

# A Lookup-Table-Based Approach for Spatial Analysis of Contact Problems

**Margarida Machado**

CT2M, Mechanical Engineering Department,  
University of Minho, Campus Azurém,  
Guimarães 4800-058, Portugal  
e-mail: margarida@dem.uminho.pt

**Paulo Flores**

CT2M, Mechanical Engineering Department,  
University of Minho, Campus Azurém,  
Guimarães 4800-058, Portugal  
e-mail: pflores@dem.uminho.pt

**Jorge Ambrósio**

IDMEC, Instituto Superior Técnico,  
University of Lisbon, Av. Rovisco Pais 1,  
Lisbon 1049-001, Portugal  
e-mail: jorge@dem.ist.utl.pt

*The aim of this work is to present an efficient methodology to deal with general 3D-contact problems. This approach embraces three steps: geometrical definition of 3D surfaces, detection of the candidate contact points, and evaluation of the contact forces. The 3D-contact surfaces are generated and represented by using parametric functions due to their simplicity and ease in handling freeform shapes. This task is carried during preprocessing, which is performed before starting the multibody analysis. The preprocessing procedure can be condensed into four steps: a regular and representative collection of surface points is extracted from the 3D-parametric surface; for each point the tangent vectors to the  $u$  and  $v$  directions of the parametric surface and the normal vector are computed; the geometrical information on each point is saved in a lookup table, including the parametric point coordinates, the corresponding Cartesian coordinates, and the components of the normal, tangent, and bitangent vectors; the lookup table is rearranged such that the  $u$ - $v$  mapping is converted into a 2D matrix being this surface data saved as a direct access file. For the detection of the contact points, the relative distance between the candidate contact points is computed and used to check if the bodies are in contact. The actual contact points are selected as those that correspond to the maximum relative indentation. The contact forces are determined as functions of the indentation or pseudopenetration, impact velocity, and geometric and material properties of the contacting surfaces. In general, lookup tables are used to reduce the computation time in dynamic simulations. However, the application of these schemes involves an increase of memory needs. Within the proposed approach, the amount of memory used is significantly reduced as a result of a partial upload into memory of the lookup table. A slider-crank mechanism with a cup on the top of the slider and a marble ball are used as a demonstrative example. A contact pair is considered between a cup and a marble ball, the contact forces for which are computed using a dissipative contact model.*

[DOI: 10.1115/1.4026894]

## 1 Introduction

The problem of contact is of paramount importance in the field of multibody dynamics because contact events can take place frequently and in many cases the function of multibody systems is based on them [1–4]. Contact modeling and analysis have received a large amount of attention over the past few decades and still remain an active field of investigation [1–10]. Proper representation of the contact phenomenon for multibody dynamics is still a challenge since it depends on factors such as the geometry of the contacting surfaces, the material properties of the contacting bodies, and the constitutive law considered representing the interaction among the different bodies that comprises the multibody systems [10–12]. Even for cases in which no surface-to-surface contact exists but there is simply loading of a deformable media in between, such as rubber bushings [13] or lubrication fluid [14], the efficient numerical characterization of the surface is of utmost importance.

Over the years, different methods have been developed to handle the contact detection problem. Hahn and Wriggers [15] proposed an explicit multibody contact algorithm where the contact detection issue was studied using a predictor-corrector scheme. An iterative form of the proposed algorithm was used to reduce the computational effort. One of the most robust and well-known methods for contact detection of complexly shaped bodies was proposed by Hippmann [16]. This algorithm, referred to as the polygonal contact model, is based on representation of the body

surfaces by polygon meshes and the contact force evaluation is done using an elastic foundation model. Wellmann et al. [17] developed an efficient contact detection algorithm for superellipsoids based on the common-normal concept. This contact detection approach was formulated as a two-dimensional unconstrained optimization problem that is solved by a combination of Newton's method and a Levenberg–Marquardt method. Lopes et al. [18] presented a methodology for contact detection between quadric and superquadric surfaces using their implicit equations.

Analytical functions of regular shapes, such as planes, spheres, and ellipsoids, are, in general, the best choice to describe simple contact geometries [18]. When the contact surfaces present complex configurations, more sophisticated fitting approaches must be considered to obtain an accurate surface representation [19,20]. However, the advanced fitting techniques take too much computational time, which may penalize the global efficiency of the method. This is the reason why some authors split complex geometries using multiple regular shapes [21,22]. Choi et al. [23] developed a new collision detection method called triangle soup average plane contact (TSAPC), which is quite robust and efficient and can be applied to convex and concave geometries without the need to use any graphics hardware. There are some collision detection methods, developed for computer graphics applications such as axis-aligned bounding box (AABB) trees, or oriented bounding box (OBB) trees that have been widely used to improve the performance of contact detection algorithms [12,23]. These bounding volumes hierarchies are refined methods based on polygonal meshes and are available in some software packages such as RAPID, I-COLLIDE, V-COLLIDE, SOLID, and V-Clip [24]. In general, these approaches are computationally

Contributed by the Design Engineering Division of ASME for publication in the JOURNAL OF COMPUTATIONAL AND NONLINEAR DYNAMICS. Manuscript received July 9, 2013; final manuscript received February 16, 2014; published online July 11, 2014. Assoc. Editor: Javier Cuadrado.

costly and can only be applied to the case of convex contact geometries.

It has been recognized by many researchers that most of the time consumed in modeling and analyzing contact problems is spent on the contact detection task. This fact is crucial in multiple impact scenarios [25], such as railway multibody models [26] and biomechanical contact situations [27]. The dynamic study of these types of multibody systems may include hundreds of function evaluations of the candidate contact points, demanding fast and efficient methods to compute such interactions. Such a problem was approached by Pombo and Ambrosio in the framework of curve geometric studies to describe rollercoaster and railway tracks [26,28]. Thus, it is the objective of this work to address this important problem and to propose a general methodology that allows for the accurate and efficient modeling and analysis of contact between freeform surfaces.

This paper is organized as follows. Section 2 deals with the generation and representation of 3D surfaces. The main issues associated with implicit and parametric surfaces are also presented in this section. In Sec. 3, a general and straightforward methodology for contact detection is developed together with a computational efficient approach. The algorithm proposed also includes a scheme that allows for reading into memory only the portion of the contacting surfaces close to the contact zone. This is a crucial aspect for the reduction of the computational effort of the contact detection task. Section 4 describes the constitutive law used to characterize the contact forces developed in the contact area. In Sec. 5, the formulation of the equations of motion for constrained multibody systems is revisited, in which the general contact detection methodology developed is embedded. The procedures adopted throughout this work are presented and analyzed in Sec. 6 with the help of a numerical simulation of a slider-crank mechanism with a slider cup and a falling marble ball. Finally, the summary and the main conclusions of this work are provided in Sec. 7.

## 2 Geometrical Definition of 3D Surfaces

The efficiency of a contact detection process relies upon the geometric description of the contact surfaces, which can be represented by means of polygonal or nonpolygonal models [29]. Polygonal models are frequently applied to model complex shapes. Nonetheless, when the contact detection method demands a continuous representation, nonpolygonal models are used. The nonpolygonal models can be classified into three main groups, namely constructive solid geometry (CSG), implicit methods, and parametric functions.

