A Local Contact Detection Technique for Very Large Contact and Self-Contact Problems: Sequential and Parallel Implementations

V.A. Yastrebov, G. Cailletaud and F. Feyel

Abstract The local contact detection step can be very time consuming for large contact problems reaching the order of time required for their resolution. At the same time, even the most time consuming technique all-to-all does not guarantee the correct establishment of contact elements needed for further contact problem resolution. Nowadays the limits on mesh size in the Finite Element Analysis are largely extended by powerful parallelization methods and affordable parallel computers. In the light of such changes an improvement of existing contact detection techniques is necessary. The aim of our contribution is to elaborate a very general, simple and fast method for sequential and parallel detection for contact problems with known a priori and unknown master-slave discretizations. In the proposed method the strong connections between the FE mesh, the maximal detection distance and the optimal dimension of detection cells are established. Two approaches to parallel treatment of contact problems are developed and compared: SDMR/MDMR – Single/Multiple Detection, Multiple Resolution. Both approaches have been successfully applied to very large contact problems with more than 2 million nodes in contact.

1 Introduction

By local contact detection we mean a procedure which detects elements (nodes, surfaces) of one part of a finite element mesh which potentially come in contact with another part of the mesh on a current computational step. In the context of the node-to-segment discretization (NTS) [15] the contact detection is an establishment of the

V.A. Yastrebov · G. Cailletaud
Centre des Matériaux, Mines ParisTech, CNRS UMR 7633, BP 87, 10 rue Henri Desbruères, 91003 Evry, France; e-mail: {vladislav.yastrebov, georges.cailletaud}@mines-paristech.fr

F. Feyel
ONERA, BP 72, 29 avenue de la Division Leclerc, 92322, Chatillon, France;
e-mail: frederic.feyel@onera.fr

G. Zavarise & P. Wriggers (Eds.): Trends in Computational Contact Mechanics, LNACM 58, pp. 227–251.
springerlink.com

closest opposing master segment for each node of the slave surface which will potentially come in contact. Further these nodes and the corresponding surfaces form abstract contact elements which contribute consequently to the weak form associated with the finite element problem. Therefore, wrong contact detection results in incorrect solution or even in its failure.

It should be distinguished two contact search phases [11]: spatial search and contact detection. The first notion is used for search between separate solids coming in contact, i.e. rather between separate geometries than discretizations. Contact spatial search methods are of big importance in multibody systems and the discrete element method where interaction between more or less identical particles such as crashed stone, sand, snow is considered to analyse mud flows, opencast mines, avalanches, etc. It is worth mentioning that previously the particular attention of the scientific community has been paid mostly to this phase of contact search, because the local discretization of solids remained rather moderate and the bucket detection method [1] in its very general form or even the simplest all-to-all approach remained rather efficient and fast enough techniques; especially in case of small slip when only one execution of the detection procedure is required. Recent progress in parallel computing makes possible extremely large implicit and explicit contact simulations between just a few but very finely meshed solids. The phase of local contact detection becomes crucial and time consuming part of the computational process, especially in case of finite slip and large deformation.

The detection phase consists in establishment of contact elements which by-turn in NTS discretization consist of one slave node and a master surface of another element. The simplest and straightforward detection method is all-to-all: each master segment is checked for proximity to each slave node. The growth rate of the method is $O(N_s \times N_m)$, where N_s and N_m are numbers of slave nodes and master segments respectively. If one considers second order master surface each check of projection requires the solution of nonlinear equation which takes several iterations. For example, let us estimate the time needed to perform the simplest contact detection procedure within two surfaces consisting of 1024×1024 elements each. If each search of projection requires 5 iteration and the computer performs the detection with 10^6 flops, then the detection will be achieved in more than two months(!). This time surpasses considerably the time needed to a parallel resolution of a FE problem possibly associated with such a FE mesh.

The first simplification of the detection is to start from node-to-node detection in stead of node-to-segment. For further improvement let us imagine a set of spatially distributed points. The problem is to detect for a given point the closest one from this set. Human vision accomplishes this task easily analyzing just few close points. It does not need any analysis of the whole point set while the simple detection algorithm does, because it is "blind" and needs to "touch" all the points one by one and compare distances between them. The techniques which have been worked out for contact detection are aimed at the reduction of quantity of points to "touch". Among these methods there are the bucket method [1,4], the heapsort and the octree method [10] and others. The two last algorithms have been developed mainly for the multibody simulations, i.e. for spatial contact search, however they can be adapted for the local contact detection as well. But these methods lack for generality (self-contact, parallel detection) and more elaborated analysis (optimal parameters), moreover several of them require facilities to handle tree data structures. Recently a new powerfull detection algorithm [13] has been proposed. It accounts the links between nodes and that is why it is more adapted for local contact detection. The method is based on the bounding volume trees associated with sets of segments. However it is designed for mortar based contact formulations and apparently for the NTS discretization its rapidity is inferior to the method proposed here.

Even if a fast enough node-to-node detection procedure is worked out, many problems remain, such as optimal bounding box construction, challenging detection of nodes in blind spots and as a particular case – detection of passing by nodes. These difficulties appear from the fact that the finite element discretization of contacting surfaces is continuous but not smooth. Finally it can be affirmed that the robustness of the detection and its rapidity depends strongly on the way the mentioned difficulties are overcame as well as on the carefulness of coding.

In this contribution the bucket or grid detection method [1] will be improved and adapted for a very general case. Its sequential and parallel implementations will be discussed in detail for very large finite element simulations in the framework of the node-to-segment discretization.

First, the principal notions are introduced and all the steps of the method are considered in details. Optimal detection parameters (proximity criterion, detection distance and cell size) are derived based on the numerous large scale detection tests. Efficient procedures for bounding box construction, neighbouring cell detection and verification of “passing by” nodes are proposed. The performance of the method is demonstrated on several extremely large contact problems containing up to 2 million nodes in contact. Further the method is extended for a very general case of unknown a priori master-slave contact surfaces, which is of a great importance for self-contact treatment. The last part is devoted to the detection phase in case of parallel treatment of contact problems, different approaches are considered.

2 Method Description

The grid (bucket) detection method [1,4] is natural and simple. But its implementation and choice of the internal parameters should be discussed in more details. First, a short description of the method is given, then each stage of the procedure is discussed in detail, the optimality of parameters is analyzed and finally some numerical examples both artificial (contact between two curved surfaces) and real engineering (tyre on road contact) are given to demonstrate the performance of the method. The ultimate aim is an improvement of the grid method and determination of detection parameters which would reduce the required CPU time.

First of all the master-slave approach and the associated node-to-segment (NTS) discretization have to be shortly explained. Two contacting surfaces in the master-slave approach are distinguished: one is called slave or impactor and the other is called master or target. For clarity the slave-master notation will be used. Such distinction comes both from the node-to-segment discretization and from geometrical description of contact, precisely from the asymmetry of the closest distance definition between contacting surfaces. The slave surface manages nodes of the first surface and neglects their connections, i.e. all the interpolations between slave nodes are not taken into account. The master manages segments of the second contacting surface and its description is closely connected with the order of the elements and consequently with the interpolation functions. It is worth mentioning that the use of nonlinear interpolations with many discretization techniques (including NTS which is considering here) leads to incorrect results of contact analysis (see e.g. [11]).

"Contact element" (here NTS contact element) is an abstract (not structural) element consisting of a slave node and several master nodes united by a master surface segment. All the geometrical quantities such as normal gap g_n and tangential velocity $\dot{\mathbf{g}}_t$ are evaluated in the master reference frame, i.e. the geometry of contact is described by an interaction of a slave node with a master surface segment. Such contact elements take care of the local contact interaction between two bodies in the resolution phase and hence they have to be created before the slave node slides on or penetrates the master surface. Thus the slave nodes which are close enough to the master surface have to be detected and included in consideration before the resolution step.

Before discussing particular details let us derive a short description of the grid detection method. Two phases can be distinguished: preliminary phase and detection phase. In the preliminary phase the optimal size of the grid is evaluated, further a potential contact area is determined and divided with an enumerated regular grid. That allows to reduce locally the area of closest nodes search. Finally all slave and master nodes situated in the detection area are distributed in the cells of the grid. In the detection phase for each slave node we check for the closest master node in the current cell. And if necessary we check one or several neighbouring cells for possible proximal master nodes. As the closest master node is found the existence of the slave node projection onto all segments attached to the master node is verified. If at least one projection exists then a contact element will be established otherwise the verification if the node is in a blind spot or is a "passing by node" is needed.

