INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2003; **58**:2083–2101 (DOI: 10.1002/nme.845)

Improvement of a frictional contact algorithm for strongly curved contact problems

M. C. Oliveira[1,*,†], J. L. Alves[2] and L. F. Menezes[1]

[1] *Department of Mechanical Engineering, University of Coimbra, Polo II, 3030 Coimbra, Portugal*
[2] *Department of Mechanical Engineering, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal*

SUMMARY

One of the challenges in contact problems is the prediction of the actual contact surface and the kind of contact that is established in each region. In numerical simulation of deep drawing problems the contact conditions change continuously during the forming process, increasing the importance of a correct evaluation of these parameters at each load step. In this work a new contact search algorithm devoted to contact between a deformable and a rigid body is presented. The rigid body is modelled by parametric Bézier surfaces, whereas the deformable body is discretized with finite elements. The numerical schemes followed rely on a frictional contact algorithm that operates directly on the parametric Bézier surfaces.

The algorithm is implemented in the deep drawing implicit finite element code DD3IMP. This code uses a mechanical model that takes into account the large elastoplastic strains and rotations. The Coulomb classical law models the frictional contact problem, which is treated with an augmented Lagrangian approach. A fully implicit algorithm of Newton–Raphson type is used to solve within a single iterative loop the non-linearities related with the frictional contact problem and the elastoplastic behaviour of the deformable body.

The numerical simulations presented demonstrate the performance of the contact search algorithm in an example with complex tools geometry.

KEY WORDS: contact with friction; large deformation; global contact search; local contact search; parametric surfaces

1. INTRODUCTION

Nowadays the importance of the numerical simulation in the shortening of the conception cycles of new deep drawn parts is unquestionable. This results from the ability it presents in the optimisation of the process parameters involved like the lubrication conditions, the

[*] Correspondence to: M. C. Oliveira, Department of Mechanical Engineering, University of Coimbra, Polo II, 3030 Coimbra, Portugal.
[†] E-mail: marta.oliveira@dem.uc.pt

Contract/grant sponsor: POCTI Programme; contract/grant number: EME/35945/99
Contract/grant sponsor: Growth Programme; contract/grant number: G1RD-CT-2000-00104

Received 6 August 2002
Revised 5 February 2003
Accepted 24 March 2003

prediction of the required tools force (punch, blank holder and die), the gap, etc. Another advantage is the fact that in function of the chosen material for the blank sheet it is possible to perform a direct comparison of the process parameters' behaviour.

Many different strategies for solving this problem have been discussed and many numerical simulation codes developed. The algorithms presented in this paper correspond to the ones implemented in the finite element code DD3IMP.[‡] The mechanical model regarded takes into account large elastoplastic strains and rotations. The plastic behaviour of the material is described by the Hill's 1948 yield criterion with isotropic and kinematic work hardening, and by an associated flow rule. Several work hardening constitutive models have been implemented in order to allow a better description of the different material mechanical behaviour [1]. The kinematic description follows an updated Lagrangian formulation, with no simplifications of membrane or shell type. The code uses solid finite elements that allow to consider the sheet as a real three-dimensional domain. Thus, the contact with both sides of the sheet is naturally solved without any particular strategy. The use of solid elements presents an enormous cost in terms of CPU time. Nevertheless, they allow an accurate calculation of the stress gradients through the thickness of the sheet, as well as the thickness evolution [2–4]. The contact with friction is described by the Coulomb law. The evolution of the contact surface and status is controlled by a mixed formulation using an augmented Lagrangian approach, which has been successfully applied in this kind of problems [5, 6]. The advantage of this formulation is the transformation of the non-differential contact and frictional laws into a partial differential problem without constraints, which resolution is possible applying an algorithm of Newton–Raphson type. It results, in each equilibrium iteration, in a linear system of equations where the unknowns are the incremental frictional contact forces and the incremental displacements. The different physical nature of the unknowns results in an ill-conditioned system. However, it is possible to transform this mixed system in a reduced linear system of equations easier to solve, where only the incremental displacements are unknowns. This is done without altering the main characteristic of the global algorithm of DD3IMP code, that is the use of a single iterative loop to solve the non-linearities related with both the mechanical behaviour and the contact with friction [7, 8].

Following previous works, this paper discusses the use of a frictional contact algorithm for the numerical simulation of contact problems that involve strongly curved rigid obstacles [7, 9, 10]. In particular, it presents a new strategy in order to improve the performance of the contact search when dealing with rigid bodies defined by parametric surface patches.

Section 2 presents a summary of the contact mechanics background and motivation. Section 3 is devoted to the description of the mixed formulation of the contact with friction problem based on an augmented Lagrangian approach. Special attention is given to the derivation of a frictional contact operator. In Section 4 the contact search algorithms are discussed, with a special focus on a new strategy for the global contact search algorithm. The influence of this global contact search algorithm on the local contact search algorithm is discussed in Section 5, where a simple example is presented in order to demonstrate such influence. Finally, the summary of the conclusions of this paper is presented in Section 6.