The use of CSG models is easy and straightforward, but it is restricted because the CSG representations have to be described by a set of Boolean operations instead of a single mathematical expression. Regarding the implicit method, a generic 3D surface is represented by an implicit mathematical function that defines the location of the points belonging to a surface and can be written as

$$f(x, y, z) = 0 \quad (1)$$

where  $x$ ,  $y$ , and  $z$  are the Cartesian coordinates of a generic point located on the surface. By analyzing Eq. (1), it can be stated that within an implicit function, the three Cartesian coordinates are not independent. This is a disadvantage of the implicit method because it does not allow for generating, in a systematic manner, a set of consecutive points located on a surface. Furthermore, efficient rendering and accurate modeling of sharp edges is usually a hard task whenever implicit representations are utilized [30]. Finally, the parametric surface representation can be expressed as

$$\mathbf{s}(u, v) = \begin{cases} x = x(u, v) \\ y = y(u, v) \\ z = z(u, v) \end{cases} \quad (2)$$

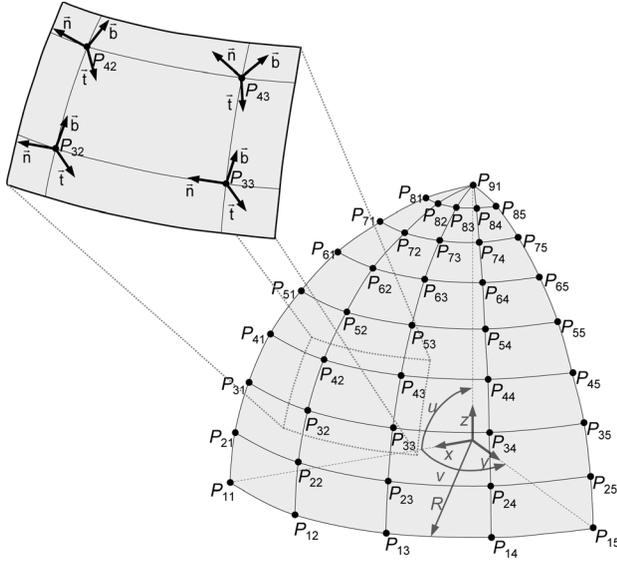
in which  $x$ ,  $y$ , and  $z$  are the Cartesian coordinates of a generic surface point and  $u$  and  $v$  denote the corresponding parametric coordinates, i.e.,  $\mathbf{s}(u, v)$  represents the parametric position vector of a surface. The major advantage of parametric representation is that it allows for the reduction of a 3D problem to the bidimensional domain, avoiding the use of complex and time-consuming numerical solutions. Nevertheless, a parametric surface is difficult to ray trace in the sense that there is no direct and appropriate approach to check if a given point in the 3D space belongs to a surface. Even so, parametric representations have been broadly used in computer-aided design (CAD) for surface modeling purposes. Farin et al. [31] pointed out two main reasons for the wide application of the parametric representations: (i) parametric surface patches can be pieced together with any desired degree of continuity and (ii) there are many intuitively meaningful techniques for controlling their shape. A comparison between implicit and parametric surface functions is offered in Table 1.

**2.1 Surface Representation.** In the present work, parametric functions are used to represent freeform contact surfaces like, for instance, the articular surfaces of the human knee. A freeform surface can be described parametrically by a single patch, such as in the case of a spherical surface, or by an assembly of multiple patches, as, for instance, those used to define ship hulls. For contact analysis purposes, it is crucial and desirable to have continuity between adjacent patches, i.e., the patches have to be fitted in such way that their boundaries are imperceptible, in order to ensure numerical stability and computational efficiency to the contact algorithm. From a mathematical point of view, these properties are related to the concept of geometric continuity ( $G$ ) at different levels, namely positional ( $G_0$ ), tangential ( $G_1$ ), and curvature ( $G_2$ ) [33].

In the particular case of parametric surface representation, the derivatives of the surface with respect to the parametric coordinates can be easily studied. This mathematical procedure is usually denominated as parametric continuity. The parametric continuity of first and second levels  $C_0$  and  $C_1$  can be considered as identical to the positional and tangential continuities. In turn, the third level of parametric continuity  $C_2$  differs from the curvature continuity in the measure that its parameterization is also continuous [34]. In short, the parametric method allows for an efficient contact search algorithm for generating and representing arbitrary surfaces that offers great flexibility and precision for handling freeform shapes. It should be highlighted that the implicit and parametric approaches can exhibit complementary characteristics and it may be convenient to convert from one form to another or combine both representations in a hybrid surface model [30]. It must be noted that although parametric, or at least geometric, continuity of a given order is desirable most of the contact-detecting tasks in contact mechanics do not have any specific requirements in terms of continuity. That is the case in the applications of the procedures proposed in this work.

**Table 1 Comparison between the geometrical properties of implicit and parametric surface functions [32]**

Property	Implicit surface	Parametric surface
General expression	$f(x, y, z) = 0$	$\mathbf{s} = \mathbf{s}(u, v)$
Accuracy	yes	yes
Intuitive specification	no	yes
Local support	no	yes
Arbitrary topology	yes	no
Guaranteed continuity	yes	yes
Efficient display	no	yes
Efficient intersections	yes	no
Efficient rendering	no	yes



**Fig. 1 Schematic representation of one-eighth of a spherical surface using the parametric method**

The conversion from parametric to implicit form, known as implicitization, is feasible. Conversely, the parameterization is not always possible because the class of implicit surfaces is much larger than that of parametric surfaces. Because of the issues related with parametrization and implicitization of curves and surfaces complex topics of computational geometry, the methods used for such purposes are not discussed here. References [35,36] provide the interested reader with different approaches for these conversions. In any case, in this work only parametrized surfaces are considered.

In the parametric method, the points that belong to a surface are given by a collection of mappings, which relate the space parameters to the object surface. As mentioned, a parametric surface maps a 2D domain into a 3D space. To better understand how the fundamental ingredients necessary to generate and represent a surface by the parametric method are handled, consider one-eighth of a spherical surface, as shown in Fig. 1. This surface can be described by a single patch and the spherical polar coordinates can be considered to be the  $u$  and  $v$  parameters. The surface radius is denoted by  $R$ . A surface-fixed coordinate system is attached at the surface geometric center. The normal, tangent, and binormal vectors to the surface at points  $P_{32}$ ,  $P_{42}$ ,  $P_{43}$ , and  $P_{33}$  are shown in Fig. 1.

The parametric surface of Fig. 1 can be expressed by a  $u$ - $v$  mapping as

$$s(u, v) = \begin{cases} x = R\sqrt{\cos^2 u \cos^2 v} & 0 \leq u \leq \frac{\pi}{2} \quad [\text{rad}] \\ y = R\sqrt{\cos^2 u \sin^2 v} & 0 \leq v < \frac{\pi}{2} \quad [\text{rad}] \\ z = R\sqrt{\sin^2 u} & \end{cases} \quad (3)$$

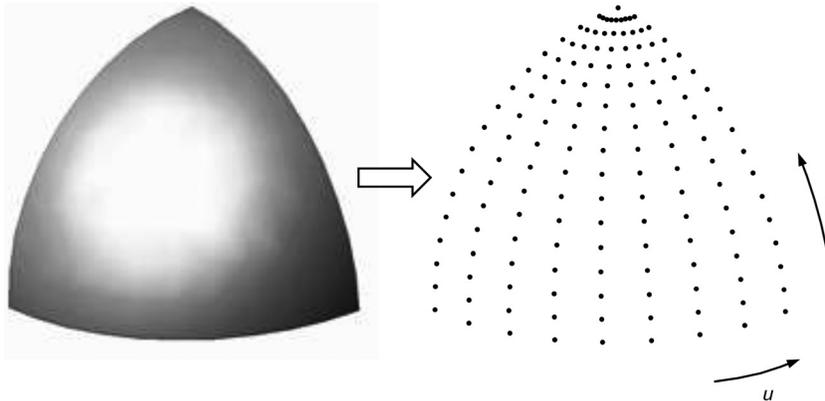
which allows for the determination of the Cartesian coordinates of the points located on the spherical surface. A dataset with a total of 14 quantities is necessary to fully characterize each point located on this surface, namely the point parametric coordinates ( $u, v$ ), the Cartesian coordinates ( $x, y, z$ ), the Cartesian components of the normal vector ( $\mathbf{n}_x, \mathbf{n}_y, \mathbf{n}_z$ ), and the Cartesian components of the tangent and binormal vectors ( $\mathbf{t}_x^u, \mathbf{t}_y^u, \mathbf{t}_z^u, \mathbf{t}_x^v, \mathbf{t}_y^v, \mathbf{t}_z^v$ ). This geometric information plays a key role in the contact detection algorithm because the contact between two generic convex surfaces can be established by solving a set of nonlinear equations that represent the collinearity between the normal and the distance vectors [37].