2.1 Preliminary Stage of Contact Detection

First of all a key parameter for the contact detection procedure has to be introduced – maximal detection distance $d_{\max}$. In case of node-to-segment detection $d_{\max}$ determines the following: if a slave node is closer to the master surface than $d_{\max}$, then it is supposed that this node can come in contact during the following time step, otherwise not. The method considered here is based on the node-to-node detection, so the meaning of the maximal detection distance is different. If the distance between two nodes $d_{ij} = \mathrm{dist}(\mathbf{r}_i, \mathbf{r}_j)$ is smaller than the maximal detection distance, then the corresponding slave node $\mathbf{r}_i$ and one of master surfaces containing the mentioned

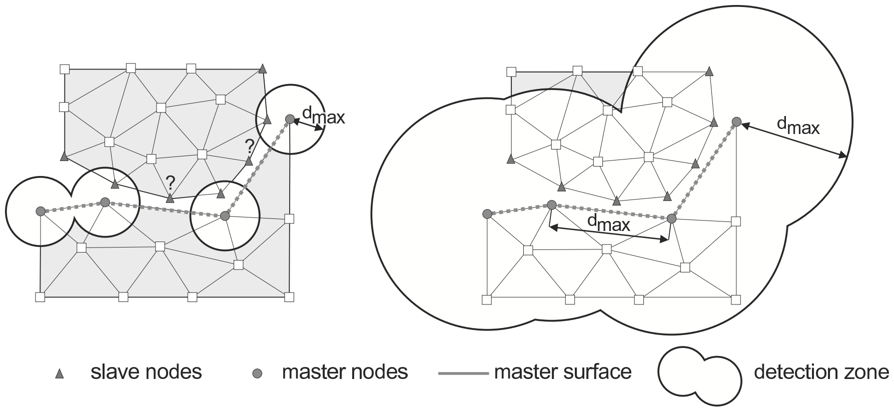

Fig. 1 Maximal detection distance $d_{\max}$: on the left not correct on the right correct choice.

node $\mathbf{r}_j$ are considered to be potentially in contact during the following time step otherwise not. This difference naturally results in a limitation on the minimal value of the $d_{\max}$. Here the $\operatorname{dist}(\mathbf{r}_i, \mathbf{r}_j)$ denotes Euclidian metric in the global reference frame $\operatorname{dist}(\mathbf{r}_i, \mathbf{r}_j) = |\mathbf{r}_i - \mathbf{r}_j|$. The value of the key parameter $d_{\max}$ for the detection procedure can be determined automatically accordingly to the discretization of the master or self-contact surface, to the loading and deformation rate. First, it should be mentioned that the maximal detection distance in proposed method has to be unique for the entire contact area and greater than one half of the maximal distance between master nodes attached to one segment

$$d_{\max} > \frac{1}{2} \max_{i=1,\, j=1,\, k=j+1}^{i=N_m,\, j=N_n^i-1,\, k=N_n^i} \operatorname{dist}(\mathbf{r}_j^i, \mathbf{r}_k^i), \tag{1}$$

where N_m is a total number of master segments, N_n^i is a total number of master nodes attached to the i-th master segment, $\mathbf{r}_j^i$ is a coordinate of the j-th node of the i-th master segment. If the condition (1) is not fulfilled, then some slave nodes coming in contact with master surface can be lost (see Figure 1, here and further for the sake of simplicity and clarity all figures represent two dimensional cases but can be easily extended to three dimensions).

For a reasonable number of time steps for geometrically or physically nonlinear problem the maximal detection distance can be determined as dimension of the biggest master segment, i.e. accordingly to the discretization of the geometry

$$d_{\max} = \max_{i=1,\, j=1,\, k=j+1}^{i=N_m,\, j=N_n^i-1,\, k=N_n^i} |r_j^i - r_k^i|. \tag{2}$$

Such estimation is reasonable in case of a regular discretization of the master surface. On the other hand if the distribution of the master nodes is very heterogeneous, i.e. fine surface mesh in one contact region and rough in another, the value of $d_{\max}$ appears to be highly overestimated for certain regions. This fact decreases the efficiency of the method, but in general for an adequate finite element mesh the increase of the detection time is not so high. The influence of the maximal detection distance on detection time will be discussed later.

In the case of linearly elastic material and frictionless contact, the geometry can change significantly during one time step. So the analysis of discretization can give only lower bound for $d_{\max}$ and that is why its value should be augmented manually or automatically accordingly to the deformation and/or displacement rate, for example in the following way:

$$d_{\max} = \max\left\{ \max_{i=1,\, j=1,\, k=j+1}^{i=N_m,\, j=N_n^i-1,\, k=N_n^i} |\mathbf{r}_j^i - \mathbf{r}_k^i|;\ 2\max_{i=1}^{N_c} |\Delta \mathbf{r}_i| \right\}, \tag{3}$$

where N_c is a total number of slave and master nodes and $\Delta \mathbf{r}_i$ is an estimation of the maximal displacement of the i-th node, 2 takes care of possible opposite translations of master and slave nodes. In case of remeshing or sufficiently large deformations of the master, the detection parameter $d_{\max}$ should be recomputed at each remeshing or at each N-th time step.

Before carrying out any detection the spatial area where contact can take place during the following time step has to be chosen. It has to contain as few master and slave nodes as possible but on the other hand it has to include all the nodes potentially coming in contact on the following step. If needed this area has to be frequently updated. We propose to confine this area by a bounding box (parallelepiped) defined in the global reference frame.

The determination of the bounding box differs for known a priori and unknown master-slave discretizations. In case of unknown master-slave the bounding box should include all possible contacting surfaces. But frequently the discretization is known a priori even if contact occurs within one body (self contact). In this case the construction of an optimal bounding box allows to exclude from consideration some nodes which cannot come in contact during the following time step (Figure 2) and consequently it results in acceleration of the detection procedure. It is worth mentioning that here the very general case is considered: any slave node can potentially come in contact with any master segment during the loading. Often it is not the case and for each slave node the set of possible master segments is limited and partly predefined. But in order to take it into account the detection technique should be tuned for each particular case. The consideration of such techniques is out of the scope of this contribution.

First of all the dimensions of master and slave surfaces are estimated. Note than even if the master surface consists of several independent zones in the grid detection method it can be considered as one set of master nodes with associated segments. It is proposed to construct two independent bounding boxes $B_s : \{\mathbf{r}_s^1, \mathbf{r}_s^2\}$ and $B_m : \{\mathbf{r}_m^1, \mathbf{r}_m^2\}$ containing all slave and master nodes respectively, where $\mathbf{r}^1$ and $\mathbf{r}^2$ are the vectors in the global reference frame of two opposite corners determining the bounding boxes. Note that each bounding box confining master and slave nodes includes also a node free margin zone of the size of maximal detection distance at each side.

