Omega \tag{53}$$

where Ω is a volume surrounding the contact surface.

3. *Error in gradient of gap function.* This is a measure in the error in the gradient of in the vicinity of the impact surface. It is given by

$$\mathrm{err}_{\nabla g} = \frac{I_2(\nabla g - \nabla g_{\mathrm{exact}})}{I_2(\nabla g)}, \quad \text{where } I_2(g) = \left(\int_\Omega \|\nabla g\|\,\mathrm{d}\Omega\right)^{1/2} \tag{54}$$

5.1. Straight line and plane

The first test of this algorithm is set of points on a straight line in two dimensions and a set of coplanar set of points in three dimensions. The zero isosurface for a linear basis should fit the points within machine precision, 10^{-11} to 10^{-12} on a MATLAB program on a PC. This degree of accuracy was obtained for both uniformly spaced points and nonuniformly spaced nodes.

5.2. Circle

Twenty to eighty nodes were placed on a circle of radius $r_{\mathrm{exact}} = R = 5$. The gap function was then computed with linear and quadratic bases. Table I gives the three error measures described above: the subdomain Ω over which the error was measured consists of the domain bounded by circles of radii $0.9R$ and $1.1R$. An example of the smoothed contact surface is given in Figure 6.

For a linear basis, the errors in the gap and radius converge with increasing number of nodes. The zero isosurface for a linear basis always inscribes the nodes, i.e. the radius is too small. The error in the radius of the approximate surface decreases rapidly as the number of nodes increases. For a quadratic basis, the zero isosurface fits the circle exactly: it has the correct radius with machine precision. Similarly, the gradient is obtained in the subdomain around the surface with machine precision. The penetration has a slight error which does not decrease with an increasing number of nodes: evidently the metric is slightly distorted by this

Table I. Error in the gap function and its gradient for a circle.

No. of nodes	Nodes/support	err_g	err_Γ	$\mathrm{err}_{\nabla g}$
Linear basis				
20	3	0.020119	0.028499	0.00051878
40	3	0.00502	0.0064339	0.00042572
60	3	0.0019735	0.0031247	0.026183
80	3	0.0012439	0.0016031	8.332E-5
Quadratic basis				
20	6	0.0010309	1.8793E-17	1.1945E-16
40	6	0.0010309	1.8947E-17	1.646E-16
60	6	0.0010309	2.0104E-17	6.0728E-17

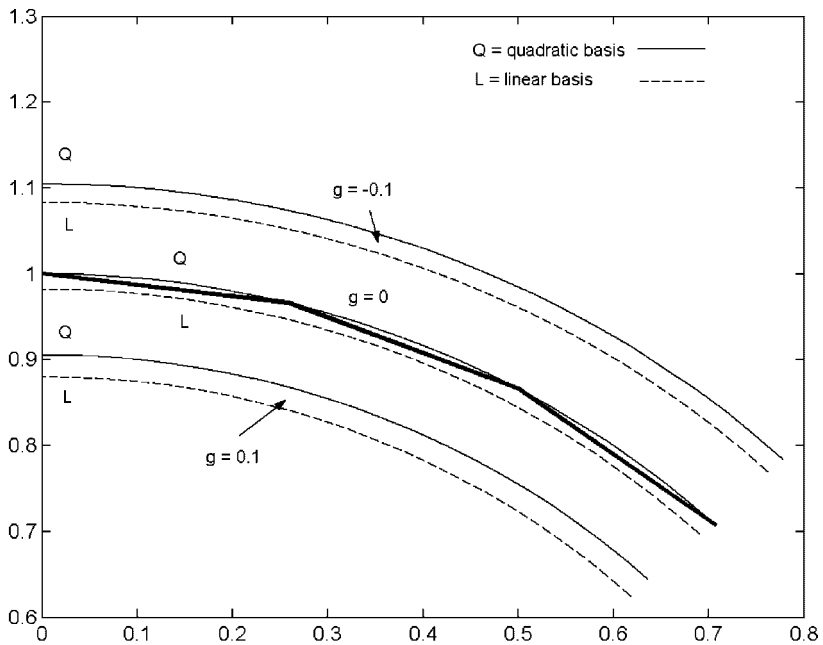

Figure 6. Piecewise linear finite element surface through nodes of a circle over a 45° sector and the zero isosurface from moving least square approximations with linear and quadratic bases.

method, regardless of the number of nodes. This should not affect the accuracy of solutions, since it only effects the magnitude of the penalty term in the contact algorithm, not where it vanishes. Nevertheless, it is surprising how well the constructed function corresponds to a signed distance function even though the normality constraint is only applied at the point the gap is computed.