**2.2 Surface Preparation.** After describing the contact surface by parametric functions, it is necessary to store this geometric representation in a file. Some of the most common formats of geometry files, such as STEP, IGES, and STL, require large memory sizes for storage. Consider the case of contacts among many objects with equally complex geometrical shapes. The storage of the complete parameterized geometries of all objects may take all available physical memory and require slow disk reading and writing operations to handle the virtual memory. As a result, the reading procedure of these files is memory consuming and, hence, significantly penalizes the computational efficiency of a contact algorithm. This drawback is even more critical in contact analysis requiring the computation of normal, tangent, and binormal vectors in each instant of the simulation. To overcome these difficulties and achieve reasonable computation times, a surface preparation is proposed. This is a preprocessing procedure that can be condensed in the following steps:

- (1) A regular and representative collection of surface points is extracted from the 3D parametric surface that has been generated analytically or modeled in appropriate CAD software. This step is usually denominated regular surface sampling and is illustrated in Fig. 2.
- (2) For each point in the surface, tangent vectors to the  $u$  and  $v$  directions, referred to as tangent and binormal vectors, respectively, and the normal vector are evaluated:

$$\mathbf{t}^v \equiv \mathbf{t}^v(u, v) = \frac{\partial \mathbf{s}(u, v)}{\partial v} \quad (4a)$$

$$\mathbf{t}^u \equiv \mathbf{t}^u(u, v) = \frac{\partial \mathbf{s}(u, v)}{\partial u} \quad (4b)$$



**Fig. 2 Schematic representation of the regular surface sampling**

$$\mathbf{n} \equiv \mathbf{n}(u, v) = \frac{\tilde{\mathbf{t}}^u \tilde{\mathbf{t}}^v}{\|\tilde{\mathbf{t}}^u \tilde{\mathbf{t}}^v\|} \quad (4c)$$

where the tilde ( $\sim$ ) placed over a vector indicates that the components of the vector are used to generate the skew-symmetric matrix required for the cross product [38]. In order to obtain the orthonormal referential associated to each contact point, the tangent vector  $\mathbf{t}$  and the binormal vector  $\mathbf{b}$  are obtained as

$$\mathbf{t} = \tilde{\mathbf{t}}^u / \|\tilde{\mathbf{t}}^u\| \quad (5)$$

$$\mathbf{b} = \tilde{\mathbf{n}} \mathbf{t} \quad (6)$$

Note that in the calculation of the normal vector it is assumed that there is no parametrization degeneration and, therefore,  $\tilde{\mathbf{t}}^u$  and  $\tilde{\mathbf{t}}^v$  are never parallel.

- (3) Once the regular 3D points surface collection has been established, the geometric information on each point is saved in the form of a lookup table. In the present case, the lookup table is composed by 14 columns, this being the number of rows equal to the number of points used to represent the surface. However, it can include more columns depending on the problem at hand, which can require, for instance, local material and mechanical properties of the surface, curvature radius, and generalized stiffness or coefficient of restitution, all necessary to the evaluation of the contact forces.
- (4) The final step of the surface preparation, named lookup table reshuffle, deals with the rearrangement of the lookup table in which the  $u$ - $v$  mapping is transformed into a 3D matrix form (see Fig. 3). In this process, the surface data are split in equal-sized records, i.e., it is considered that the discretized parametrization is equally spaced in the  $u$  and  $v$  directions, and saved as a direct access file. In contrast to the sequential file, the direct file permits reading the data in any order, which is convenient for contact point searching.

In short, the proposed methodology to generate and represent a freeform surface is quite straightforward and simple to implement in any general-purpose multibody code. Moreover, the way in which the surface data are organized and stored in the preprocessing scheme allows for the partial reading of the surface data, i.e., only the surface data corresponding to the contact region are read and stored in physical memory. This surface portion is named storage window and is discussed in detail in Sec. 3. It is worth mentioning that the density of the points considered during sampling is arbitrary and, therefore, it is a user decision based on the geometry that is being represented. Furthermore, complementary measures to optimize the loading procedure, including memory mapping, can be investigated to improve the access speed to surface data.

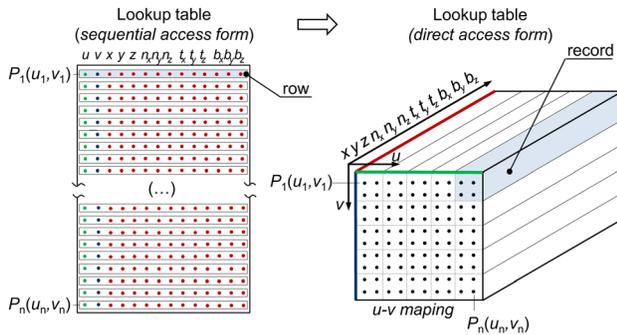


Fig. 3 Schematic representation of the lookup table reshuffle of a surface defined with 100 points that are stored in 25 records

### 3 Methodology for Contact Detection

A straightforward formulation for the contact detection of generalized surfaces is used here. Consider two moving surfaces, belonging to bodies  $i$  and  $j$ , with absolute velocities  $\dot{\mathbf{r}}_i$  and  $\dot{\mathbf{r}}_j$ , respectively. Figure 4 shows these two moving surfaces, which are represented by means of parametric functions. Both contacting surfaces are assumed to be convex at least in the neighborhood of the candidate contact points. The surface parameters  $u$  and  $v$  are ordered such that the vector  $\mathbf{n}$  becomes the outward normal. The center of mass of bodies  $i$  and  $j$  are  $O_i$  and  $O_j$ , while the origin of the surfaces are denoted by  $Q_i$  and  $Q_j$ . Local coordinate systems  $\xi_i^O, \eta_i^O, \zeta_i^O$  and  $\xi_j^O, \eta_j^O, \zeta_j^O$  are attached at the center of mass of each body and to the origin of each surface, respectively.  $P_i$  and  $P_j$  represent the candidate contact points. The geometric and positional vectors relevant for the contact detection process are depicted in Fig. 4.

**3.1 General Formulation.** With reference to Fig. 4, the minimum distance vector  $\mathbf{d}$ , which connects the candidate contact points, is calculated as

$$\mathbf{d} = \mathbf{r}_i^P - \mathbf{r}_j^P \quad (7)$$

where both  $\mathbf{r}_i^P$  and  $\mathbf{r}_j^P$  are described in global coordinates with respect to the inertial reference frame [38]

$$\mathbf{r}_k^P = \mathbf{r}_k + \mathbf{A}_k^O (\mathbf{s}_k^{iO} + \mathbf{A}_k^Q \mathbf{s}_k^{jP}), \quad (k = i, j) \quad (8)$$

in which  $\mathbf{r}_k$  is the global position vector of body  $k$ ,  $\mathbf{s}_k^{iO}$  is the local component of  $Q_k$  with respect to body reference frame, and  $\mathbf{s}_k^{jP}$  is the local component of  $P_k$  with respect to the surface reference frame.  $\mathbf{A}_k^O$  and  $\mathbf{A}_k^Q$  ( $k = i, j$ ) are the rotational transformation matrices of the body reference frame and surface reference frame, respectively.