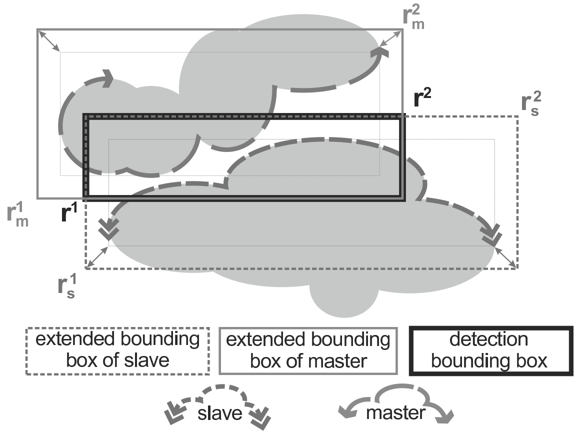

Fig. 2 Determination of the bounding box for the contact detection procedure in case of simple master-slave contact.

$$
\begin{aligned}
\mathbf{r}^1 : r^1_{\{x,y,z\}} &= \min_{i=1}^{N_b}\{\mathbf{e}_{\{x,y,z\}} \cdot \mathbf{r}_i\} - d_{\max}\mathbf{e}_{\{x,y,z\}}, \\
\mathbf{r}^2 : r^2_{\{x,y,z\}} &= \max_{i=1}^{N_b}\{\mathbf{e}_{\{x,y,z\}} \cdot \mathbf{r}_i\} + d_{\max}\mathbf{e}_{\{x,y,z\}},
\end{aligned}
\tag{4}
$$

where N_b is a number and $\mathbf{r}_i$ is a vector of nodes to be included in the bounding box and $\mathbf{e}_{\{x,y,z\}}$ are orthonormal basis vectors in the global reference frame. The margin of $\pm d_{\max}$ is introduced to avoid any loss of possible contact elements. Some improvements can be introduced in order to reduce the time of bounding box construction. The user can precise that one or several contact surfaces are rigid and do not move, then permanent bounding boxes can be assigned to these surfaces and there is no need to update them. Another possible feature is the prediction by the user that the deformation and displacement of a contact surface is connected to the displacement of certain nodes. It allows to avoid the verification of all nodes in (4). Since the nodal coordinates are kept in memory in the global reference frame it is much more faster to work directly with these coordinates so no rotation to the bounding boxes must be applied. The resultant bounding box $B : \{\mathbf{r}^1, \mathbf{r}^2\}$ is taken as the intersection of master and slave bounding boxes $B = B^m \bigcap B^s$. The practice shows that a further contraction of the bounding box does not reduce significantly the detection time.

When a bounding box is determined, an internal grid should be constructed in a proper way. In the grid detection method this grid should be regular and the cell size d_c should be optimum: not too large in order to keep the number of slave and master nodes in the cell as small as possible and not too small at least greater than the maximal detection distance $d_c \geq d_{\max}$. For smaller cells, the determination of the neighbouring cells which should be investigated is not evident; moreover, the growth rate of their maximal number N_c is cubical

$$
\text{if } d_c = \frac{d_{\max}}{n}, n > 1 \Rightarrow N_c = (3+2n)^3 \tag{5}
$$

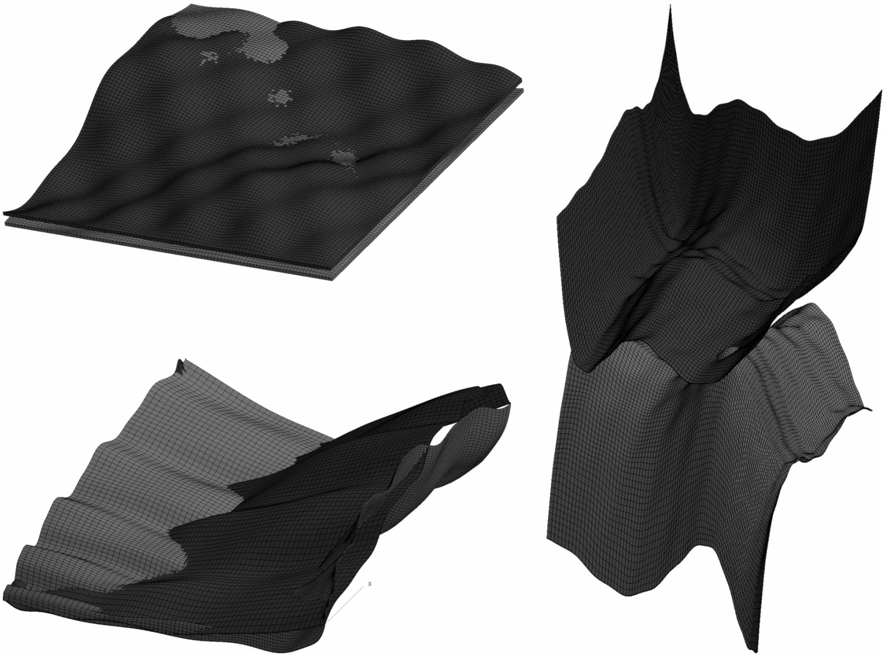

Fig. 3 Example of finite element meshes used to determine the optimal cell size. Proximal meshes with homogeneous (left top) and heterogeneous (left bottom) spatial node distribution and convex meshes (right).

The smaller the cell size, the higher the total number of cells and consequently the smaller the number of contact nodes per cell. But on the other hand small cell size increases the necessity to carry out the detection in neighbouring cells. It can be shown analytically by means of probability methods that for homogeneous node distribution both in 2D and 3D cases the minimal detection time is unique and corresponds to the minimal cell size. Such simple analysis predicts quadratic growth of the detection time in 2D case and cubic in 3D. To demonstrate it for real cases let us analyse the dependence of the detection CPU time t on the cell size d_c. Several finite element problems have been considered, each problem consists of two separate finite element meshes curved in a different way. Slave and master surfaces consist of over 10200 nodes. Three sets have been considered: proximal meshes with homogeneous (Figure 3, left top) and heterogeneous (Figure 3, left bottom) node distribution and convex mesh with heterogeneous node distribution (Figure 3, right). Each set is represented by 5 different realizations of curved surfaces. By homogeneous node distribution we mean that the maximal segment dimension does not exceed 200% of the minimal one, otherwise the node distribution is considered to be heterogeneous.

In Figure 4 the dependence of the average detection CPU time and the average number of investigated neighbouring cells on the normalized cell size $d_c/d_{\max}$ is

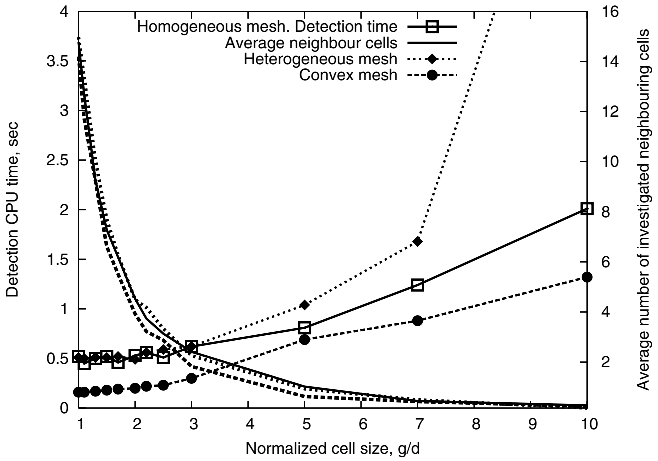

Fig. 4 The dependence of the detection time and the average number of neighbouring cell investigated during the detection on the normalized cell size.

represented for different sets. As shown in the figure, the detection time increases nonlinearly with higher slope for heterogeneous mesh than for homogeneous, because of initially higher maximal detection distance for such type of mesh. As expected, the detection time for convex meshes is smaller because of the smaller associated bounding boxes. Different discretizations (256 × 256, 512 × 512) have been tested and in all the cases the same dependence takes place. Accordingly to the analytical estimation and carried out tests, the optimal grid size is the minimal one, i.e. equal to the maximal detection distance

$$d_c = d_{\max}. \tag{6}$$

For such a choice, each grid cell contains the minimal number of nodes, but on the other hand it is necessary to carry out the detection procedure in many neighbouring cells: on average 12–16 cells (of 26 surrounding cells in 3D) (Figure 4).

When the maximal detection distance is determined and the bounding box is constructed, the internal grid has to be established in the bounding box and all the slave and master nodes have to be distributed in the cells of the grid. Since the optimal cell size is $d_{\max}$ the number of cells in each dimension of the grid is defined as

$$N_{x,y,z} = \max\left\{ \left[\frac{r^2_{x,y,z} - r^1_{x,y,z}}{d_{\max}} \right] ;\ 1 \right\}, \tag{7}$$

where $[x]$ stands for the integer part of x. Such choice of cell numbers provides the grid sizes Δx, Δy and Δz not smaller than the maximal detection distance in the case of $N > 1$

$$\Delta\{x,y,z\} = \frac{r^2_{x,y,z} - r^1_{x,y,z}}{N_{x,y,z}} \geq d_{\max}. \tag{8}$$

Each cell of the grid has to be enumerated, the unique integer number $N \in [0;\, N_x \times N_y \times N_z - 1]$ is given to each cell with spatial "coordinates" i_x, i_y and i_z, where $i_{x,y,z} \in [0;\, N_{x,y,z} - 1]$

$$N = i_x + i_y N_x + i_z N_x N_y. \tag{9}$$

Now the growth rate of the method can be estimated roughly as $O(N_s N_m / N_x N_y N_z)$. If number of master and slave nodes per cell is supposed to be constant $\rho = N/N_c$, where $N_c = N_x N_y N_z$ and N is an average number of master and slave nodes, then the growth rate of the method can be rewritten as $O(N)$. However, in practice the distribution of nodes is not homogeneous and this value appears to be underestimated.