[‡] Contraction of 'Deep Drawing 3-D IMPlicit'.

2. CONTACT MECHANICS

To determine if a node that belongs to the boundary of the deformable body (Γ^{def}) is in contact with the rigid body, it is necessary to know the current position $\mathbf{x}(\mathbf{u})$ of the node. From the updated Lagrangian formulation, the nodal current position $\mathbf{x}(\mathbf{u})$ can be determined from the equilibrium position in the *prior* load step added of the corresponding nodal displacement vector $\mathbf{u}$. Since the rigid obstacle is not stationary, it is preferable to associate a local frame on the boundary of the rigid obstacle (Γ^{rig}), and to determine the relative motion using the rigid body as reference. The local contact frame in the rigid body is defined with the outward unit normal vector to Γ^{rig}, $\mathbf{n}(\mathbf{u})$. It is important to notice that due to the curved contact, in particular in case of large slip, this vector is not constant. It is unknown at the beginning of each load step, and it can assume different values during the iterative equilibrium loop.

The contact kinematics is determined as a function of two variables: the signed normal distance d^{n}, and the relative tangential velocity $\dot{\boldsymbol{\delta}}^{\text{t}}$. They are both defined between a node of Γ^{def} and a reference position on Γ^{rig}, designated here as $\mathbf{x}^{\text{ref}}$ (origin of the local contact frame). The signed normal distance is involved in the formulation of the unilateral contact law, and the relative tangential velocity in the formulation of the friction law.

2.1. *Reference position*

To determine the reference position $\mathbf{x}^{\text{ref}}$ two different approaches can be used: an impact algorithm or a projection algorithm. The impact algorithm is based in the relative displacement of the node of Γ^{def} and the obstacle Γ^{rig}. If the node was in contact in the previous time step, the reference position can be easily established from the displacement of the contact obstacle. But, if the node was not in contact at the end of the previous step, the calculus of a reference position is less trivial (see Section 4.3). The projection algorithm determines the reference position with the orthogonal projection of the current position of the node of Γ^{def} on Γ^{rig}, minimizing the normal distance between both positions (see Section 4.4).

In Figure 1 the use of the impact and projection strategies is described. For a known updated current position $\mathbf{x}(\mathbf{u})$, the solution using the impact strategy is unique ($\mathbf{x}^{\text{ref}}$). On the contrary, the computation of a projection point over a tool defined with several parametric surface patches can conduct to multiple solutions. As depicted in this figure, for this case, the projection strategy leads to two different solutions ($\mathbf{P}_1^{\text{Proj}}, \mathbf{P}_2^{\text{Proj}}$). The figure also reveals

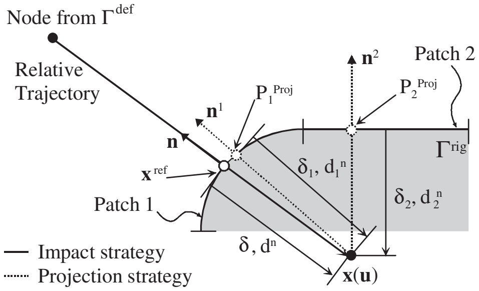


Figure 1. Definition of the kinematic variables with different local contact search strategies. Different solutions for the contact search projection strategy.

the fact that although the projection strategy performs a selection based on the minimum normal distance, this may lead to an intuitively erroneous reference position ($\mathbf{P}_2^{\text{Proj}}$). The impact solution is unique if the relative displacement between the node and the obstacle is non-zero, however this algorithm is numerically less robust than the projection one. In some particular situations the solution from both algorithms can be the same.

The deep drawing implicit code DD3IMP takes advantage of both strategies for the computation of the reference position in its numerical model. At each time increment, the global algorithm of DD3IMP code can be divided into two phases. The first determines an approximate first solution for the incremental displacements, stresses and frictional contact forces by an explicit method. Due to the non-linearity of the behaviour law and to the kinematics of the process this explicit solution generally satisfies neither the variational principle nor the consistency condition. In the second phase the explicit solution is therefore successively corrected using an implicit method, until the structural equilibrium is ensured [3, 7].

In the determination of the explicit first solution is necessary to guarantee that a reference position is established for all candidate nodes belonging to Γ^{def}, in order to avoid contact problems during the corrector phase. This reference position must be determined whether the node was previously in contact or not. The impact search strategy allows a unique definition of a reference position for each candidate node belonging to Γ^{def}, which is associated with the imposed displacement of the rigid body. Thus, this is the privileged strategy for the computation of the reference position in the first phase. However, in order to assure that no candidate node is neglected, if no reference position is established with the impact algorithm, an assessment is performed with the projection algorithm. Finally, in the corrector phase the static structural balance is assured by an iterative scheme in the vicinity of the explicit configuration. To perform these corrections it is better to calculate the reference position with the projection algorithm, since this is numerically more robust and allows a better selection between multiple close solutions.