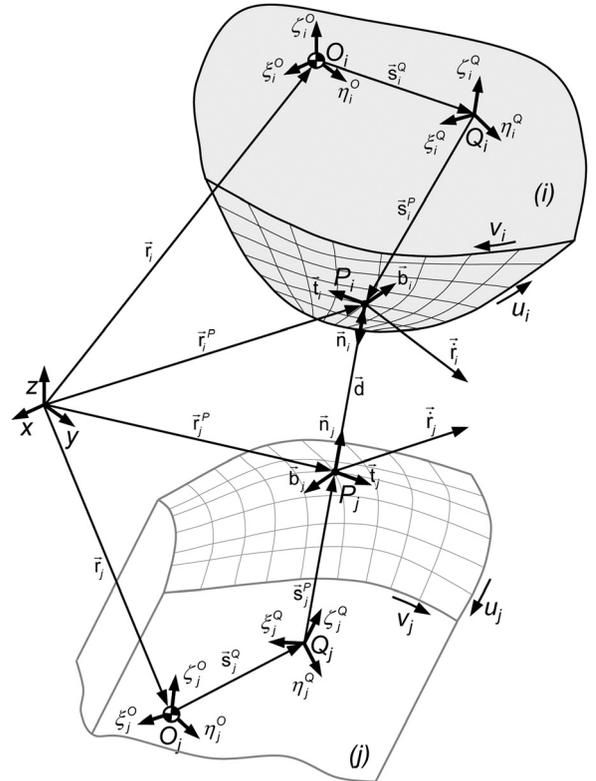


Fig. 4 Representation of two generalized contact surfaces

The magnitude of the distance vector is evaluated as

$$\delta = \sqrt{\mathbf{d}^T \mathbf{d}} \quad (9)$$

The minimum distance condition is not enough to guarantee that a pair of points is effectively the actual contact pair of points. The conditions for contact and a search method for the contact points have been proposed by Pombo et al. [26] and proceed as follows. In order to ensure that  $P_i$  and  $P_j$  are the actual contact points, the surface normal vectors  $\mathbf{n}_i$  and  $\mathbf{n}_j$  have to be collinear with the distance vector  $\mathbf{d}$ , as Fig. 4 depicts. These geometric conditions can be expressed by two cross-products between vectors  $\mathbf{d}$  and  $\mathbf{n}_i$ , and  $\mathbf{d}$  and  $\mathbf{n}_j$ . Because the dot products are more convenient for computations, the collinear geometric conditions previously described can be written by a set of dot products as

$$\tilde{\mathbf{d}}\mathbf{n}_i = \mathbf{0} \equiv \begin{cases} \mathbf{d}^T \mathbf{t}_i = 0 \\ \mathbf{d}^T \mathbf{b}_i = 0 \end{cases} \quad (10)$$

$$\tilde{\mathbf{d}}\mathbf{n}_j = \mathbf{0} \equiv \begin{cases} \mathbf{d}^T \mathbf{t}_j = 0 \\ \mathbf{d}^T \mathbf{b}_j = 0 \end{cases} \quad (11)$$

where  $\mathbf{t}_k$  and  $\mathbf{b}_k$  ( $k = i, j$ ) denote the tangent and binormal vectors illustrated in Fig. 4.

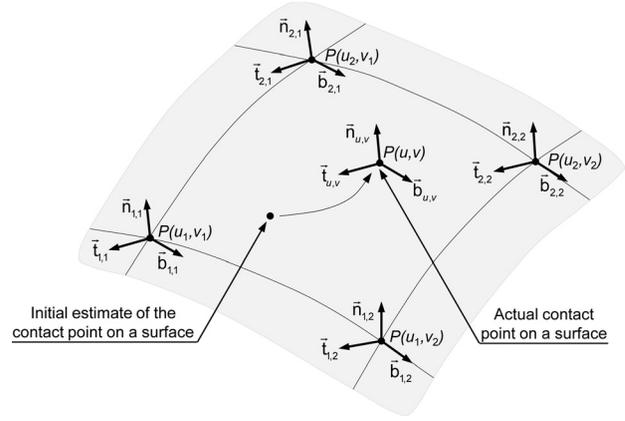
The geometric conditions given by Eqs. (10) and (11) constitute four nonlinear equations with four unknowns, i.e., the parametric coordinates  $u_i$ ,  $v_i$ ,  $u_j$ , and  $v_j$ . This system of nonlinear equations, which can be solved using an iterative method such as the Newton–Raphson method, provides the solution for the location of the candidate contact points [26]. Once the candidate contact points are determined, the next step deals with the evaluation of their relative distance using Eq. (9). Finally, it is necessary to verify the indentation condition, which ensures that contact exists, i.e., the candidate contact points are actual contact points. This condition is given by

$$\mathbf{d}^T \mathbf{n}_j \leq 0 \quad (12)$$

In short, the positions of an actual contact pair of points between two freeform surfaces cannot be predicted a priori due to the influence of the kinematic constraints and other interactions on the bodies of the complete system. Therefore, during dynamic simulation the evaluation of the actual contact pair of points requires the resolution of the system of nonlinear equations, given by Eqs. (10) and (11). For this purpose, an initial estimate of the contact points on both surfaces needed to be provided. Then, the information obtained from the previous time step is used as an initial guess to find the actual contact point at the current instant of time. With this procedure, only a few iterations are required to achieve the desired solution.

**3.2 Lookup-Table-Based Approach.** Within the proposed contact approach, the contacting surfaces are described by a set of points that is stored in a lookup table. As a result, geometric data of the actual contact point have to be evaluated whenever this point does not belong to the regular sample of points stored in the lookup table during preprocessing. The geometric data that need to be computed are the data relevant for the contact detection process, that is, Cartesian point coordinates, normal vector, tangent vector, and bitangent vector. The evaluation of this geometric data is carried out by means of bilinear interpolation, which is depicted in Fig. 5 [39].

A bilinear interpolation of the 12 point coordinates and vector components is performed in each instant of simulation in order to evaluate all the relevant geometric information for the contact detection process. Since the bilinear interpolation is an extension of the linear case for interpolating functions of two variables on a regular grid, it is considered that this numerical procedure does not penalize the computational efficiency of the proposed contact

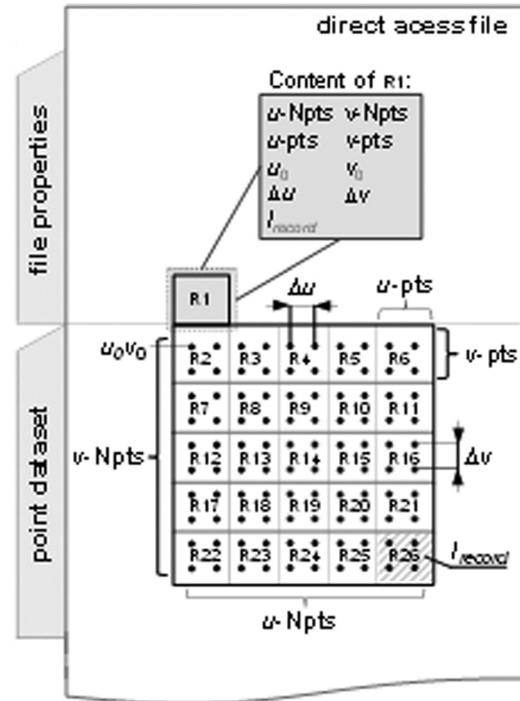


**Fig. 5 Schematic representation of the bilinear interpolation for a surface described by four points**

approach [39]. It is worth mentioning that a more accurate solution for the location of the contact points is achieved when a higher surface discretization is utilized, i.e. a higher number of points is used to describe the surfaces.

During the preprocessing, the geometric data of each surface are organized and stored in a direct access file. Within a direct access file, the dataset is organized and divided into equal-sized records. Each record is identified by an index number and all records have the same length, which is defined when the file is opened. As the name suggests, the direct access files permit direct access to a particular record of the file and facilitate the operations of reading, deleting, updating, and inserting records into the file. The direct access files allow for a partial reading of the surface data, i.e., the reading of only a surface portion that includes the candidate contact point and some points nearby. This characteristic avoids storing the complete surface data in physical memory.