Slave and master nodes situated in the bounding box have to be distributed in the cells. For this purpose two arrays A^s and A^m corresponding to slave and master nodes respectively are to be created. They contain slave and master node identification numbers (ID). For example element A^s_{ij} keeps the ID of the j-th slave node in the i-th cell of the grid, $i \in [0; N_x N_y N_z - 1]$, $j \in [0; N^s_i]$, N^s_i being the number of slave nodes in the i-th cell. In average the number of integer (32 bits) elements in array does not exceed the number of contact nodes and so even for extremely large problems it makes just a minor contribution in memory requirement. However, the arrays can be replaced by linked-list storages as in [4].

For each node with coordinates $\mathbf{r} : \{r_x, r_y, r_z\}$ inside the bounding box, the corresponding cell number is easily determined as

$$N_{\text{cell}} = \left[\frac{r_x - r_x^1}{\Delta x}\right] + \left[\frac{r_y - r_y^1}{\Delta y}\right] N_x + \left[\frac{r_z - r_z^1}{\Delta z}\right] N_x N_y. \tag{10}$$

2.2 Contact Detection

All steps described previously represent the preliminary part of the detection algorithm which demands in general 7–10% of the total detection time. The next steps correspond to the closest node detection, the determination of projections for the slave nodes onto the corresponding master segments and the establishment of contact elements. Let us discuss this stage in details.

For each grid cell c_i and for each slave node $\mathbf{r}^s_{ij}$ in this cell, i.e. for each node A^s_{ij} we look for the closest master node $\mathbf{r}^m_{ik}$ in the current cell, i.e. the closest node among A^m_{ik} if A^m_i is not empty. Among all master nodes in the cell the distance to the closest one is $d^s_{ij} \le d_{\max}$:

$$d^s_{ij} = \min\left\{ d_{\max}, \min_k \{ |r^s_{ij} - r^m_{ik}| \} \right\}. \tag{11}$$

It is obvious that master nodes situated in neighbouring cells (maximum 8 cells in 2D, 26 in 3D) have to be checked as well. Not all the cells are considered, but only those that are sufficiently close to the slave node. The criterion of the proximity is the

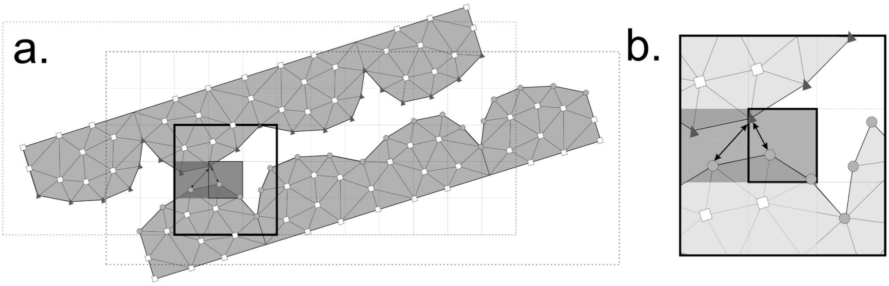

Fig. 5 Detection of the closest master node in current and neighbouring cell. Slave and master nodes represented by triangles and circles respectively.

following: if any boundary of the current cell (face, edge or vertex) is closer than the closest master node found up to the moment, i.e. closer than d^s_{ij}, then the detection procedure has to be carried out in neighbouring cells attached to this boundary one by one (Figure 5).

For example, let us consider a vertex of the i-th cell $\mathbf{r}^v_i$. For instance, after checking all master nodes in the current cell it was determined that the closest master node is situated at the distance of d^s_{ij} from the slave. If the considered slave node is closer to the vertex than this distance $\mathbf{r}^v_i$: $|\mathbf{r}^s_{ij} - \mathbf{r}^v_i| < d^s_{ij}$, then all master nodes in one of neighbouring cells attached to the vertex $\mathbf{r}^v_i$ have to be checked and consequently d^s_{ij} has to be decreased or kept the same (if no closer master node was found in this cell). And so on for other cells attached to this corner. In general the same procedure has to be performed for all 8 vertices, 12 edges and 6 faces of the i-th cell. To get more optimal algorithm such an investigation of neighbouring cells is better to start from the closest faces, further edges and finish the verification with vertices. Note that each verified cell may decrease the d^s_{ij} and consequently can decrease the number of cells to be checked. In such a manner all possibly proximal slave and master nodes are detected cell by cell. The average number of verified neighbouring cells for different meshes is represented in Figure 4. This number decreases with increasing normalized grid size $d_c/d_{\max}$ but as the optimal ratio $d_c/d_{\max} = 1$ the average number of verified neighbouring cells remains quite high (12–16 cells).

Every slave node in the bounding box has been considered and for certain of them $\mathbf{r}^{s*}_j$ corresponding proximal master nodes $\mathbf{r}^{m*}_j$ have been detected. To construct contact elements it is necessary to project each of such slave nodes $\mathbf{r}^{s*}_j$ onto surfaces containing its homologue master node $\mathbf{r}^{m*}_j$. The case when only one projection is found is trivial. There remains only to create the corresponding contact element spanned on the slave node and master surface possessing this projection. If several projections are found we choose the closest one and create the contact element. The case when no projection is found is not as trivial as the preceding ones and has to be considered in details. There are two possibilities:

1. the slave node is situated in a “blind spot” of the discretized master surface;
2. the slave node does not come in contact but just passes by close to the boundary of master surface.

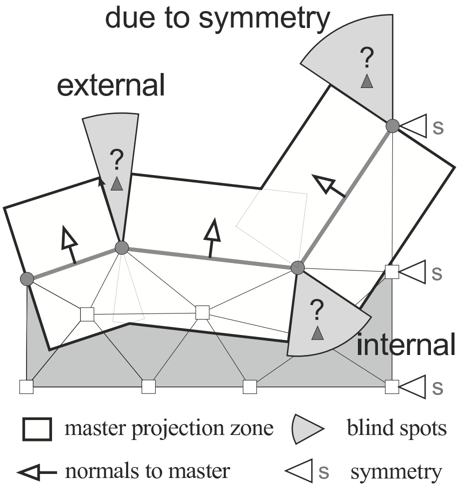

Fig. 6 Examples of blind spots: external, internal and due to the symmetry boundary conditions.

Case 1. Since the finite element method requires only continuity of the discretization ($\Gamma_c \in C^0$) so the contacting surface may be not smooth ($\Gamma_c \notin C^1$). Each master segment has its "projection" zone (Figure 6), each point in this zone has at least one projection onto the master surface [7, 8]. But often in the junction zone of master segments (at common edges and nodes) the intersections of "projection" zones does not fill the surrounding space entirely but with some gaps of form of prisms and pyramids in 3D or of form of sectors in 2D. This problem exists not only for linear master elements but for any order. Three types of blind spots can be distinguished: internal, external or blind spot due to boundary conditions (see Figure 6). If a slave node in a blind spot is overlooked, different consequences depending on the type of blind spot are possible.

- External blind spot. Slave nodes situated in this kind of spot are not detected before they penetrate under the master surface. After such penetration during the next time step it can be detected and brought back onto the surface, but the solution has been already slightly changed. In certain cases especially in force driven problems such penetration can lead to a failure of solution.
- Internal blind spot. Contact is predicted correctly, but if slave node penetrates just a little under the master surface and appears in its internal blind spot this node will be lost for the contact detection at least during the next time step. Such little penetrations take place if the penalty method for contact resolution is used or just due to the limited precision of the iterative solution.
- Blind spot due to boundary conditions. This type of blind spot is situated at the boundary and can be either internal or external. It appears due to the presence of symmetric or periodic boundary conditions on the master surface, for example, the basic Hertz contact problem with axisymmetrical 2D finite element mesh.