2.2. *Kinematic variables*

The knowledge of the reference position $\mathbf{x}^{\text{ref}}$ allows the calculus of the kinematic variables. The outward unit normal vector $\mathbf{n}(\mathbf{u})$, evaluated in $\mathbf{x}^{\text{ref}}$, changes in agreement with the predicted implicit local reference position. In this way, for each candidate node to contact, a normal distance to the rigid body that imposes the constraints is established:

$$\forall \mathbf{x}(\mathbf{u}) \in \Gamma^{\text{def}}, \quad d^{\text{n}} = \lfloor \mathbf{x}(\mathbf{u}) - \mathbf{x}^{\text{ref}} \rfloor \cdot \mathbf{n}(\mathbf{u}) \tag{1}$$

In quasi-static formulations the relative tangential velocity ($\dot{\boldsymbol{\delta}}^{\text{t}}$) can be directly replaced by the tangential slip increment ($\boldsymbol{\delta}^{\text{t}}$). The outward unit normal vector $\mathbf{n}(\mathbf{u})$ leads to the definition of the relative tangential displacement increment as:

$$\boldsymbol{\delta}^{\text{t}} = [\mathbf{x}(\mathbf{u}) - \mathbf{x}^{\text{ref}}] - \{\lfloor \mathbf{x}(\mathbf{u}) - \mathbf{x}^{\text{ref}} \rfloor \cdot \mathbf{n}(\mathbf{u})\}\mathbf{n}(\mathbf{u}) \tag{2}$$

2.3. *Contact and friction laws*

The kinematic variables are essential in the formulation of the contact and friction laws, which also depend on the static variables. The normal and tangential components of the contact force are:

$$\boldsymbol{\lambda} = \lambda^{\mathrm{n}}\mathbf{n}(\mathbf{u}) + \boldsymbol{\lambda}^{\mathrm{t}} \tag{3}$$

The unilateral contact law is usually defined by an impenetrability condition, a compression condition, and a complementary condition, respectively:

$$d^{\mathrm{n}} \geqslant 0, \quad \lambda^{\mathrm{n}} \leqslant 0, \quad d^{\mathrm{n}}\lambda^{\mathrm{n}} = 0 \tag{4}$$

The Coulomb classic friction contact law is traditionally expressed by a slip rule, a friction criterion, and a complementary condition, respectively:

$$\dot{\boldsymbol{\delta}}^{\mathrm{t}} = |\dot{\boldsymbol{\delta}}^{\mathrm{t}}|\frac{\boldsymbol{\lambda}^{\mathrm{t}}}{|\boldsymbol{\lambda}^{\mathrm{t}}|}, \quad |\boldsymbol{\lambda}^{\mathrm{t}}| \leqslant -\mu\lambda^{\mathrm{n}}, \quad |\dot{\boldsymbol{\delta}}^{\mathrm{t}}|\lfloor|\boldsymbol{\lambda}^{\mathrm{t}}| + \mu\lambda^{\mathrm{n}}\rfloor = 0 \tag{5}$$

where μ is the friction coefficient.

3. MIXED FORMULATION

In the absence of contact and friction, the necessary conditions for a structural balance can be express with the following generic form:

$$\mathbf{F}^{\mathrm{int}}(\mathbf{u}) - \mathbf{F}^{\mathrm{ext}} = \mathbf{0} \tag{6}$$

$\mathbf{F}^{\mathrm{int}}(\mathbf{u})$ and $\mathbf{F}^{\mathrm{ext}}$ are the vectors of internal and external forces and $\mathbf{u}$ is the displacement field vector, solution of the problem. Coupling the contact with friction problem corresponds to constrain Equation (6) with the boundary conditions imposed by the contact with friction. To obtain a system of equations is necessary to eliminate the ambiguous character of the contact and friction relations (Equations (4) and (5)). Therefore, an augmented multiplier is introduced:

$$\begin{aligned} \boldsymbol{\sigma}(\mathbf{u},\boldsymbol{\lambda}) &= \sigma^{\mathrm{n}}(\mathbf{u},\boldsymbol{\lambda})\mathbf{n}(\mathbf{u}) + \boldsymbol{\sigma}^{\mathrm{t}}(\mathbf{u},\boldsymbol{\lambda}) \\ &= (\lambda^{\mathrm{n}}(\mathbf{u},\boldsymbol{\lambda}) + rd^{\mathrm{n}}(\mathbf{u}))\mathbf{n}(\mathbf{u}) + (\boldsymbol{\lambda}^{\mathrm{t}}(\mathbf{u},\boldsymbol{\lambda}) + r\boldsymbol{\delta}^{\mathrm{t}}(\mathbf{u})) \end{aligned} \tag{7}$$

where r is the penalty coefficient. The contact and friction conditions are computed based on a linear combination of the kinematic and static variables [5, 6]. The use of an augmented multiplier allows the formulation of an exactly equivalent problem for all strictly positive value of the penalty parameter r, which can generically be written as follows [5]:

$$\begin{Bmatrix} \mathbf{F}^{\mathrm{int}}(\mathbf{u}) + \mathbf{F}^{\mathrm{equi}}(\mathbf{u},\boldsymbol{\lambda}) \\ \mathbf{F}^{\mathrm{sup\,p}}(\mathbf{u},\boldsymbol{\lambda}) \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}^{\mathrm{ext}} \\ \mathbf{0} \end{Bmatrix} \tag{8}$$