Figure 6 shows how the dataset of a parametric representation of a contact surface is organized and stored in a direct access file. The point dataset is saved in the record number two and



**Fig. 6 Structure of a direct access file of a parametric representation of a surface with 100 points**

subsequent records. The record number one stores some properties of the surface file, namely the number of point per surface on  $u$ -direction ( $u$ -Npts), the number of points per surface on  $v$ -direction ( $v$ -Npts), the number of point per record on  $u$ -direction ( $u$ -pts), the number of points per record on  $v$ -direction ( $v$ -pts), the lower bound of surface on  $u$ -direction ( $u_0$ ), the lower bound of surface on  $v$ -direction ( $v_0$ ), the  $u$ -increment ( $\Delta u$ ), the  $v$ -increment ( $\Delta v$ ), and the record length ( $l_{\text{record}}$ ). The record length  $l_{\text{record}}$  is given by

$$l_{\text{record}} = n_{pr} n_d n_{\text{bit}} \quad (13)$$

where  $n_{pr}$  is the number of points per record;  $n_d$  is the number of parameters that are stored into the lookup table for each point (in this study,  $n_d = 14$ );  $n_{\text{bit}}$  is assigned to 64 and is related to the memory space needed to store each parameter (datum size).

**3.3 Storage Window.** The partial reading of a surface file is suitable for contact point searching, in particular when the surface file is very large or when different rigid bodies share common geometric features. In these cases, the storage window must be located at the contact zone, that is, it should contain the records to which the candidate contact points belong. Figure 7(a) shows one-eighth of a spherical surface, previously illustrated in Fig. 1, with a storage window that includes the contact point at the instant of time  $t$ . If at the next instant of time,  $t + \Delta t$ , the contact point remains in a region close to the previous contact point, the storage window does not need to be updated. In contrast, when the next contact point is distant from the previous one, the storage window has to be updated, as Fig. 7(b) depicts.

The necessity for updating the storage window is checked in each contact calculation. This process can be summarized and condensed by the following steps:

- (1) Locate contact record, i.e., the record that contains the initial guess ( $u_0, v_0$ ).
- (2) Check if contact record belongs to storage window. If not, go to step (4).
- (3) Check if contact record is a border record. If not, go to step (7).
- (4) Locate storage window considering the contact record the center of the new storage window.
- (5) Store the index of the records of the new storage window, namely the first record and the border records.
- (6) Read and store the data of the records that belong to the new window.
- (7) Proceed with the contact computation.

The possibility to partially read the surface is useful in the cases where the contact point moves slightly and smoothly and, therefore, it remains in the vicinity of the previous contact point. Thus, the amount of memory used is significantly reduced and the contact detection process is more efficient.

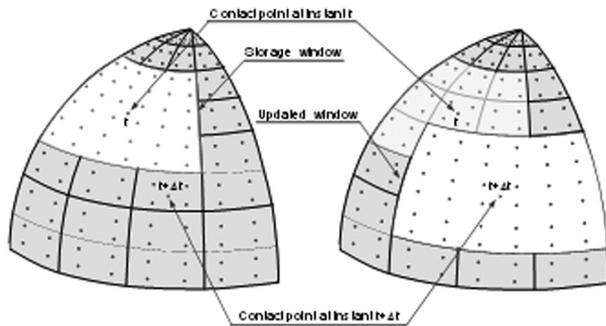


Fig. 7 Representation of an update of a storage window

## 4 Evaluation of the Contact-Impact Forces

The normal contact forces developed in an elastic contact can be modeled by applying Hertz law, which can be written as

$$F_N = K \delta^n \quad (14)$$

where  $K$  is a generalized stiffness parameter and  $\delta$  is the relative indentation, given by Eq. (9). The exponent  $n$  is equal to  $3/2$  for the case where there is a parabolic distribution of contact stresses. The generalized stiffness parameter depends on the geometry and physical properties of the contacting surfaces. The Hertz contact law does not account for the energy dissipation process that characterizes the contact-impact events in mechanical systems [10]. This issue has led several researchers to extend the Hertz law to accommodate energy dissipation in the form of internal damping by

$$F_N = K \delta^n + D \dot{\delta} \quad (15)$$

where the first term corresponds to the elastic force and the second term to the energy dissipated during the impact. In Eq. (15), the quantity  $D$  is the hysteresis coefficient and  $\dot{\delta}$  is the relative normal contact velocity. The hysteresis coefficient is given by

$$D = \chi \delta^n \quad (16)$$

In turn, the damping term  $\chi$  has different expressions depending on the approach considered, which may be valid for very elastic and/or inelastic contacts [8]. For instance, the hysteresis factor for the Flores et al. model is given by

$$\chi = \frac{8(1 - c_r) K}{5c_r \dot{\delta}^{(-)}} \quad (17)$$

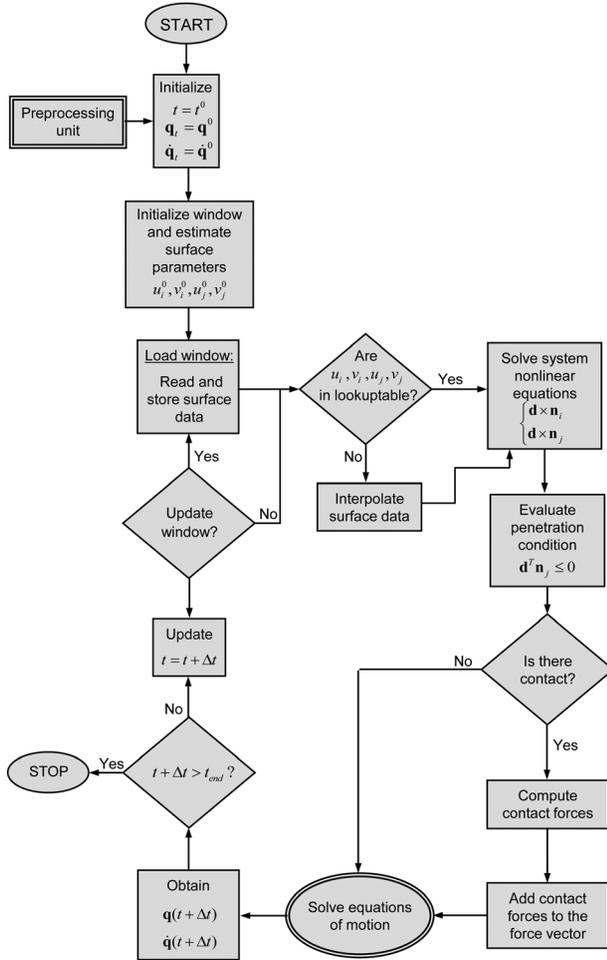
where  $c_r$  represents the classical restitution coefficient and  $\dot{\delta}^{(-)}$  is the initial relative normal contact velocity, the remaining parameters having the same meaning as presented above.

## 5 Computational Algorithm for Contact in Multibody Dynamics

Figure 8 presents the flow chart of the computational algorithm of the multibody approach proposed here for analysis of contact problems with 3D freeform surfaces [38].

The algorithm can be summarized by the following steps:

- (1) Run the preprocessing unit, which corresponds to the surface generation and preparation described in Sec. 2.
- (2) Set the initial conditions of the system: initial time  $t^0$ , initial positions  $\mathbf{q}^0$ , initial velocities  $\dot{\mathbf{q}}^0$ , storage window dimensions, and initial guesses for surface contact point  $u_i^0, v_i^0, u_j^0, v_j^0$ .
- (3) Load window: Read and store surface data.
- (4) Check if the surface point  $u_i^0, v_i^0, u_j^0, v_j^0$  belongs to the lookup table; if it does not, a bilinear interpolation is performed to determine its geometric properties.
- (5) Solve the system of nonlinear equations (10) and (11) that expresses the geometric conditions that a surface point has to fulfill to be considered a candidate contact point. The detailed procedure to find the solution of Eqs. (10) and (11) is described in Ref. [26].
- (6) Store the parametric coordinates of the candidate contact points ( $u'_i, v'_i, u'_j, v'_j$ ) as initial guesses ( $u_i^0, v_i^0, u_j^0, v_j^0$ ) of the next time step.
- (7) Evaluate indentation condition (12) and check for contact; if there is contact, evaluate contact forces using a continuous contact force law.
- (8) Add contact forces to the vector of generalized forces.
- (9) Apply a multibody formulation in order to obtain the new generalized positions and velocities of the system for time



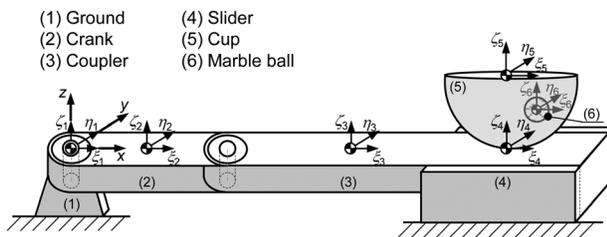
**Fig. 8 Computational algorithm proposed to deal with 3D contact problems in multibody systems**

step  $t + \Delta t$ . In the present work, an explicit algorithm of the Gear type with varying order and time step is used [40].