Obviously if the detection procedure which does not consider blind spots is repeated at every iteration then some problems can be avoided but on the other hand it is very

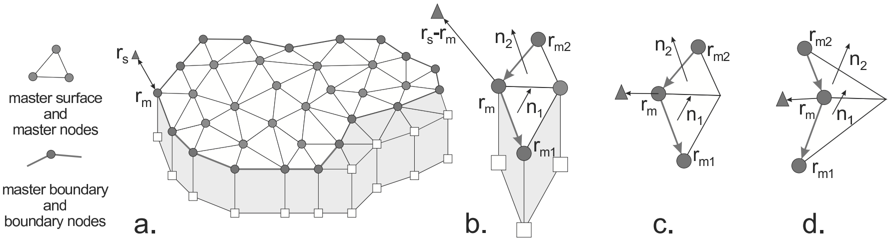

Fig. 7 Detection of the passing by nodes. (a) Master surface and its boundary. (b) Zoom on the geometry close to the passing by node. (c) Convex master boundary. (D) Concave master boundary.

time consuming and not always efficient (for example, if a convex edge with an external blind spot becomes a concave with an internal blind spot the penetration of the slave node situated in this spot can be irreversible).

There are different possibilities to avoid the loss of contact in blind spots:

- Artificial smoothing of master surface for large sliding contact problems [8, 12]. There are no more gaps in "projection" zones except gaps due to symmetry, i.e. there are almost no more blind spots and the problem of passing by nodes (Case 2) does not exist. However most of these methods have some inherent drawbacks: and derive sometimes not correct deformation close to the edge of the master surface boundary.
- A "proximal volume" can be constructed by an extrusion of the master surface in the normal direction and in the opposite one which fills both projection zones and blind spots. If a slave node is situated in this volume then it is considered as node in contact and the master surface is further detected. "Passing by" nodes can be easily detected as they do not appear in the "proximal volume".

The first group of methods in general is too "expensive" if one uses them only for the detection purpose and are not applicable for arbitrary meshes, the second one is quite time consuming as well.

We use rather rough but quite simple and robust treatment of blind spots. If a detected slave node has no projection and is not a passing by node then the corresponding contact element is constructed with the closest [15] or randomly chosen master surface attached to the closest master node. For sufficiently small time step such approach is quite reliable. There remains only to determine if the node is passing by or not. One possible technique is represented in Figure 7.

Case 2. First of all, in the preliminary phase the boundary master nodes surrounding the master contact surface have to be marked. Let us assume the situation when one of such marked nodes $\mathbf{r}_m$ is found to be the closest to slave node $\mathbf{r}_s$. If it has no projection onto master segments attached to the marked master node, then two alternatives are possible: either the slave node is situated in a blind spot or it passes by the master surface. To choose between these possibilities it is possible either to verify if the slave node is situated in one of blind spots attached to the master node or to check if the slave node is situated in the local proximal volume of the master surface. The second possibility seems to be more simple and natural. Note that such a verification is slightly different for locally convex and concave master surface boundaries. The convexity can be known as nodes of each master segment are ordered. The condition of convexity is

$$(\mathbf{r}_m - \mathbf{r}_m 2) \times (\mathbf{r}_{m1} - \mathbf{r}_m) \cdot (\mathbf{n}_1 + \mathbf{n}_2) \geq 0, \tag{12}$$

where $\mathbf{n}_1$ and $\mathbf{n}_2$ denote average normals to master segments possessing the edges $\{\mathbf{r}_m, \mathbf{r}_{m1}\}$ and $\{\mathbf{r}_m, \mathbf{r}_{m2}\}$ respectively. Then the criterion of the slave node being in the proximal volume is

$$\mathbf{n}_2 \times (\mathbf{r}_m - \mathbf{r}_{m2}) \cdot (\mathbf{r}_s - \mathbf{r}_m) \geq 0 \text{ AND } \mathbf{n}_1 \times (\mathbf{r}_{m1} - \mathbf{r}_m) \cdot (\mathbf{r}_s - \mathbf{r}_m) \geq 0. \tag{13}$$

If this condition is fulfilled then the slave node is taken into account and the contact element is established with the closest master segment. For the concave surface AND should be replaced by OR in (13).

More elaborated approaches (see, for example, [7]) take into account node-to-node and node-to-edge contacts in case of blind spot detection, average normals can be established at edges and vertices. But for most of the contact problems, consideration of only node-to-segment discretization provides the correct results.

As one can see the proposed algorithm is quite simple and natural except may be the verification of passing by nodes. The proposed method does not require any special data storage nor particular code structure and consequently can be easily implemented in any finite element code.

2.3 Validation and Performance

The preliminary validation of the grid detection method is easy to carry out on simple meshes. Normally a visual analysis of the constructed contact elements is sufficient. The further validation consists in comparison with the all-to-all detection method which is trivial to implement.

To demonstrate the performance of the grid detection method we consider a tyre-road contact problem. Such simulation can be rather helpful for example for an improvement of tread patterns (stick increase and noise reduction). We are particularly interested in this problem because the contact elements change intensively at each time step and consequently a fast detection procedure is highly desirable.

A finely and regularly meshed tyre wheel is translated over an artificially rough road surface and its FE mesh is deformed manually accordingly to the road roughness and next the contact detection procedure is executed. The finite element mesh of the tyre (Figure 8) consists of about 550,000 nodes with contact zone of about 105,000 nodes. The finite element mesh approximating the road roughness (Figure 9) consists of about 400,000 nodes one half of them being included in the master contact zone. Established contact elements are demonstrated in Figure 9 for different tyre-road dispositions and imprint deep. It can be noted that the choice of the bounding box as an intersection of master and slave bounding boxes reduces sig-

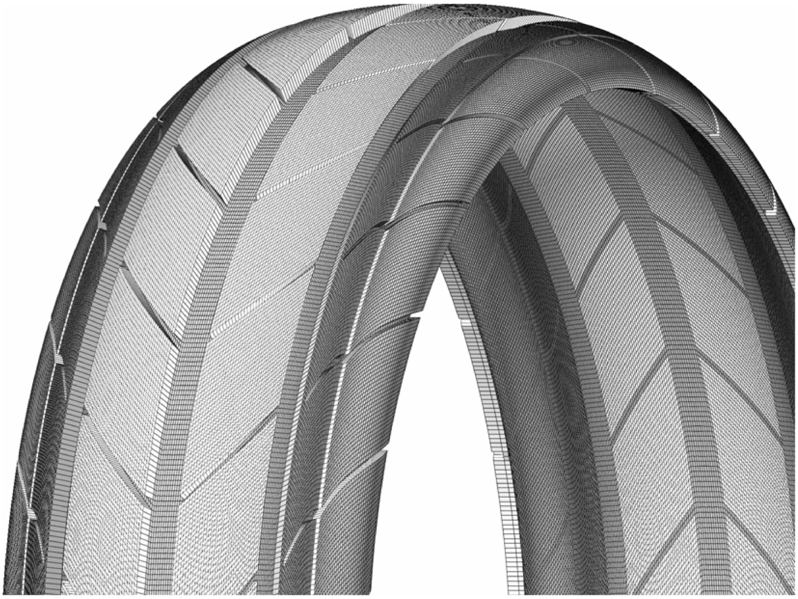

Fig. 8 A part of the tyre finite element mesh consisting of about 550,000 nodes.