The non-linear system of equations (8) corresponds, for each node belonging to Γ^{def}, to a system of six equations with six unknowns, the displacements $\mathbf{u}$ and the contact forces $\boldsymbol{\lambda}$. The sub-operators introduced represent the constraints imposed by the obstacle in the equilibrium, $\mathbf{F}^{\mathrm{equi}}(\mathbf{u},\boldsymbol{\lambda})$, and the supplementary equations necessary to evaluate the contact forces, $\mathbf{F}^{\sup \mathrm{p}}(\mathbf{u}, \boldsymbol{\lambda})$, defined as follows:

$$\begin{aligned}
\mathbf{F}^{\mathrm{equi}}(\mathbf{u}, \boldsymbol{\lambda}) &= \mathrm{proj}_{\Re^-}\{\sigma^{\mathrm{n}}(\mathbf{u}, \boldsymbol{\lambda})\}\mathbf{n}(\mathbf{u}) + \mathrm{proj}_{C^{\mathrm{aug}}(\lambda^{\mathrm{n}} + r d^{\mathrm{n}}(\mathbf{u}))}\{\boldsymbol{\sigma}^{\mathrm{t}}(\mathbf{u}, \boldsymbol{\lambda})\} \\
\mathbf{F}^{\sup \mathrm{p}}(\mathbf{u}, \boldsymbol{\lambda}) &= -\frac{1}{r}\,[\lambda^{\mathrm{n}}(\mathbf{u}, \boldsymbol{\lambda}) - \mathrm{proj}_{\Re^-}\{\sigma^{\mathrm{n}}(\mathbf{u}, \boldsymbol{\lambda})\}]\mathbf{n}(\mathbf{u}) \\
&\quad - \frac{1}{r}[\boldsymbol{\lambda}^{\mathrm{t}}(\mathbf{u}, \boldsymbol{\lambda}) - \mathrm{proj}_{C^{\mathrm{aug}}(\lambda^{\mathrm{n}} + r d^{\mathrm{n}}(\mathbf{u}))}\{\boldsymbol{\sigma}^{\mathrm{t}}(\mathbf{u}, \boldsymbol{\lambda})\}]
\end{aligned} \tag{9}$$

where the classical convex that results from the Coulomb friction law [11] is replaced by an augmented one, defined by

$$C^{\mathrm{aug}}(\lambda^{\mathrm{n}} + r d^{\mathrm{n}}(\mathbf{u})) = C\{\mathrm{proj}_{\Re^-}(\lambda^{\mathrm{n}} + r d^{\mathrm{n}}(\mathbf{u}))\} \tag{10}$$

The sub-operators depend directly on the kinematic and static contact variables, conducting to different definitions according with the contact status. Owing to the use of the augmented multiplier the contact status of each node is determined by a linear combination of both kinematic and static variables. A node is considered in contact only if a negative normal component of the augmented multiplier, $\sigma^{\mathrm{n}}(\mathbf{u}, \boldsymbol{\lambda})$, is detected; otherwise, the 'gap' status is assigned. For a node in contact it is necessary to evaluate the magnitude of the tangential component of the augmented multiplier $\boldsymbol{\sigma}^{\mathrm{t}}(\mathbf{u}, \boldsymbol{\lambda})$. If $|\boldsymbol{\sigma}^{\mathrm{t}}(\mathbf{u}, \boldsymbol{\lambda})| + \mu\sigma^{\mathrm{n}} < 0$ the 'stick' status is assigned to the node, otherwise it will correspond to a 'slip' node. Therefore, the expressions for the sub-operators can be derived from (9) as function of the augmented multiplier [10]:

$$\sigma^{\mathrm{n}} \geqslant 0 \Rightarrow \left\{\begin{matrix} \mathbf{F}^{\mathrm{equi}}(\mathbf{u}, \boldsymbol{\lambda}) \\ \mathbf{F}^{\sup \mathrm{p}}(\mathbf{u}, \boldsymbol{\lambda}) \end{matrix}\right\}\Bigg|\Bigg|_{\mathrm{gap}} = \left\{\begin{matrix} \mathbf{0} \\ -\frac{1}{r}\boldsymbol{\lambda} \end{matrix}\right\} \tag{11}$$

$$|\boldsymbol{\sigma}^{\mathrm{t}}| + \mu\sigma^{\mathrm{n}} < 0 \Rightarrow \left\{\begin{matrix} \mathbf{F}^{\mathrm{equi}}(\mathbf{u}, \boldsymbol{\lambda}) \\ \mathbf{F}^{\sup \mathrm{p}}(\mathbf{u}, \boldsymbol{\lambda}) \end{matrix}\right\}\Bigg|\Bigg|_{\mathrm{stick}} = \left\{\begin{matrix} \sigma^{\mathrm{n}}(\mathbf{u}, \boldsymbol{\lambda})\mathbf{n}(\mathbf{u}) + \boldsymbol{\sigma}^{\mathrm{t}} \\ d^{\mathrm{n}}(\mathbf{u})\mathbf{n}(\mathbf{u}) + \boldsymbol{\delta}^{\mathrm{t}}(\mathbf{u}) \end{matrix}\right\} \tag{12}$$