Table II gives results for a set of nodes on the circle that are randomly spaced. The errors are only slightly largely than for the uniformly spaced nodes, and for most purposes the differences are irrelevant.

Table II. Error in the gap function and its gradient for a circle for randomly spaced nodes.

No. of nodes	Nodes/support	err_g	err_Γ	$\mathrm{err}_{\nabla g}$
Linear basis				
20	3	0.017953	0.029015	0.083999
40	3	0.0045053	0.0072771	0.041485
60	3	0.0019181	0.0032469	0.027744
80	3	0.0011293	0.0018312	0.020585
Quadratic basis				
20	6	0.0010309	2.1565E-17	4.4805E-17
40	6	0.0010309	2.9445E-17	3.0652E-16
60	6	0.0010309	1.1333E-16	4.1634E-14

Table III. Error in the gap function and its gradient for a cosine function.

No. of nodes	Nodes/support	err_g	err_Γ	$\mathrm{err}_{\nabla g}$
Linear basis				
10	3	0.078935	0.0043936	0.18056
20	3	0.017854	0.001042	0.0080975
40	3	0.0041367	0.00023672	0.0043724
80	3	0.0041367	0.00023627	0.0043724
Quadratic basis				
10	6	0.011021	0.00027488	0.026673
20	6	0.011122	0.00010637	0.0040426
40	6	0.011172	5.8487E-6	0.0010493
80	6	0.011139	2.9109E-6	0.00027357

5.3. *Cosine function*

The objective of this problem is to test the algorithm for a situation where the curvature varies markedly. The surface is given by

$$y = \cos x, \quad 0 \leqslant x \leqslant \frac{\pi}{2}$$

For this curve, the curvature varies from 0 to 1. The results are given in Table III.

The moving least square approximation and the straight line segments connecting the nodes for a mesh of linear or bilinear finite elements are shown in Figure 7. It can be seen that the moving least-squares zero isosurface is a little inside the nodes. This figure is for a coarse mesh. For a finer mesh, the isosurface passes almost exactly through the nodes.

5.4. *Cross-section of cylinder with hemispherical caps*

The nodes are placed on a cross-section of a cylindrical vessel with spherical caps. This problem is included because the transition from a curved surface to a linear surface causes some difficulties if the solver is not robust. Table IV gives the errors in the approximation

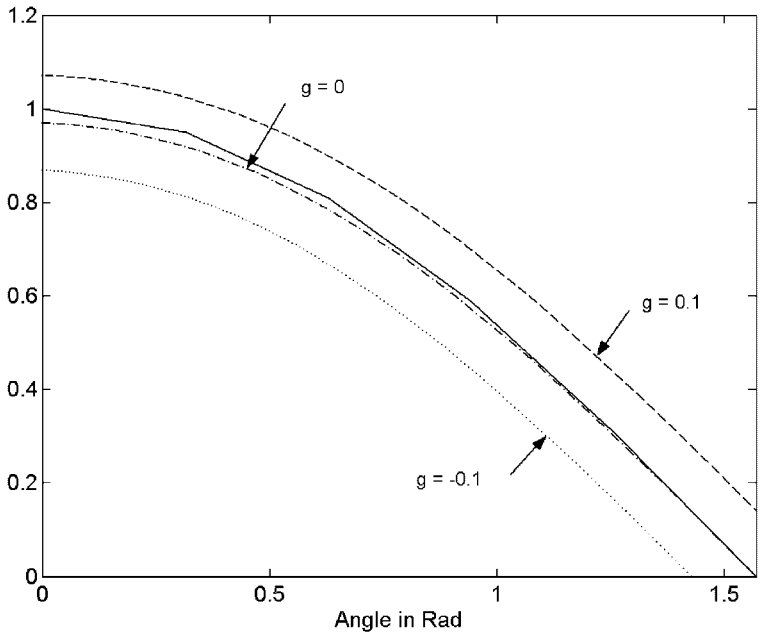

Figure 7. Piecewise linear finite element surface through nodes of a cosine function and the zero isosurface from moving least square approximations with linear and quadratic bases.