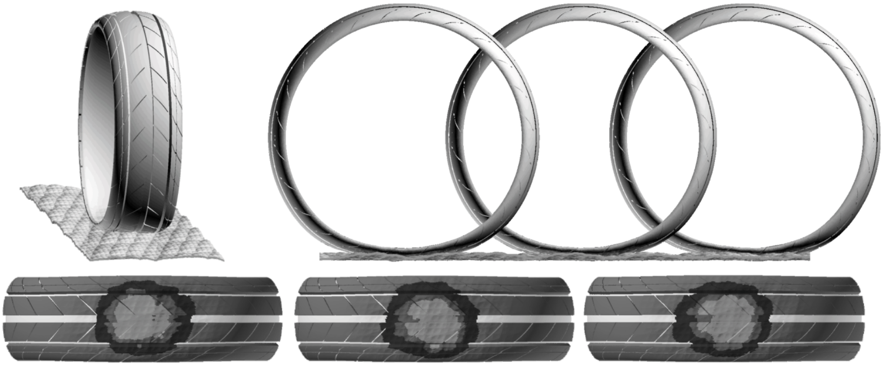

Fig. 9 Tyre-road contact problem: general view, three tyre-road dispositions and corresponding contact elements on the bottom of the tyre for different imprint deep.

nificantly the number of contact nodes to be considered. The bounding box of the road is kept constant, whereas the bounding box of the tyre is updated at each step. The contact detection time at each time step in average is just 1.5–2 seconds on a laptop, i.e. the contact detection time can be neglected in comparison to the system resolution time. The analysis of the detection time shows that the estimation of the maximal detection distance takes about 30% of the time, preliminary stage takes about 20% and the detection procedure requires just 50% of the time.

Table 1 Detection of contact between rough surfaces (2 millions of master and slave nodes).

Geometry	Nodes in BB*	Contact elements	Detection time	Gain, $T_{\mathrm{all-to-all}}/T_{\mathrm{grid}}$
Two close surfaces	2,100,000	75,300	35 minutes	>300 times
Two convex surfaces	340,000	15,800	1 minute	>10,500 times
Two close but not contacting surfaces	50,000	0	4 seconds	>160,000 times

*BB – Bounding box

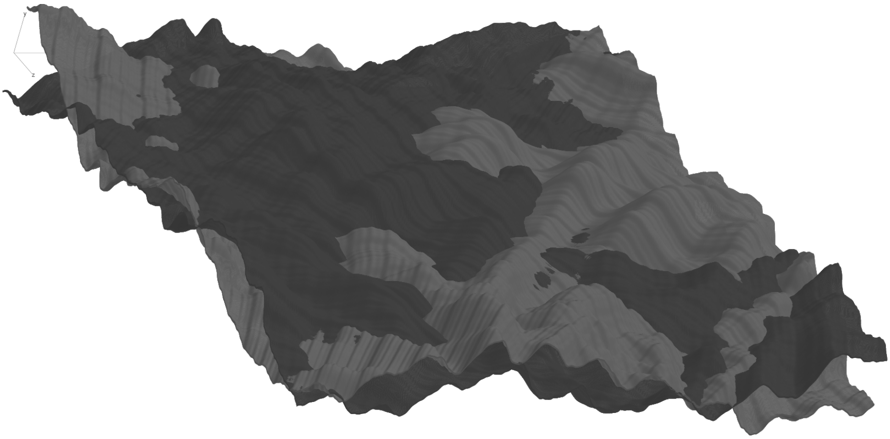

Fig. 10 Rendered surfaces of two finite element meshes (each contains 2^{20} contact nodes).

Another example is an artificial contact between two rough surfaces, each consisting of 2^{20} contact nodes. Rendered surfaces corresponding to the meshes are represented in Figure 10. Such a kind of problems requires a longer time for contact detection because the bounding box includes all or almost all contact nodes and there are as many slave nodes as master ones. If one uses the modified all-to-all method (not node-to-segment but node-to-node), the reliable estimation of the needed detection time exceeds 180 hours (!) (almost 8 days) and 2^{40} distance verifications are needed. The proposed grid detection method requires much less time than all-to-all method. The time strongly depends on the geometry and discretization, consequently on the constructed bounding box and the number of contact nodes located in it, for example, for close enough rough surfaces (Figure 10) the detection time is much higher than for convex surfaces and it is almost negligible if two surfaces are close enough but not so close to come in contact. The results are summarized in Table 1. Let us note that in presented computations the quadrilateral master segments are supposed to remain flat for both considered methods.

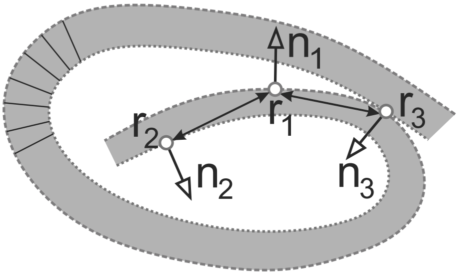

Fig. 11 Indistinguishable contact nodes in self-contact in case of the maximal detection distance higher than the minimal structure thickness.

3 Self-Contact Detection

There are mechanical problems for which determination of master and slave surfaces presents a big challenge or may be impossible. Among such problems there are multibody systems, problems with complicated geometries (for example highly porous media like metal foams), large deformation problems with not regular discretization and self-contact problems.

This class of contact problems needs a particular contact detection procedure. Here an adaptation of the grid detection method to problems with unknown a priori master-slave discretization is proposed, particular attention is paid to self-contact problems. Such adaptation demands considerable modifications in all stages of the grid detection procedure. Moreover an adapted finite element mesh can be required to make the detection possible. The growth rate of the method is the same as for the case of known a priori master-slave discretization. The method is straightforward and it does not need any complicated constructions and three data organization, as for example in recently proposed technique for mortar formulation of contact [14].

A self-contact is more probable for thin or oblong solids, for which one or two dimensions are much smaller than others, than for solids with all dimensions of the same order. But there is a challenge which reveals itself in Figure 11. For a thin solid with two sided contact zone in general case it is impossible to distinguish the contact with the reverse side node ($\mathbf{r}_1$ can be in contact with $\mathbf{r}_3$) from a simple neighbouring with it ($\mathbf{r}_1$ is close but cannot come in contact with $\mathbf{r}_2$). Even if in addition to node positions their normals $\mathbf{n}_1$, $\mathbf{n}_2$, $\mathbf{n}_3$ and corresponding surfaces are taken into account there is no way to distinguish $\mathbf{r}_2$ and $\mathbf{r}_3$. A possible solution to overcome this problem is to generate a finite element mesh with contact surfaces smaller than the minimal thickness of the structure (Figure 11). It provides the maximal detection distance higher than the distance to the back side and allows to avoid this confusion. But in a less general case the possibility of two sided contact can be omitted and two sides can be treated independently. In this case, the maximal detection distance should be limited by the doubled minimal thickness of the structure.

Let us enumerate the features of the implementation of the grid detection method in case of unknown a priori master-slave discretization. The main modification is that not only node coordinates but also associated normals have to be taken into account to determine potentially contacting elements as in [1].

1. The bounding box has to include all nodes of contact surface; it can be either constant if we know a priori a sufficiently small area where from the contact nodes do not escape or it can be a bounding box spanned all contact nodes.
2. In the beginning of every time step the normal have to be assigned to each contact node. An average of average normals of attached contact surfaces can be used.
3. Only one array A^c is created and filled with contact nodes. The logic is the same as in the case of simple contact.
4. Since we cannot distinguish master and slave nodes the detection of the closest node has to be carried out for each contact node $\mathbf{r}^c_{ij}$ against all other nodes $\mathbf{r}^c_{il}$, $l \neq j$ in the cell i. To be sure that the closest nodes can come in contact and are not attached to a common segment, the normals associated with nodes are checked to form an obtuse angle $\mathbf{n}_{ij} \cdot \mathbf{n}_{ik} \leq 0$. Obviously some neighbouring cells have to be verified as in case of simple contact.
5. When two proximal contact nodes j and k are detected then in order to determine the NTS contact element the local mesh density has to be analyzed. If the surface mesh surrounding j node is found to be more rough than the surface mesh of the second node k, then the node j is considered as master node, otherwise as slave. If one local mesh is as fine as another one, then the node j has to be checked against each surface attached to the node k and vice versa. If the projection exists then the contact element is created otherwise arbitrary one of nodes is considered as a slave and its opponent as a master and if the slave node is not passing by then the contact element is created.

Being adapted for the case of unknown master-surface, the detection procedure has been verified on the challenging artificial problem of the self-contact within a snail-operculum-like structure containing over 130,000 nodes on the surface, all nodes with attached segments are included in the contact detection step (see Figure 12). The detection time is higher than for the contact of the same order with known a priori master-slave discretization, because the preliminary stage requires the assignation of normals to every node and also because the main detection stage requires significantly more verifications of distances and normals than in master-slave conception. In practice the difference in detection time between known a priori and unknown master-slave depends significantly on the geometry and its evolution. For example, for the snail operculum problem for the known master-slave the detection time (≈ 9 sec) is only three times faster than for unknown a priori master-slave (≈ 30 sec).