$$|\boldsymbol{\sigma}^{\mathrm{t}}| + \mu\sigma^{\mathrm{n}} \geqslant 0 \Rightarrow \left\{\begin{matrix} \mathbf{F}^{\mathrm{equi}}(\mathbf{u}, \boldsymbol{\lambda}) \\ \mathbf{F}^{\sup \mathrm{p}}(\mathbf{u}, \boldsymbol{\lambda}) \end{matrix}\right\}\Bigg|\Bigg|_{\mathrm{slip}} = \left\{\begin{matrix} \sigma^{\mathrm{n}}(\mathbf{u}, \boldsymbol{\lambda})[\mathbf{n}(\mathbf{u}) - \mu\,\mathbf{t}(\mathbf{u}, \boldsymbol{\lambda})] \\ d^{\mathrm{n}}(\mathbf{u})\mathbf{n}(\mathbf{u}) - \frac{1}{r}[\boldsymbol{\lambda}^{\mathrm{t}}(\mathbf{u}) + \mu\sigma^{\mathrm{n}}(\mathbf{u}, \boldsymbol{\lambda})\mathbf{t}(\mathbf{u}, \boldsymbol{\lambda})] \end{matrix}\right\} \tag{13}$$

where $\mathbf{t}(\mathbf{u}, \boldsymbol{\lambda})$ is the tangential slip direction unit vector defined by

$$\mathbf{t}(\mathbf{u}, \boldsymbol{\lambda}) = \boldsymbol{\sigma}^{\mathrm{t}}(\mathbf{u}, \boldsymbol{\lambda}) / \left|\boldsymbol{\sigma}^{\mathrm{t}}(\mathbf{u}, \boldsymbol{\lambda})\right| \tag{14}$$

A correct prediction of the contact surface and status is of fundamental importance since the sub-operators depend on the local contact frame through $\mathbf{n}(\mathbf{u})$ [10]. This dependency is even more evident in the elemental contact Jacobian matrix, which has to be computed in order to use the Newton–Raphson method to solve the global non-linear problem (Equations (8)):

$$\mathbf{J}(\mathbf{u}, \boldsymbol{\lambda}) = \begin{bmatrix} [\nabla_{\mathbf{u}}\mathbf{F}^{\mathrm{equi}}(\mathbf{u}, \boldsymbol{\lambda})] & [\nabla_{\boldsymbol{\lambda}}\mathbf{F}^{\mathrm{equi}}(\mathbf{u}, \boldsymbol{\lambda})] \\ [\nabla_{\mathbf{u}}\mathbf{F}^{\sup \mathrm{p}}(\mathbf{u}, \boldsymbol{\lambda})] & [\nabla_{\boldsymbol{\lambda}}\mathbf{F}^{\sup \mathrm{p}}(\mathbf{u}, \boldsymbol{\lambda})] \end{bmatrix} \tag{15}$$

Deriving Equation (11) for the case of a 'gap' node, the Jacobian matrix takes the form:

$$\mathbf{J}^{\text{gap}} = \begin{bmatrix} [0_{ij}] & [0_{ij}] \\ [0_{ij}] & [-I_{ij}/r] \end{bmatrix}, \quad i,j=1,2,3 \tag{16}$$

where **I** is the identity second-order tensor. Taking into account Equation (12), the elementary contributions for a node in contact with 'stick' status are evaluated as

$$\mathbf{J}^{\text{stick}} = \begin{bmatrix} [P_{ij} + C_{ij}] & [I_{ij}] \\ [(P_{ij} + C^{*}_{ij})/r] & [0_{ij}] \end{bmatrix}, \quad i,j=1,2,3 \tag{17}$$

Unlike **C** and **C***, **P** does not depend on the gradient of the normal to the rigid body:

$$P_{ij} = r\lfloor I_{ij} - n_i n_j + n_i \nabla_{u_j} d^{\text{n}} \rfloor, \quad C_{ij} = -r\lfloor (\delta^{\text{t}}_k + d^{\text{n}} n_k)\nabla_{u_j} n_k n_i \rfloor$$

and

$$C^{*}_{ij} = -r\lfloor \delta^{\text{t}}_k n_i \nabla_{u_j} n_k + \delta_p n_p \nabla_{u_j} n_i \rfloor, \quad i,j,k,p=1,2,3 \tag{18}$$

Analogously, deriving Equation (13), for the 'slip' case, the Jacobian matrix is given as

$$\mathbf{J}^{\text{slip}} = \begin{bmatrix} [rM_{ij} + N_{ik}\nabla_{u_j} n_k] & [E_{ij}] \\ [M_{ij} + N_{ik}\nabla_{u_j} n_k/r] & [E_{ij}/r - I_{ij}] \end{bmatrix}, \quad i,j,k=1,2,3 \tag{19}$$