Table IV. Error in the gap function and its gradient for the cross-section of a cylinder with hemispherical ends.

No. of nodes	Nodes/support	err_g	err_Γ	$\mathrm{err}_{\nabla g}$
Linear basis				
66	4	0.050885	0.049046	0.13345
202	4	0.021137	0.017865	0.053211
Quadratic basis				
66	10	0.001649	1.491lE-33	1.8118E-16
202	10	0.0016497	7.1012E-07	1.3874E-06

to the surface. The radius of the spherical caps is unity, while the length of the cylindrical segment is 2.

5.5. *Sphere*

A coarse and fine mesh on the surface of a sphere of radius one are shown in Figure 8; only an octant is shown. The errors in g and its gradient were computed in a hollow sphere with inner and outer radii of 0.9 and 1.1. Note that both triangular and quadrilateral surface elements occur in the surface mesh, but as can be seen from Table V, the accuracy is still

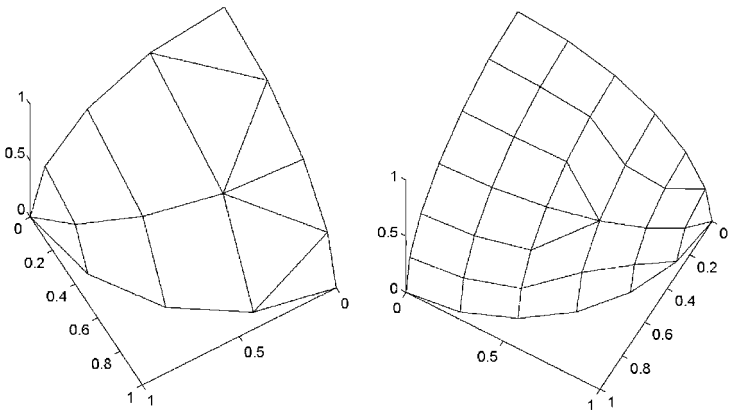

Figure 8. Coarse and fine meshes for an octant of the sphere; all nodes are on the surface of the sphere.

Table V. Error in the gap function and its gradient for coarse and fine models of sphere.

No. of nodes	Nodes/support	err_g	err_Γ	$\mathrm{err}_{\nabla g}$
Linear basis				
66	4	0.050885	0.049046	0.13345
202	4	0.021137	0.017865	0.053211
Quadratic basis				
66	4	0.001649	1.4911E-33	1.8118E-16
66	10	0.001649	1.4911E-33	1.8118E-16
202	4	0.0016497	7.0209E-07	1.0598E-06
202	10	0.0016497	7.1012E-07	1.3874E-06

very good. The zero isosurface and the gradient of the sphere is obtained with machine precision for quadratic basis functions. The error in the linear basis is significantly larger but still acceptable.

5.6. Ogival surface

To evaluate the performance of the method for a surface with a sharp point, we considered a surface of revolution with an ogival cross-section. The surface is given by

$$x^2 + y^2 - \sqrt{z} = 0 \tag{55}$$

The error is computed on the plane $z = 0.5$ using definitions of error analogous to those given above but limited to this plane, and given in Table VI.

5.7. Spinning disc

This problem is illustrated in Figure 9. A disc is spinning on a frictionless surface with an angular velocity of 5000 rad/s. The material properties are that of steel: Young's modulus

Table VI. Error in gap function and its gradient for an ogival three-dimensional surface.