In conclusion we affirm that the grid detection method can be adapted to the class of contact problems with unknown a priori master-slave discretization. The required detection time is of the same order of magnitude as the time needed for simple contact detection for the same problem. Availability of such powerful method extends significantly the capacities of the finite element analysis of contact problems.

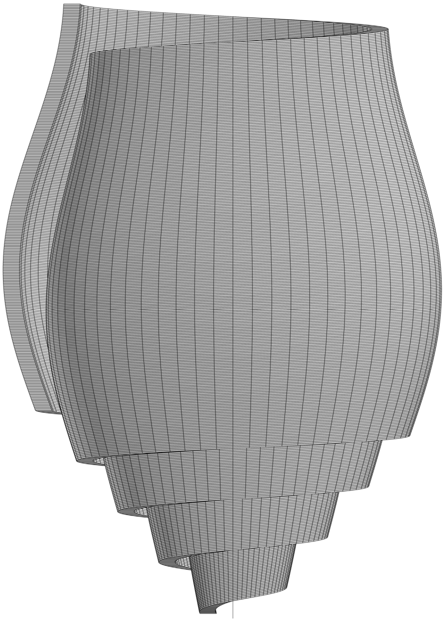

Fig. 12 Finite element mesh used to test the detection procedure for self-contact problems.

4 Parallelization

Sequential treatment of the problems presented above requires either too long computational time or even impossible due to the great amount of memory needed. The use of the parallelization paradigm is a good way out. Many parallelization techniques are available nowadays, the class of non-overlapping domain decomposition, also called iterative substructuring methods, is successfully and widely used in computational mechanics, see [3, 5, 9]. It implies a splitting of an entire finite element mesh into subdomains which intersect only on their interfaces. Each subdomain is treated by one or several associated processors and further the continuity of the solution across subdomain interfaces is enforced by displacement and force equality. The use of these techniques with affordable and powerful parallel computers allows to solve very large mechanical problems in reasonable terms. Since the resolution follows the detection procedure so the last one is very important for the efficiency of parallel computations [2]. It should not present a bottleneck in the whole process and if it is possible it has to use all available capacities of parallel computers.

The essential thing for the contact detection procedure in parallel treatment is the fact that the finite element mesh and possibly the contact surface is divided into some parts associated with different processors and in the case of distributed memory it is not available entirely on a particular processor. Since in principle we need the entire contact surface(s) to perform the detection procedure this repartition implies the data exchange between subdomains containing different parts of this surface(s). The smaller the amount of data transfer between subdomains on distributed memory, the faster the algorithm computers. It will be demonstrated below how this data transfer can be reduced significantly in the framework of the grid contact detection method.

Two ways of parallel treatment of contact problems are proposed and analyzed: Single processor Detection, Multiple processor Resolution (SDMR) and Multiple Detection, Multiple Resolution (MDMR). As it is evident from the notations SDMR carries out the contact detection on a single processor whereas MDMR uses all available resources. The last implies a parallelization of the detection procedure which will be discussed in details and tested.

First, let us consider the SDMR approach. The main idea is that all necessary information is collected by one processor which carries out the contact detection and distributes consequently the created contact elements among all concerned subdomains. This method can be efficiently applied to any contact problem and is easy to implement. On the other hand this method does not use efficiently all available resources, i.e. all except one processors are idle and inactive during the main detection phase however all processors possessing contact surface are active during the preliminary stage.

At first, the bounding box for the contact detection has to be defined. This task is easily performed in parallel. Each subdomain $i \in [1;N^c]$ possessing a part of contacting surfaces examines it and derives the corresponding bounding boxes ${}^m\mathbf{r}^1_i$, ${}^s\mathbf{r}^1_i$, ${}^m\mathbf{r}^2_i$, ${}^s\mathbf{r}^2_i$ and the maximal dimension of master segment $d^i_{\max}$. Further by means of data transfer the global maximal detection distance $d_{\max} = \max_{i=1}^{N^c}\{d^i_{\max}\}$ and master and slave bounding boxes are determined

$$ {}^{m,s}r^1_{\{x,y,z\}} = \min_{i=1}^{N^c}\{{}^{m,s}r^1_{i\{x,y,z\}}\} - d_{\max}, \quad {}^{m,s}r^2_{\{x,y,z\}} = \max_{i=1}^{N^c}\{{}^{m,s}r^2_{i\{x,y,z\}}\} + d_{\max}. \qquad (14) $$

Finally, the resultant bounding box $\{\mathbf{r}^1,\ \mathbf{r}^2\}$ is constructed as the intersection of master and slave bounding boxes, exactly as in the sequential procedure. The data transfer consists in maximum $3N^c$ sends but the load is not uniformly distributed between processors, because not all of them contain the contact surface and among possessing it, the size of this surface can be quite different. In all cases this operation is quite fast even for huge meshes.

Next step consist in the union of all necessary parts of the contact surface at one processor-detector. First, the information about the global bounding box is distributed among the subdomains possessing the contact surface, each of them counts the number of master and slave nodes located in the bounding box, further the subdomain with the maximal number of master and slave nodes is chosen as the detector. Another possibility is that this choice can be made in concordance with processors network topology to accelerate the data transfer on the next detection step. As one can see, at this stage the data exchange between subdomains remains very limited.

It remains to transfer all master and slave nodes from the bounding box (global IDs, hosting subdomain ID, coordinates, and attached surfaces) to the detector subdomain, to carry out the detection as it is described in Section 2 and to distribute the

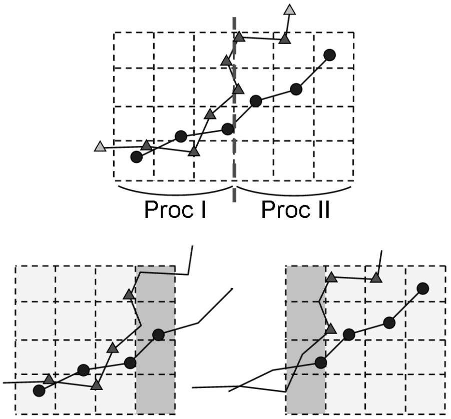

Fig. 13 Example of cells partition between two processors. Each one gets one half of the total number of cells (with slave and master nodes – represented by triangles and circles respectively) as well as one boundary layer from another half which contain only master nodes.

constructed contact elements between the corresponding subdomains. If a contact element unions slave node and master nodes from different subdomains, the interface between them has to be created or updated as well as duplicated slave or master nodes have to be formed.

In MDMR (Multiple Detection, Multiple Resolution) the preliminary part of a bounding box construction is exactly the same as in SDMR approach. The key difference between MDMR and SDMR consists in the following step. Instead of transferring all the necessary information to the detector, in MDMR this information is distributed between all subdomains in a special way. As it was shown above the grid is constructed in a way that for each slave node only one surrounding layer of neighbouring cells has to be verified to find the closest master node. If the self-contact is excluded from the consideration we do not care about slave nodes in neighbouring cells. That is why the bounding box can be divided into *N* non-overlapping parts, each part consists of integer number of cells. Further, each part is extended in all directions (not exceeding the bounding box) by a one-cell-overlapping layer; the extended part is filled only with master nodes (see an example for two subdomains in Figure 13). In other words each part consists of internal cells (non-overlapping with other parts) including both master and slave nodes and external cells (shared with neighbouring parts) including only master nodes. Each part is associated with a processor and all necessary data: nodes and surfaces located in the part (global IDs, hosting subdomain ID, coordinates, and attached surfaces) is collected from different subdomains and transferred to the considered one. Consequently the detection can be carried out absolutely independently, i.e. in parallel in each part. No additional data exchange is needed and it increase significantly the performance and scalability of the MDMR approach. The advantage of the method is that the total number of operations per processor during the main phase of detection does not increase with proportional increasing number of processors and contacting nodes. However, during the main detection phase the number of operations is not distributed homogeneously between processors.