- (10) Increment the system time variable.
- (11) Check if the storage window needs to be updated. If so, go to step (3).
- (12) Go to step (4) and proceed with the whole process for the new time step until the final time of the analysis is reached.

## 6 Demonstrative Example of Application

The multibody system (MBS) used here, as a demonstrative application, is illustrated in Fig. 9. This model comprises a slider-crank mechanism with a cup on the top of the slider and a marble ball. The initial positions and mass of the six bodies that comprise the model are listed in Table 2. The



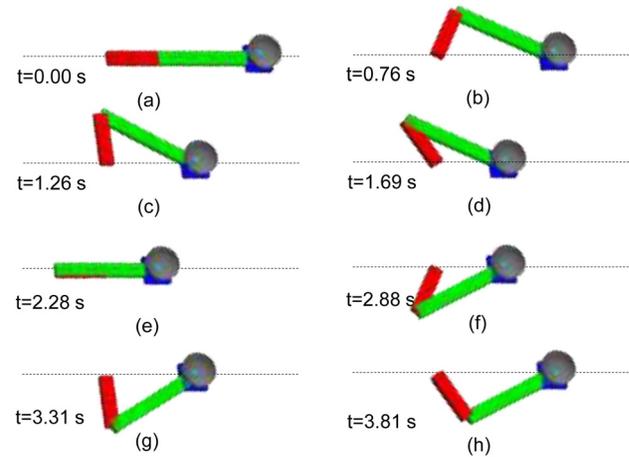
**Fig. 9 Initial configuration of a slider-crank mechanism with a cup (body 5) on the top of the slider and a marble ball (body 6)**

**Table 2 Initial position and mass of the bodies of the MBS model**

Body	x (mm)	y (mm)	z (mm)	Mass (kg)
Crank (2)	76.2	0.0	0.0	1.726
Coupler (3)	304.8	0.0	0.0	3.451
Slider (4)	457.2	0.0	0.0	1.726
Cup (5)	457.2	0.0	50.0	0.016
Marble ball (6)	457.2	5.6	12.0	0.011

**Table 3 Geometric and material properties of the cup and the ball**

Properties	Cup	Ball
Radius (mm)	50	10
$uv$ -points	999 × 999	
Material	steel	glass
Stiffness (GPa)	5.0181	
Coefficient of restitution	0.6275	



**Fig. 10 Illustration of the resultant motion of the MBS in study**

slider-crank mechanism is initialized with a crank angular velocity equal to 2 rad/s. At the start of the dynamic analysis, the crank and the coupler are aligned in the  $x$ -direction, corresponding to the top dead point.

A contact pair is considered between the cup and the marble ball. This ball is released from the initial position under the action of gravity only, which is taken to act in the negative  $z$ -direction. The geometric and material properties of the cup and the marble ball are summarized in Table 3. The resultant motion of the multibody system is illustrated in Fig. 10.

Two contact force models are considered in the analysis carried hereafter, namely the elastic Hertz law and the dissipative model of Flores et al. Figures 11 and 12 show the  $xy$ -trajectory and the  $xz$ -trajectory of the contact points of the marble ball when both contact force models are applied.

The indentation of the marble ball resultant of the application of the two contact force approaches is depicted in Fig. 13. Figure 14 plots the force-indentation relations.

The ball falls down until it collides with the cup, which is considered to be rigid. When the ball collides with the cup, a contact takes place and the ball rebounds, producing jumps. This first contact occurs at the same time in both simulations. Nonetheless, the duration and amplitude of the jumps depend on the nature of the applied force model, as can be seen in Fig. 13. Three rebounds are reported during 0.1 s when the purely elastic Hertz is utilized.

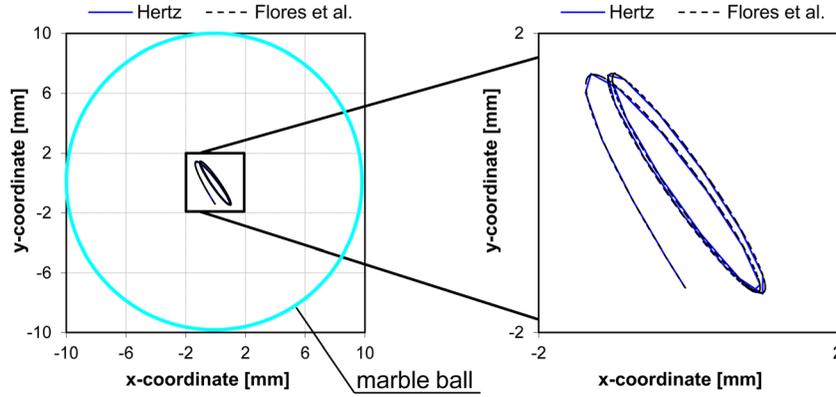


Fig. 11  $xy$ -trajectory of the contact points of the marble ball when using the Hertz law and the Flores et al. model

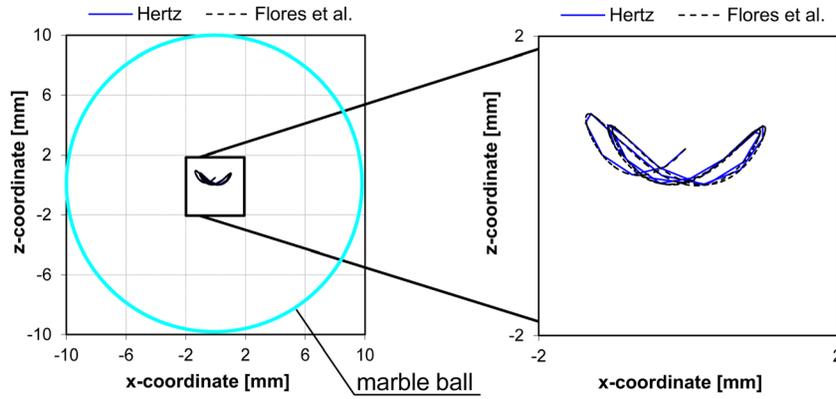


Fig. 12  $xz$ -trajectory of the contact points of the marble ball when using the Hertz law and the Flores et al. model

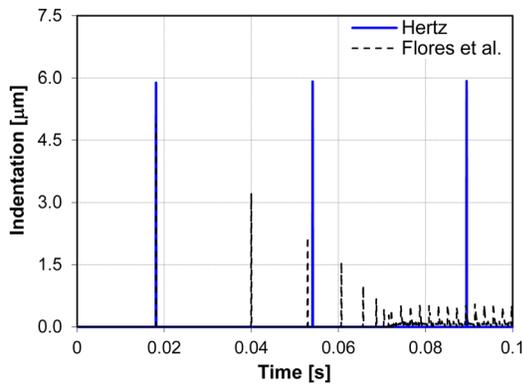


Fig. 13 Contact indentation between the marble ball and the cup during the first 0.1 s of simulation

These rebounds produce three jumps with approximately the same amplitude and duration. This is due to the fact that the Hertz law does not take into consideration the energy loss due to internal damping. In contrast to the Hertz law, the Flores et al. model accounts for the energy dissipation during impact and, therefore, the producing jumps have subsequently short duration and amplitude. As a consequence, many more jumps result by using the Flores et al. force approach. The area of the hysteresis loop represented in Fig. 14 corresponds to the amount of energy dissipated in the impact, which decreases alongside with the duration and amplitude of the jumps. Figure 15 depicts the trajectory of the ball center and the relative contact points of the marble ball during 0.1 s of simulation. By observing Fig. 15(a), it can be drawn that