No. of nodes	Nodes/support	err_g	err_Γ	$\mathrm{err}_{\nabla g}$
Linear basis				
242	4	0.021064	0.0087873	0.058576
439	4	0.017045	0.0044053	0.05494
687	4	0.0093902	0.00237	0.038656
1334	4	0.0016786	0.00087825	0.03001
Quadratic basis				
242	10	0.015944	0.00061205	0.018643
439	10	0.0094599	0.00021484	0.017543
687	10	0.0085432	0.00013093	0.017582
1334	10	0.0038863	0.00014861	0.013094

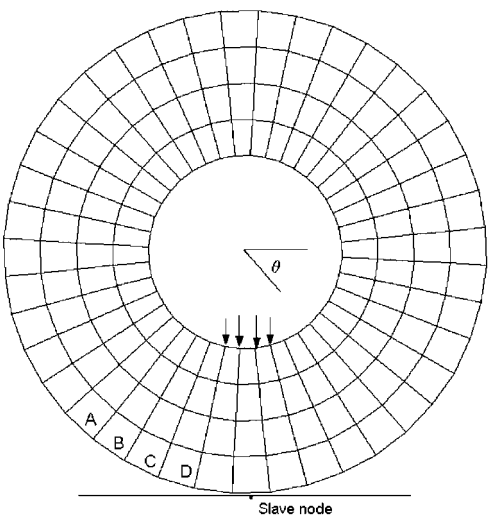

Figure 9. Finite element mesh for spinning wheel problem.

$E = 211$ GPa and Poisson's ratio $= 0.3$. The penalty parameter is $4E$. The disc is preloaded as shown with a load of 400 N. The central difference method was used for time integration. The forces at several of the surface nodes are shown in Figure 10. It can be seen that they are very smooth; and they increase almost linearly to their peaks and then decrease linearly. The forces at the sequential nodes repeat each other as expected.

5.8. *Concentric rings*

The inner of two concentric rings was rotated with an angular velocity of 4000 rad/s. The interface between the two rings is a single surface in the initial configuration, i.e. the outer radius of the inner ring equals the inner radius of the outer ring. The outer ring is fixed at its

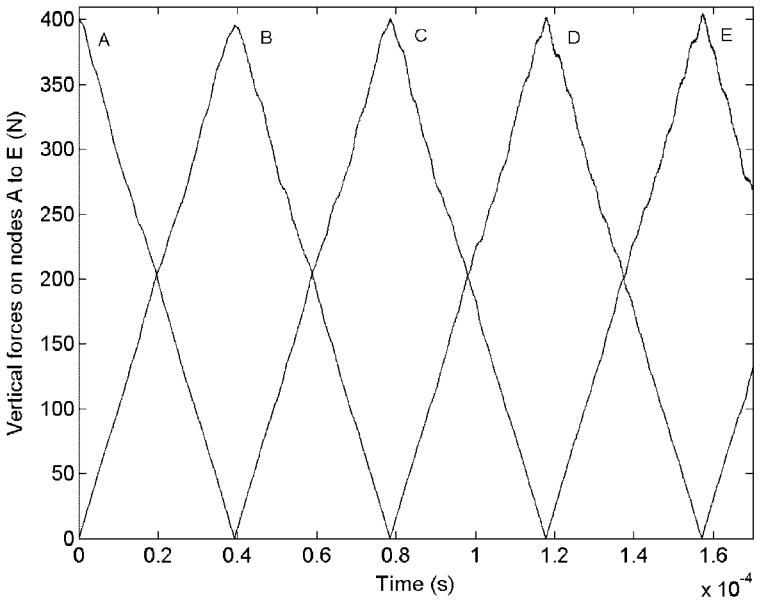

Figure 10. Vertical forces at nodes that contact the surface of the spinning wheel.