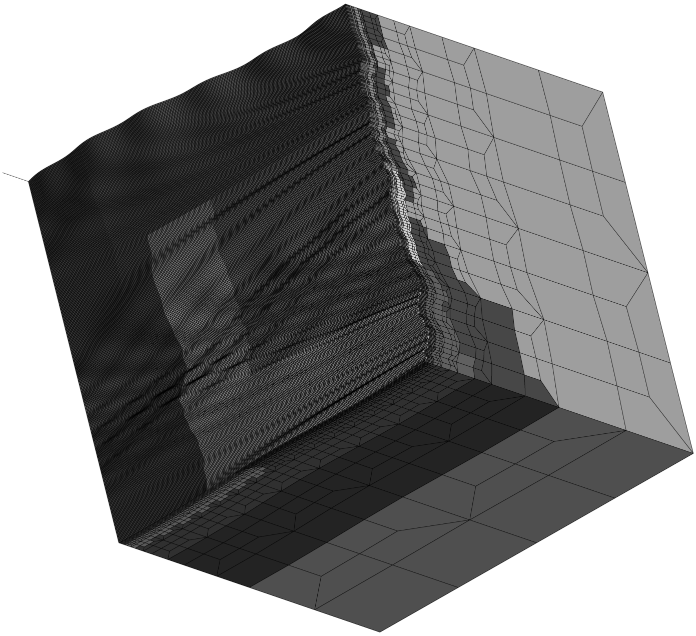

Fig. 14 The split of the FE mesh split into 16 sub-domains for parallel computations.

The same parallel procedure can be used for self-contact problems. The only difference is that master and slave nodes are not distinguished and hence all contact nodes have to be included in the overlapping cells. The described method is very similar to the parallelization of the Linked Cell Method widely used in molecular dynamic simulations for short-range interactions [6].

In Figure 14 the finite element mesh of a rough surface is presented, in the figure different tones of grey correspond to subdomains. The scalability test for MDMR approach has been performed between two such meshes containing over 560,000 nodes and over 66,000 contact nodes each. The scalability test for such meshes with slightly different surface roughness is represented in Figure 15. The heterogeneous distribution of active contact zones means that the parts of bounding box associated with different processors have quite different number of potential contact elements; the homogeneous distribution means that this number is more or less similar for different parts. And the average gain stands for averaged time of processors work. The difference between linear gain and the average gain makes evident the time necessary for the data exchange between subdomains. The pronounced difference between the gain for heterogeneous and homogeneous active contact zones distributions can be explained by the following observation. If there is no master node in the cell of the slave node, nor in the neighbouring cells, the time needed to conclude that is very small. On contrary if the considered cells are not empty and contain several master nodes it takes a longer time to derive the coordinates of these nodes, to

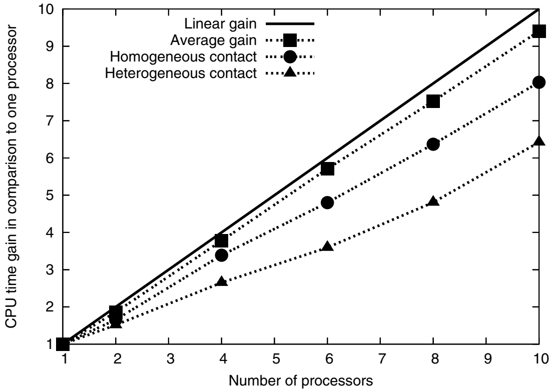

Fig. 15 Time gain for parallel contact detection procedure.

compare them with the slave node and to verify a projection availability. Nevertheless the gain is quite high and its rate does not decrease with increasing number of detecting processors (for reasonable ratio of contact nodes to number of processors).

The SDMR and MDMR approaches can be efficiently applied to parallel contact treatment. The second approach requires a larger amount of programming but its performance allows to neglect the detection time for large and extremely large contact problems.

5 Conclusion

The very fast local detection method has been elaborated on the base of the bucket method. Sequential and parallel implementations of the method have been discussed in details for known a priori and unknown master-slave discretizations.

The strong connections between the finite element mesh of the master surface, the maximal detection distance and the optimal dimension of detection cells are established. Analytical estimation and numerous tests demonstrate that the optimal cell size is equal to the maximal detection distance which by-turn is equal to the dimension of the biggest master segment. The particular attention in the article has been paid to the bounding box construction, optimal choice of the neighbouring cells to be verified, “passing by node” and blind spot analysis, master–slave surfaces definition in self-contact and especially to an efficient implementation of the method on distributed memory parallel computers.

The method is very flexible but it is not well adapted neither for very heterogeneous distribution of master segment dimensions nor for very different mesh densities of master and slave surfaces. In the method, the dimension of the biggest master segment is strongly connected with the maximal detection distance and consequently with the cell size. Therefore if the master surface has at least one segment which dimension is 10–100 times larger than the dimension of an average segment the detection time can be rather high, but always less than in all-to-all approach.

The validation of the method has been performed on different contact problems in sequential and parallel cases: contact between rough surfaces with different geometries, tyre-road contact, self-contact of a snail operculum and on the extremely large contact problem between two rough meshes including more than 1,000,000 segments at master surface against 1,000,000 slave nodes. In the latter problem, the detection time varies significantly for different geometries from several seconds to 30–40 minutes in comparison to almost 8 days needed for all-to-all detection techniques.

Acknowledgements The implementation of the algorithm and all demonstrated analyses have been carried out in the FEA package ZéBuLoN developed by the Centre des Matériaux (MINES ParisTech, France), ONERA (France) and Northwest Numerics (USA). This work has been performed with the support of the CNRS-SNECMA grant 920 78122.

References

1. Benson, D.J., Hallquist, J.O.: A single surface contact algorithm for the post-buckling analysis of shell structures. Computer Methods in Applied Mechanics and Engineering 78, 141–163 (1990)
2. Brown, K., Attaway, S., Plimpton, S., Hendrickson, B.: Parallel strategies for crash and impact simulations. Computer Methods in Applied Mechanics and Engineering 184, 375–390 (2000)
3. Farhat, C., Roux, F.-X.: Implicit parallel processing in structural mechanics. Computational Mechanics Advances 2(1), 1–24 (1994)
4. Fujun, W., Jiangang, C., Zhenhan, Y.: A contact searching algorithm for contact-impactor problems. Acta Mechanica Sinica (English series) 16(4), 374–382 (2000)
5. Gosselet, P., Rey, C.: Non-overlapping domain decomposition methods in structural mechanics. Archives of Computational Methods in Engineering 13(4), 515–572 (2006)
6. Griebel, M., Knapek, S., Zumbusch, G.: Numerical Simulation in Molecular Dynamics. Springer, Berlin (2007)
7. Konyukhov, A., Schweizerhof, K.: On the solvability of closest point projection procedures in contact analysis: Analysis and solution strategy for surfaces of arbitrary geometry. Computer Methods in Applied Mechanics and Engineering 197(33/40), 3045–3056 (2008)
8. Pietrzak, G., Curnier, A.: Large deformation frictional contact mechanics: Continuum formulation and augmented Lagrangian treatment. Computer Methods in Applied Mechanics and Engineering 177, 351–381 (1999)
9. Toselli, A., Widlund, O.: Domain decomposition methods – Algorithms and theory. Springer, Berlin (2005)
10. Williams, J.R., O'Connor, R.: Discrete element simulation and the contact problem. Archives of Computational Methods in Engineering 6(4), 279–304 (1999)

11. Wriggers, P.: Computational Contact Mechanics, 2nd edn. Springer, Berlin (2006)
12. Wriggers, P., Krstulovic-Opara, L., Korelc, J.: Smooth c1-interpolations for two-dimensional frictional contact problems. International Journal for Numerical Methods in Engineering 51(12), 1469–1495 (2001)
13. Yang, B., Laursen, T.A.: A contact searching algorithm including bounding volume trees applied to finite sliding mortar formulations. Computational Mechanics 41(2), 189–205 (2008)
14. Yang, B., Laursen, T.A.: A large deformation mortar formulation of self contact with finite sliding. Computer Methods in Applied Mechanics and Engineering 197(6/8), 756–772 (2008)
15. Zavarise, G., De Lorenzis, L.: The node-to-segment algorithm for 2D frictionless contact: Classical formulation and special cases. Computer Methods in Applied Mechanics and Engineering 198, 3428–3451 (2009)