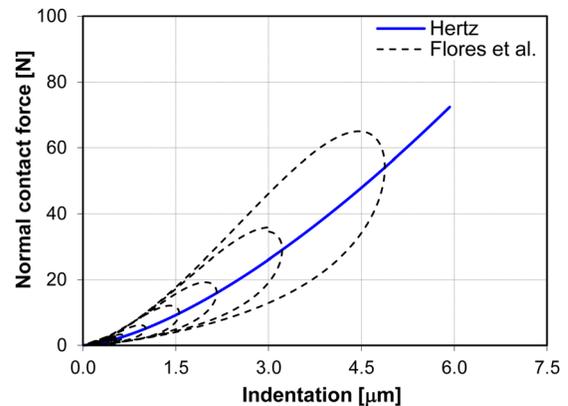
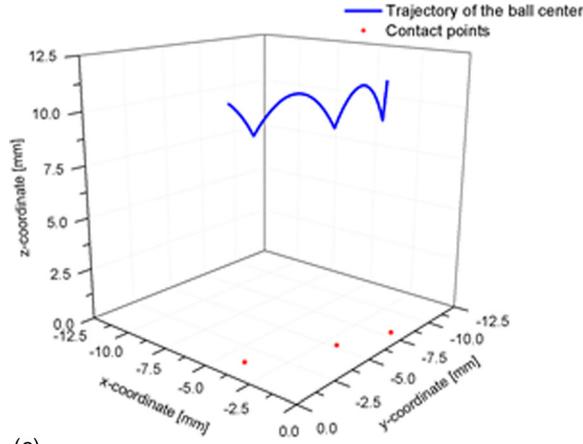


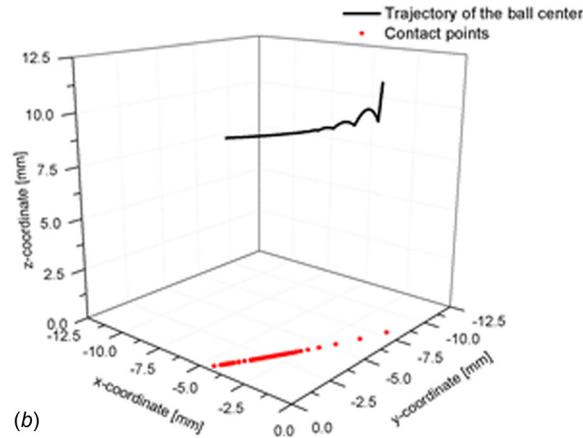
Fig. 14 Force-indentation relations of the contact between the marble ball and the cup during the first 0.1 s of simulation

when the Hertz law is applied, the amplitude of the jumps is approximately the same. In turn, when the Flores et al. model is used, the amplitude of the jumps decreases alongside the simulation until it reaches a very small value. At this time, the amplitude of the jumps stabilizes and keeps nearly constant, as depicted in Fig. 15(b). Despite the energy dissipated in each contact, the vertical velocity of the ball does not become null due to the motion of the slider crank, which provides energy to the ball. Indeed, for the time interval under analysis, the ball does not stop because of the slider motion, which acts as a driver.

In order to infer the influence of the size of the storage window on the computational time, two simulations are performed. In the



(a)



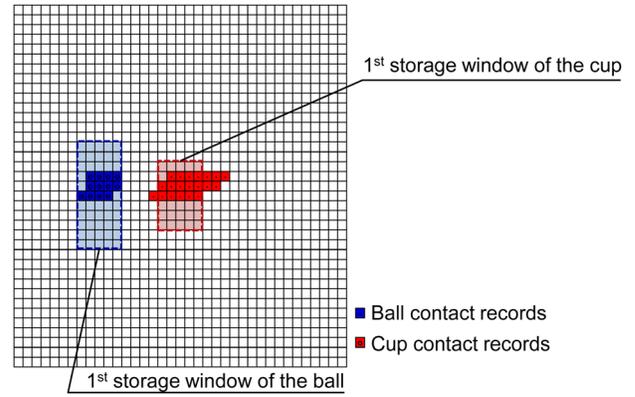
(b)

**Fig. 15 Trajectory of the ball center and the relative contact points during 0.1s of simulation using two contact force approaches: (a) Hertz law; (b) Flores et al. model**

first simulation, the size of the storage windows is the complete size of the surface files. In the second simulation, storage windows with reduced sizes are considered. The size of the storage windows and the central processing unit (CPU) time are listed in Table 4. Note that the reported CPU time concerns the actual dynamic analysis and includes any eventual storage window updating. As stated before, it does not include data preparation, in general, or lookup table loading, in particular, as such tasks are only done once during the preprocessing stage. The storage windows are illustrated in Fig. 16, as well as the contact records. Figure 16 represents the discretization of an eighth of a sphere. In a broad sense, the results show that the use of storage windows reduces the time of computation, even when window updates are required.

**Table 4 Window sizes and CPU time**

Simulation	1	2
Size of the window for the ball ( $u \times v$ )	$37 \times 37$	$5 \times 11$
Size of the window for the cup ( $u \times v$ )	$37 \times 37$	$5 \times 7$
Number of window updates for the ball	0	7
Number of window updates for the cup	0	50
Simulation time (s)	5	5
Time step (s)	$10^{-6}$	$10^{-6}$
CPU time (s)	1074	801



**Fig. 16 Representation of the storage windows**

## 7 Conclusions

A three-step methodology to deal with 3D contact problems within multibody systems was proposed. The first step concerns the generation and representation of the contact surfaces. This task relies on parametric functions to define the contact geometries, which allow the reduction of a 3D problem to a 2D space and comprises a set of suitable properties for geometric modeling. A modeling strategy to reduce the CPU time was outlined. This strategy consisted of developing a preprocessing technique to prepare contact surfaces for dynamic contact simulations and only keeping in memory the part of the surface geometric information relevant to the actual state of the contact.

A mathematical formulation for contact detection based on the common-normal concept was presented. This method states that two points in space belonging to different surfaces are candidate contact points if the normal vectors at these points are collinear to each other and perpendicular to the tangential vectors. This evaluation requires the resolution of a system of nonlinear equations. To check if a pair of points is an actual contact pair, an indentation condition has also to be verified. When contact is detected, a continuous contact force law is applied to evaluate the contact forces.

To verify the accuracy and efficiency of the proposed contact approach, computational simulations were performed using as an application example a slider-crank mechanism with a cup on the top of the slider and a falling marble ball. The first set of simulations was carried out to analyze the dynamics of the overall system and to investigate the influence of the contact force model on the contact response. Within this set of simulations, a typical kinematic behavior of the slider-crank mechanism was observed.

The application of the proposed preprocessor unit allows for reading only the surface portion in the vicinity of the contact area, which is called a storage window. This approach aims to avoid the need of reading all surface data to the CPU memory and, hence, to decrease the computational time. The storage window is suitable for contact point searching, in particular when the surface file is very large and the contact point moves slightly and smoothly in the same area. In order to check the advantage of using a storage window, a second set of simulations was performed. The results depict that the use of the storage window speeds up the simulation. Although it is expected that for a larger number of bodies and wider contact surfaces the benefits of the proposed approach becomes evident, only a clear demonstration with more complex surfaces and contact conditions can clarify the full potential of the approach proposed.

## Acknowledgment

The authors acknowledge the support given by Portuguese Foundation for Science and Technology (FCT) under the project BIOJOINTS (PTDC/EME-PME/099764/2008). The first author thanks FCT for the grant SFRH/BD/40164/2007.