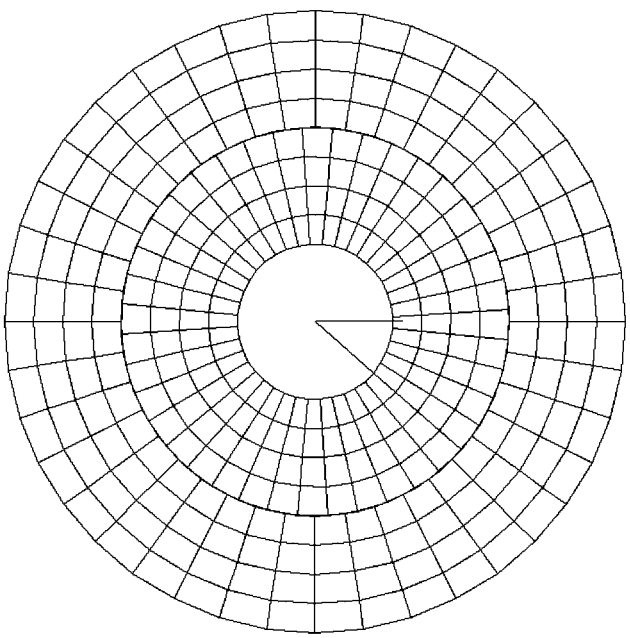

Figure 11. Finite element mesh for the concentric ring problem at 40 μs; in the initial configuration, the nodes of the inner and outer rings that are on the contact surface are coincident.

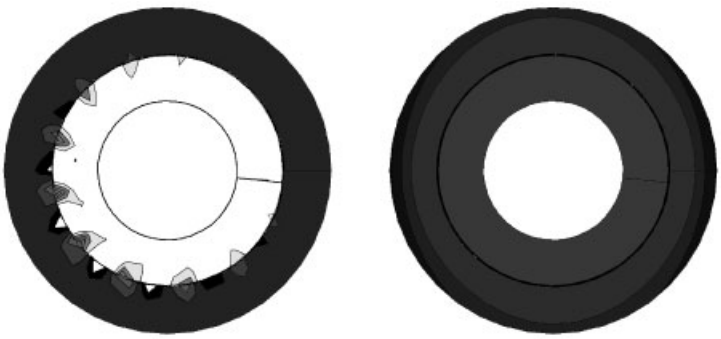

Figure 12. Contours of pressure in concentric rings at 40 μs for solutions with smoothed contact (right) and standard master-slave contact (left) on linear surfaces; one contour corresponds to 500 GPa.

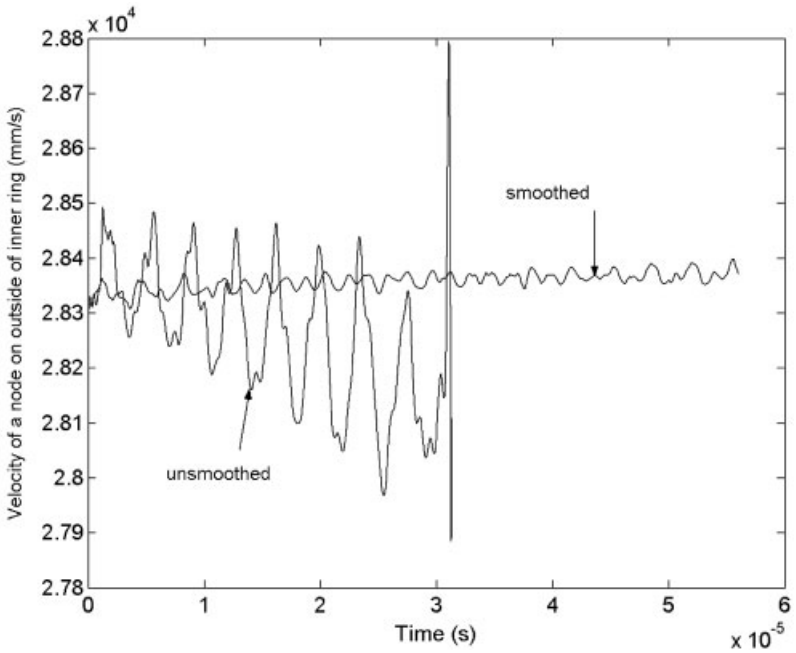

Figure 13. Time histories of circumferential velocity of a node of the inner ring for smoothed and standard master-slave contact.

outer radius. The material is an isotropic Kirchhoff elastic material with the Lame constants given by $\lambda = 121\,730\,\text{Pa}$, $\mu = 81\,154\,\text{Pa}$, density $= 7.8 \times 10^{-9}\,\text{g/m}^3$. An explicit time integrator was used with a time step of 0.03 μs. An initial pressure of 10 000 Pa was applied to the inner ring and the complete system was brought to equilibrium before starting the rotation. The surfaces between the two rings are frictionless. A penalty parameter of $100E$ was used in the contact algorithm.