In this case is possible to couple all the terms that depend on the gradient of the curvature in the second-order tensor, **N**, defined as

$$N_{ij} = \lambda_k[n_i - \mu t_i] + \sigma^{\text{n}} I_{ik} + \rho[(\lambda_k + r\delta_k)n_i - (\lambda_p + r\delta_p)n_p(t_i t_k - I_{ik})], \quad i,j,k,p=1,2,3 \tag{20}$$

M and **E** are given by the following expressions:

$$\begin{aligned} M_{ij} &= [n_i - \mu t_i]\nabla_{u_j} d^{\text{n}} + \rho[I_{ij} - n_i n_j - t_i t_j] \\ E_{ij} &= [n_i - \mu t_i]n_j + \rho\lfloor I_{ij} - n_i n_j - t_i t_j \rfloor, \quad i,j=1,2,3 \end{aligned} \tag{21}$$

with $\rho = -\mu\sigma^{\text{n}}/|\boldsymbol{\sigma}^{\text{t}}|$. In 2D applications, these terms disappear since $\lfloor I_{ij} - n_i n_j - t_i t_j \rfloor = \mathbf{0}$ [5].

The definitions of the sub-operators $\mathbf{F}^{\text{equi}}(\mathbf{u}, \boldsymbol{\lambda})$ and $\mathbf{F}^{\text{sup p}}(\mathbf{u}, \boldsymbol{\lambda})$ presented in Equation (9) are independent of the geometrical description of the tools [9, 10]. The generality of the numerical simulation codes use a finite element mesh description of the rigid body. This discretization of the forming tools clearly leads to problems in the contact treatment, and it is necessary to use reliable algorithms aim of which is to introduce the necessary continuity for general quadratic convergence [12–14]. In the following sections is assumed that the forming tools are modelled by parametric Bézier surfaces [15, 16]. The use of parametric Bézier surfaces to describe the rigid body simplifies the contact search algorithms as long as the necessary continuity between adjacent patches is guaranteed [17, 18]. Besides, the patches surfaces are naturally smooth.

The contact detection algorithm must find the correct patch where each node of the deformable body establishes contact with the tools. Function of the complexity of the process to be simulated, a high number of patches may be needed to model each tool. In these cases one needs to test, for each node, all the surface patches that define the each tool. This increases the difficulty in the contact detection algorithm. However, it is possible to associate to each node of the deformable body a reduced number of surface patches, based on simple geometrical considerations. To perform this, a global contact search algorithm should be used at each time increment.

4. GLOBAL AND LOCAL CONTACT SEARCH

In terms of global contact search, the first geometrical consideration that can be performed is to associate to each node α of the deformable body a predefined set of tools. The organization of these sets of tools is based on the orientation of each surface patch outward normal vector to the blank sheet. After this first geometrical selection, let us suppose that the set of candidate tools to contact with node α contains N patches. This number is highly dependent on the complexity of the geometry of the forming tools. So, it is desirable to reduce this number of candidates to $\bar{N} < N$ patches. This strategy is adopted in order to carry out the local contact search algorithms of impact and projection only on the $\bar{N}$ previously selected patches, during the time increment [7, 9]. In this work, two different global contact search algorithms for the selection of the candidate patches to contact will be presented and compared.

4.1. Global contact search – 1st algorithm

The group of candidate patches to establish contact with node α is reduced according to the distance between the co-ordinates of the node ($\mathbf{x}^\alpha$) and the middle point of each patch. For each of the N patches of the set, the middle point is calculated according to the total displacement of the tool containing the patch, $\mathbf{u}^{\text{tool}}|_{\text{S}}$ in the loading step S:

$$P_k^{\text{mid}}(\beta,\eta) = P_k^{\text{mid}}\left(\frac{1}{2},\frac{1}{2}\right) = \sum_{i=0}^{m}\sum_{j=0}^{n} b_{ijk}\left(\frac{1}{2}\right)^i\left(\frac{1}{2}\right)^j + u_k^{\text{tool}}|_{\text{S}}, \quad k = 1,2,3 \tag{22}$$

The patch is defined as a function of the curvilinear coordinates $(\beta,\eta) \in [0,1]$ and the corresponding polynomial coefficients b_{ijk}, with m and n being the degrees of the polynomial functions in β and η, respectively.