## References

- [1] Pfeiffer, F., and Glocker, C., 1996, *Multibody Dynamics With Unilateral Contacts*, Wiley, New York.
- [2] Gilardi, G., and Sharf, I., 2002, "Literature Survey of Contact Dynamics Modeling," *Mech. Mach. Theory*, **37**(10), pp. 1213–1239.
- [3] Pedersen, S. L., Hansen, J. M., and Ambrósio, J., 2004, "A Roller Chain Drive Model Including Contact With Guide-Bars," *Multibody Sys. Dyn.*, **12**(3), pp. 285–301.
- [4] Ambrósio, J., 1996, "Dynamics of Structures Undergoing Gross Motion and Nonlinear Deformations: A Multibody Approach," *Comput. Struct.*, **59**(6), pp. 1001–1012.
- [5] Goldsmith, W., 1960, *Impact: The Theory and Physical Behaviour of Colliding Solids*, Edward Arnold, Sevenoaks, UK.
- [6] Brach, R. M., 1991, *Mechanical Impact Dynamics, Rigid Body Collisions*, Wiley, New York.
- [7] Johnson, K. L., 1999, *Contact Mechanics*, Cambridge University Press, Cambridge, UK.
- [8] Stronge, W. J., 2000, *Impact Mechanics*, Cambridge University Press, Cambridge, UK.
- [9] Wriggers, P., 2006, *Computational Contact Mechanics*, 2nd ed., Springer-Verlag, Berlin.
- [10] Machado, M., Moreira, P., Flores, P., and Lankarani, H. M., 2012, "Compliant Contact Force Models in Multibody Dynamics: Evolution of the Hertz Contact Theory," *Mech. Mach. Theory*, **53**, pp. 99–121.
- [11] Flores, P., and Ambrósio, J., 2010, "On the Contact Detection for Contact-Impact Analysis in Multibody Systems," *Multibody Sys. Dyn.*, **24**(1), pp. 103–122.
- [12] Machado, M., Flores, P., Pimenta Claro, J. C., Ambrósio, J., Silva, M., Completo, A., and Lankarani, H., 2010, "Development of a Planar Multibody Model of the Human Knee Joint," *Nonlin. Dyn.*, **60**(3), pp. 459–478.
- [13] Ambrósio, J., and Verissimo, P., 2009, "Improved Bushing Models for Vehicle Dynamics," *Multibody Sys. Dyn.*, **22**(4), pp. 341–365.
- [14] Flores, P., Ambrósio, J., Claro, J. C. P., Lankarani, H. M., and Koshy, C. S., 2009, "Lubricated Revolute Joints in Rigid Multibody Systems," *Nonlin. Dyn.*, **56**, pp. 277–295.
- [15] Hahn, C., and Wriggers, P., 2003, "An Explicit Multi-Body Contact Algorithm," *Proc. Appl. Math. Mech.*, **3**, pp. 280–281.
- [16] Hippmann, G., 2004, "An Algorithm for Compliant Contact Between Complexly Shaped Bodies," *Multibody Sys. Dyn.*, **12**(4), pp. 345–362.
- [17] Wellmann, C., Lillie, C., and Wriggers, P., 2008, "A Contact Detection Algorithm for Superellipsoids Based on the Common-Normal Concept," *Eng. Computat.*, **25**(5), pp. 432–442.
- [18] Lopes, D. S., Silva, M. T., Ambrósio, J., and Flores, P., 2010, "A Mathematical Framework for Rigid Contact Detection Between Quadric and Superquadric Surfaces," *Multibody Sys. Dyn.*, **24**(3), pp. 255–280.
- [19] Delp, S. L., and Loan, J. P., 1995, "A Graphics-Based Software System to Develop and Analyze Models of Musculoskeletal Structures," *Comp. Biol. Med.*, **25**(1), pp. 21–34.
- [20] Shuxian, Z., Wanhua, Z., and Bingheng, L., 2005, "3D Reconstruction of the Structure of a Residual Limb for Customising the Design of a Prosthetic Socket," *Med. Eng. Phys.*, **27**(1), pp. 67–74.
- [21] Silva, P., Silva, M. T., and Martins, J. M., 2010, "Evaluation of the Contact Forces Developed in the Lower Limb/Orthosis Interface for Comfort Design," *Multibody Sys. Dyn.*, **24**(3), pp. 367–388.
- [22] Dopico, D., Luaces, A., Gonzalez, M., and Cuadrado, J., 2010, "Dealing With Multiple Contacts in a Human-in-the-Loop-Application," *Multibody Sys. Dyn.*, **25**(2), pp. 167–183.
- [23] Choi, J., Ryu, H. S., Kim, C. W., and Choi, J. H., 2010, "An Efficient and Robust Contact Algorithm for a Compliant Contact Force Model Between Bodies of Complex Geometry," *Multibody Sys. Dyn.*, **23**(1), pp. 99–120.
- [24] Lin, M., and Gottschalk, S., 1998, "Collision Detection Between Geometric Models: A Survey," Proceedings of IMA Conference on Mathematics of Surfaces, San Diego, CA, pp. 37–56.
- [25] Tasora, A., Negrut, D., and Anitescu, M., 2008, "Large-Scale Parallel Multibody Dynamics With Frictional Contact on the Graphical Processing Unit," *Proc. IMechE K J. Multibody Dyn.*, **222**(4), pp. 315–326.
- [26] Pombo, J., Ambrósio, J., and Silva, M., 2007, "A New Wheel-Rail Contact Model for Railway Dynamics," *Veh. Sys. Dyn.*, **45**(2), pp. 165–189.
- [27] Bei, Y., and Fregly, B. J., 2004, "Multibody Dynamic Simulation of Knee Contact Mechanics," *Med. Eng. Phys.*, **26**(9), pp. 777–789.
- [28] Pombo, J., and Ambrósio, J., 2003, "General Spatial Curve Joint for Rail Guided Vehicles Kinematics and Dynamics," *Multibody Sys. Dyn.*, **9**(3), pp. 237–264.
- [29] Ericson, C., 2005, *Real-Time Collision Detection*, Morgan Kaufmann Publishers, San Francisco, CA.
- [30] Pauly, M., Keiser, R., Kobbelt, L. P., and Gross M., 2003, "Shape Modeling With Point-Sampled Geometry," *ACM Trans. Graphics*, **22**(3), pp. 641–650.
- [31] Farin, G. E., Hoschek, J., and Kim, M.-S., 2002, *Handbook of Computer Aided Geometric Design*, North-Holland, Amsterdam.
- [32] Campbell, R. J., and Flynn, P. J., 2001, "A Survey of Free-Form Object Representation and Recognition Techniques," *Comp. Vis. Image. Understand.*, **81**(2), pp. 166–210.
- [33] Lai, J.-Y., and Ueng, W.-D., 2001, "G2 Continuity for Multiple Surfaces Fitting," *Int. J. Adv. Manf. Tech.*, **17**(8), pp. 575–585.
- [34] Hohmeyer, M., and Barsky, B. A., 1989, "Rational Continuity: Parametric, Geometric, and Frenet Frame Continuity of Rational Curves," *ACM Trans. Graphics*, **8**(4), pp. 335–359.
- [35] Wurm, E., Juttler, B., and Kim, M.-S., 2005, "Approximate Parametrization of Implicitly Defined Surfaces," Mathematics of Surfaces XI, R. Martin, H. Bez, M. Sabin, eds., Springer, Berlin, Germany, pp. 434–447.
- [36] Hartmann, E., 2000, "Numerical Parametrization of Curves and Surfaces," *CAD Design*, **17**, pp. 251–266.
- [37] Pombo, J., and Ambrósio, J., 2008, "Application of a Wheel-Rail Contact Model to Railway Dynamics in Small Radius Curved Tracks," *Multibody Sys. Dyn.*, **19**(1–2), pp.91–114.
- [38] Nikravesh, P., 1988, *Computer-Aided Analysis of Mechanical Systems*, Prentice Hall, Englewood Cliffs, NJ.
- [39] Gribbon, K., and Bailey, D., 2004, "A Novel Approach to Real-Time Bilinear Interpolation," 2nd IEEE International Workshop on Electronic Design, Test and Applications, Perth, Australia, pp. 126–131.
- [40] Gear, C. W., 1971, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, NJ.