A quadratic basis was employed for the moving least-squares smoothing of the contact surface. The inner surface of the outer ring is the master surface. Even with the smoothing, some noise appears, so an integrator with high-frequency dissipation, the explicit generalized alpha method of Hulbert and Chung [22], with $\rho_b = 0$ was used. Figure 11 shows the configuration at 40 μs. Figure 12 shows the pressure contours at the same time for the smoothed contact and a standard node-to-edge contact. It can be seen that the pressure varies very smoothly for the solution with the smoothed surfaces. The radial variation of pressure in the inner cylinder is due to the centrifugal body force. The node-to-edge contact shows a very irregular pressure distribution, with peaks around the contacting nodes of the outer ring.

The circumferential velocity at a node on the circumference of the inner cylinder is shown in Figure 13. The velocity for the smoothed case shows a slight drift but is smooth and stable. For the node-to-edge contact, the oscillations are much more severe and an instability occurs in the simulation. These oscillations would depend on the exact implementation of the node-to-edge contact, but seem to be unavoidable in unsmoothed contact of concentric rings.

6. CONCLUSIONS

A method has been developed for smoothing and computing the gap between bodies in contact-impact problems. The method is based on constructing a smooth signed distance function by means of a moving least-squares approximation. This function is directly used to compute the gap function and the normal to the implicitly smooth surface. The smoothed gap function is computed on the 'run', i.e. it is computed for any slave node as needed; no preliminary steps of setting up a smoothed function are needed. Therefore, we call algorithm monolithic.

Both linear and quadratic forms of the algorithm have been tested in two and three dimensions. The quadratic algorithm gives better smoothing for coarse meshes and tends to closely fit the nodes. In fact for circular (or cylindrical in three dimensions) meshes and for spheres, the quadratic fit is exact. However, the linear fit tends to be more robust, although it tends to inscribe curved surfaces. This shortcoming can be ameliorated to some extent by setting a non-zero gap value for contact.

The algorithm is quite fast. Although a system of non-linear equations needs to be solved at each point, the procedure is often far less involved and more readily vectorized than algorithms for gap calculations for piecewise linear surfaces, since in the latter there are often many contingencies that have to be dealt with.

Although the methodology has been presented in the context of a penalty method, it can readily be extended to Lagrange multiplier and augmented Lagrangian methods. There may be some challenge in developing Lagrange multiplier fields that are stable, but preliminary indications are that this is not impossible. The method is applicable to a wide variety of other methods, see References [23–32].

A key feature of the method is how easily it treats three dimensional bodies. Most of the other methods published previously employ spline or polynomial fits for smoothing. These are quite awkward on three-dimensional surfaces of arbitrary meshes. Moving least square approximation, on the other hand, can easily treat very irregular meshes in three dimensions, and as shown by the example, the smoothing is very effective in recovering shapes such as spheres, cylinders and harmonic surfaces.

APPENDIX A: GENERAL ALGORITHMS FOR MLS COEFFICIENTS

In Section 4 an analytic procedure for the determination of the MLS coefficients in the 2-D case using the linear basis has been reported. Here the general case is addressed, including 3-D problems and the more general basis of functions (e.g. quadratic).

The problem is to minimize

$$J_0 = \frac{1}{2} \sum_{I \in S_{\mathbf{x}}} w(\mathbf{x} - \mathbf{x}_I(t))(a_i(\mathbf{x}, t) p_i(\mathbf{x}_I(t))^2 \tag{A1}$$

subject to the condition that the gradient of the signed distance function have unit norm

$$\|\nabla \phi^A(x, t)\| - 1 = 0 \tag{A2}$$

This problem has been solved in Section 4 by analytically finding the saddle point of the Lagrangian functional J. Here the general case is addressed by introducing the augmented Lagrangian functional

$$J_{\mathrm{AL}} = \frac{1}{2} \sum_{I \in S_{\mathbf{x}}} w(\mathbf{x} - \mathbf{x}_I(t))(a_i(\mathbf{x}, t) p_i(\mathbf{x}_I(t))^2 + \mu(\|\nabla \phi^A(x, t)\| - 1) + \frac{1}{2} \alpha(\|\nabla \phi^A(x, t)\| - 1)^2$$

where both the Lagrangian and quadratic penalty terms appear, α being the penalty coefficient. The saddle point of J_{AL} can be determined by several iteration techniques, among which the most widely used is the HP iteration, [33]. It is based on repeating the following steps in sequence until convergence:

(i) Compute the minimum of J_{AL} w.r.t. a_i considering the multiplier μ as a constant;
(ii) Update the value of μ by the formula $\mu := \mu + \alpha(\|\nabla \phi^A(x, t)\| - 1)$;
(iii) Increment the penalty coefficient if the detected convergence rate is too low.