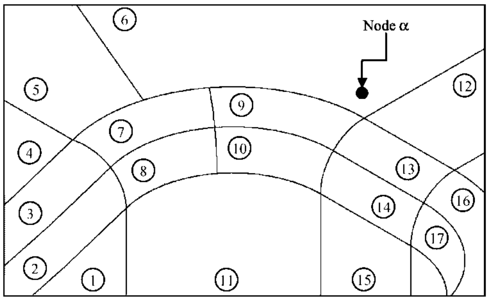
The distance between node α, of the deformable body, and the middle point of each surface patch is given by

$$\Delta D^{\text{mid}} = [(P_k^{\text{mid}} - x_k^\alpha)(P_k^{\text{mid}} - x_k^\alpha)]^{1/2}, \quad k = 1,2,3 \tag{23}$$

The algorithm will only retain the $\bar{N}$ surfaces closest to the node α, ($\bar{N} \leqslant N$). The number $\bar{N}$ of surfaces to be retained is also important. If the tools are modelled with surfaces of simple geometry and similar dimensions, $\bar{N}$ can be set equal to 4, what still guarantees that the surface that will result in the correct reference vector will be correctly selected. If the geometry is more complex, the node can be near many surfaces, even of different tools. That is why it is more suitable to increase $\bar{N}$ to a higher value (Example with $\bar{N} = 9$ in Reference [9]).

4.2. Global contact search – 2nd algorithm

In the case of tools of complex geometry defined by many surfaces with different dimensions, a similar algorithm that uses not only the co-ordinates of the middle point of the patch, but



$\bar{N}$ Patches	
1st algorithm	2nd algorithm
9	6
13	9
10	12
14	13
12	10
7	14
16	7
8	8
17	11

Figure 2. Selected patches for the global contact search strategies with the first and second algorithms.

also the co-ordinates of each vertex may determine the $\bar{N}$ group of candidates. In this case it is necessary to calculate the position of the middle point (Equation (22)) and of each vertex, as follows:

$$\left.\begin{aligned} P_k^1(\beta,\eta) &= P_k^1(0,0) = \sum_{i=0}^{m}\sum_{j=0}^{n} b_{ijk}0^i0^j + u_k^{\text{tool}}|_{\text{S}} \\ P_k^2(\beta,\eta) &= P_k^2(0,1) = \sum_{i=0}^{m}\sum_{j=0}^{n} b_{ijk}0^i1^j + u_k^{\text{tool}}|_{\text{S}} \\ P_k^3(\beta,\eta) &= P_k^3(1,0) = \sum_{i=0}^{m}\sum_{j=0}^{n} b_{ijk}1^i0^j + u_k^{\text{tool}}|_{\text{S}} \\ P_k^4(\beta,\eta) &= P_k^4(1,1) = \sum_{i=0}^{m}\sum_{j=0}^{n} b_{ijk}1^i1^j + u_k^{\text{tool}}|_{\text{S}} \end{aligned}\right\} \quad k=1,2,3 \tag{24}$$

The distance between the middle point and the node α of the deformable body is calculated from Equation (23). The distance between each vertex and the node α is calculated as

$$\Delta D^q = [(P_k^q - x_k^\alpha)(P_k^q - x_k^\alpha)]^{1/2}, \quad k=1,2,3 \quad q=1,2,3,4 \tag{25}$$

The minimum distance between node α and the four vertexes and the middle point of the patch ($\min(\Delta D^q, \Delta D^{\text{mid}})$) will be retained. After testing the N candidate patches of the set, the algorithm retains only the $\bar{N}$ patches closest to node α, as in the previous case.

Figure 2 presents an example of a complex geometry, defined with more than 17 patches, all with different dimensions. Both global contact search algorithms were performed in order to determine the $\bar{N}$ closest patches to node α. The table in Figure 2 presents the sequence of the nine closest patches to node α, determined with both algorithms. Although the patch that results in a correct reference position is the number six, this patch will never belong to the candidate group $\bar{N}$ if the selection algorithm is based only in the distance to the middle point of the patch. This is caused by the differences in area between this patch and its neighbours.

4.3. Local contact search – impact algorithm

For each of the $\bar{N}$ pre-selected patches candidates to establish contact with node α it is necessary to determine the associated reference vector in order to choose the correct patch. This is performed with an impact algorithm, based on the intersection of displacement trajectories of the patch and the node.

The impact algorithm is based on the total displacement of the tool, in the loading step S, $\mathbf{u}^{\mathrm{tool}}|_{\mathrm{S}}$ and the displacement of the node $\alpha(\mathbf{u}^{\alpha}|_{\mathrm{S}})$. The relative displacement is defined as $\Delta U_k^{\mathrm{rel}} = u_k^{\mathrm{tool}}|_{\mathrm{S}} - u_k^{\alpha}|_{\mathrm{S}}$. An additional parameter T is introduced in order to adjust the relative displacement $\Delta U_k^{\mathrm{rel}}$, such that an impact between the node and the obstacle can be established. The co-ordinates $(\bar{\beta}, \bar{\eta}, \bar{T})$ define the position of the impact point on the patch, and are solution of the equation:

$$F_k^{\mathrm{Im\,p}}(\bar{\beta}, \bar{\eta}, \bar{T}) = x_k^{\alpha} - \sum_{i=0}^{m} \sum_{j=0}^{n} b_{ijk} \bar{\beta}^i \bar{\eta}^j - \bar{T} \Delta U_k^{\mathrm{rel}} = 0_k, \quad k = 1,2,3, \;\; \bar{\beta}, \bar{\eta} \in [0,1], \;\; \bar{T} \in [\Re] \tag{26}$$

Equation (26) is non-linear in $(\bar{\beta}, \bar{\eta})$. To its solution a Newton–Raphson algorithm is applied leading to the following solution for iteration I:

$$\bar{a}_k^I = \bar{a}_k^{I-1} - \frac{F_j^{\mathrm{Im\,p}}|^{I-1}}{\bar{\nabla}_j F_k^{\mathrm{Im\,p}}|^{I-1}}, \quad j,k = 1,2,3 \tag{27}$$

where $\bar{\mathbf{a}}^{I-1} = [\bar{\beta}, \bar{\eta}, \bar{T}]$ is the solution vector at iteration $I - 1$, and $\bar{\nabla}_j F_k^{\mathrm{Im\,p}}$ is the Jacobian matrix of Equation (26).