The application of the augmented Lagrangian iteration in the present case necessitates some special techniques due to the particular mathematical structure of the function. In particular two points are noteworthy:

1. The existence of multiple solutions in some cases;
2. The singularity of the Hessian of J_{AL} for particular arrangements of nodes and MLS bases;

The first issue can be solved by appropriately choosing the initial guess of coefficients a_i through a rough estimate of the normal. In the practical application of the method this estimate is always known from the finite element discretization. The second issue arises, for example, when a plane or a slightly curved surface is to be represented by a quadratic basis.

This difficulty has been overcome by modifying the solver in step (i) of the iteration. The minimum is determined by a Quasi Newton's method where the new coefficients a_i at the iteration $(k + 1)$ are given by

$$\mathbf{a}^{(k+1)} = \mathbf{a}^{(k)} - \mathbf{H}^{(k)} \nabla J_{\mathrm{AL}}(\mathbf{a}^{(k)})$$

where $\mathbf{H}^{(k)}$ is determined from the Hessian of J_{AL} by adding to it a scaled identity matrix as follows:

$$\mathbf{H} = \nabla^2 J_{\mathrm{AL}} + 10^{-6}\,\mathrm{Tr}(\nabla^2 J_{\mathrm{AL}})\mathbf{I}$$

where Tr is the trace and $\mathbf{I}$ the identity matrix. Thus the identity matrix (i.e., steepest descent bias) is added to the original, nearly singular Hessian. As the original Hessian is always positive semidefinite, the modified matrix $\mathbf{H}$ is always non-singular. Note that in the formula the trace gives a rough estimate of the spectral radius, while the factor 10^{-6} is the stepsize of the steepest descent bias. This stepsize is small enough not to alter the properties of Newton's method but eliminates the singularity.

ACKNOWLEDGEMENT

The support of the Office of Naval Research is gratefully acknowledged.

REFERENCES

1. Kikuchi N. A smoothing technique for reduced integration penalty method in contact problems. *International Journal for Numerical Methods in Engineering* 1982; **18**:343–350.
2. Wriggers P, Vu Van, Stein E. Finite element formulation of large deformation contact-impact problems with friction. *International Journal for Numerical Methods in Engineering* 1990; **37**:319–331.
3. Wriggers P, Krstulovic-Opara L, Korelc J. Smooth C-1 interpolations for two-dimensional frictional contact problems. *International Journal for Numerical Methods in Engineering* 2001; **51**(12):1469–1495.
4. El-Abbasi N, Meguid SA, Czekanski A. On the modeling of smooth contact surfaces using cubic splines. *International Journal for Numerical Methods in Engineering* 2001; **50**(4):953–967.
5. Puso MA, Laursen TA. A 3D contact smoothing method using Gregory patches. *International Journal for Numerical Methods in Engineering* 2002; **54**(8):1161–1194.
6. McDevitt TW, Laursen TA. A mortar finite element formulation for frictional contact problems. *International Journal for Numerical Methods in Engineering* 2000; **48**(10):1525–1547.
7. Laursen T, Love GR. Improved implicit integrators for transient impact problems—geometric admissibility within the conserving framework. *International Journal for Numerical Methods in Engineering* 2002; **53**(2):245–274.
8. El-Abbasi N, Bathe KJ. Stability and patch test performance of contact discretizations and a new solution algorithm. *Computer Structures* 2001; **79**(16):1473–1486.
9. Jones RE, Papadopoulos P. A novel three-dimensional contact finite element based on smooth pressure interpolations. *International Journal for Numerical Methods in Engineering* 2001; **51**(7):791–811.
10. Kloosterman G, van Damme RMJ, van den Boogaard AH *et al.* A geometrical-based contact algorithm using a barrier method. *International Journal for Numerical Methods in Engineering* 2001; **51**(7):865–882.
11. Bajer A, Demkowicz L. Dynamic contact/impact problems, energy conservation and gear trains, TICAM report 01-12.
12. Stupkiewicz S. Extension of the node-to-segment contact element for surface-expansion dependent contact laws. *International Journal for Numerical Methods in Engineering* 2001; **50**(3):739–759.
13. Belytschko T, Liu WK, Moran B. *Nonlinear Finite Elements for Continua and Structures.* Wiley: New York, 2000.
14. Wriggers P. Finite element algorithms for contact problems. *Archives of Computational Methods in Engineering* 1995; **2**(4):1–49.
15. Wriggers P, Miehe C. Contact constraints within coupled thermomechanical analysis—a finite element model. *Computer Methods in Applied Mechanics and Engineering* 1994; **113**:301–319.
16. Belytschko T, Lu YY, Gu L. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering* 1994; **37**:229–256.
17. Belytschko T, Law SE. An assembled surface normal algorithm for interior node removal in three-dimensional finite element meshes. *Engineering with Computers* 1985; **1**:55–60.
18. Pandolfi A, Kane C, Marsden JE, Ortiz M. Time discretized variational formulation of non-smooth frictional contact. *International Journal for Numerical Methods in Engineering* 2002; **53**(8):1801–1829.
19. Krongauz Y, Belytschko T. EFG Approximations with discontinuous derivatives. *International Journal for Numerical Methods in Engineering* 1998; **41**:1215–1233.

20. Belytschko T, Moes N, Usui S, Parimi C. Arbitrary discontinuities in finite elements. *International Journal for Numerical Methods in Engineering* 2001; **50**(4):993–1013.
21. Belytschko T, Lin JI. A three-dimensional impact-penetration algorithm with erosion. *Computers and Structures* 1987; **25**(6):909–918.
22. Hulbert GM, Chung JT. Explicit time integration algorithms for structural dynamics with optimal numerical dissipation. *Computer Methods in Applied Mechanics* 1996; **137**(2):175–188.
23. Armero F, Petocz E. A new dissipative time-stepping algorithm for frictional contact problems: formulation and analysis. *Computer Methods in Applied Mechanics and Engineering* 1999; **179**:151–178.
24. Bathe KJ, Chaudhary A. A solution method for planar and axisymmetric contact problems. *International Journal for Numerical Methods in Engineering* 1985; **21**(1):65–88.
25. Belytschko T, Neal MO. Contact-impact by the Pinball algorithm with Penalty and Lagrangian methods. *International Journal for Numerical Methods in Engineering* 1991; **31**:547–572.
26. Demkowicz J, Oden T. On some existence and uniqueness results in contact problems with nonlocal friction. Texas Institute of Computational Mechanics (TICOM). *Report* 81-13, University of Texas, Austin, 1981.
27. Heinstein MW, Mello FJ, Attaway SW, Laursen TA. Contact-impact modelling in explicit transient dynamics. *Computer Methods in Applied Mechanics and Engineering* 2000; **187**:621–640.
28. Kikuchi N, Oden JT. *Contact Problems in Elasticity*: *A Study of Variational Inequalities and Finite Element Methods.* SIAM: Philadelphia, PA, 1988.
29. Laursen TA, Simo JC. A continuum-based finite element formulation for the plicit solution of multi-body, large deformation frictional contact problems. *International Journal for Numerical Methods in Engineering* 1993; **36**:3451–3485.
30. Padmanabhan V, Laursen TA. A framework for development of surface smoothing procedures in large deformational friction contact analysis. *Finite Elements in Analysis and Design*, 2001; **37**:173–198.
31. Padmanabhan V, Laursen TA. Surface smoothing procedure for large deformation contact analysis. *Finite Elements in Analysis and Design* 2002, to appear.
32. Wang F, Cheng J, Yao Z. FFS contact searching algorithm for dynamic finite element analysis. *International Journal for Numerical Methods in Engineering* 2001; **52**:655–672.
33. Fletcher R. *Practical Methods of Optimization.* Wiley: New York, 1987.