The convergence criterion is based on the simultaneous satisfaction of the two following conditions:

$$\begin{cases} |\bar{a}_k^I - \bar{a}_k^{I-1}| \leqslant \varepsilon_a^{\mathrm{Conv.}} \\ |F_k^{\mathrm{Im\,p}}|^{I-1}| \leqslant \varepsilon_F^{\mathrm{Conv.}} \end{cases} \Rightarrow \text{Convergence in iteration } I \tag{28}$$

where $\varepsilon_a^{\mathrm{Conv.}}$ and $\varepsilon_F^{\mathrm{Conv.}}$ are some predefine limit values.

If convergence is attained within the maximum allowed number of iterations, then the reference vector is the one that corresponds to vector $\bar{\mathbf{a}} = [\bar{\beta}, \bar{\eta}, \bar{T}]$ with the minimum absolute value for T [7, 9]:

$$x_k^{\mathrm{ref}} = \sum_{i=0}^{m} \sum_{j=0}^{n} b_{ijk} \bar{\beta}^i \bar{\eta}^j + u_k^{\mathrm{tool}}|_{\mathrm{S}}, \quad k = 1,2,3 \tag{29}$$

If no impact solution is found for node α, the 'gap' status will be assigned for the current load step.

This impact algorithm has no solution if the patch belongs to a tool with no imposed displacement and if node α presents also null displacement. In this case it is necessary to perturb the relative displacement between the patch and the node, as follows:

$$\text{If} \quad \Delta U_k^{\mathrm{rel}} = 0_k \Rightarrow \Delta U_k^{\mathrm{rel}} = \varepsilon_k, \quad k = 1,2,3 \tag{30}$$

This perturbation vector ε dictates the convergence of the impact algorithm described earlier. The magnitude of the perturbation has no influence on the result. However, the direction of the perturbation vector determines the precision of the reference vector calculated by the impact algorithm [9].

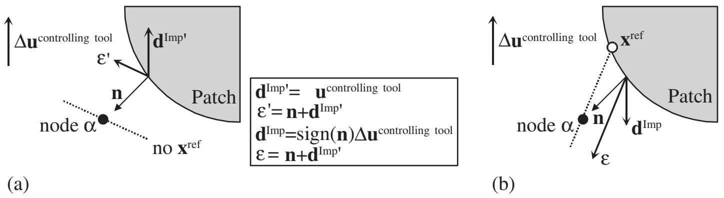


Figure 3. Influence of the direction of the perturbation vector in the solution for the impact algorithm: (a) no impact due to wrong perturbation vector; and (b) impact correctly calculated.

In order to improve the impact algorithm solution for this particular case, the perturbation vector must be calculated based on the 'expected impact direction', $\mathbf{d}^{\mathrm{Im\,p}}$. This vector is function of the direction of the displacement of the tool that controls the loading step, and of the outward normal of the candidate patch, as follows:

$$d_k^{\mathrm{Im\,p}} = \mathrm{sign}(n_k)\Delta u_k^{\mathrm{controlling\ \ tool}}, \quad k = 1,2,3 \tag{31}$$

The outward normal vector is not known (particularly in patches of high curvature) once the impact point in the patch is the unknown of algorithm. This normal is taken in the point of the patch considered as trial solution for the Newton–Raphson algorithm in Equation (27), and then the perturbation vector is calculated as follows:

$$\varepsilon_k = n_k + \mathrm{sign}(n_k)\Delta u_k^{\mathrm{controlling\ \ tool}}, \quad k = 1,2,3 \tag{32}$$

Figure 3 illustrates the influence of the sign of the outward normal vector on the impact direction vector and consequently on the direction of the perturbation vector.

4.4. Local contact search – projection algorithm

When a fully implicit algorithm is used to solve the global elastoplastic problem with contact and friction, during the equilibrium loop the node α must be projected on the candidate surface. This allows to define the implicit co-ordinate system associated with the node α and to update the reference vector $\mathbf{x}^{\mathrm{ref}}$. To perform this, one needs to determine for each candidate surface patch previously selected in the global contact search, the co-ordinates $(\hat{\beta},\hat{\eta})$ and $\hat{d}^{\mathrm{n}}$ such as

$$F_k^{\mathrm{Proj}}(\hat{\beta},\hat{\eta},\hat{d}^{\mathrm{n}}) = \sum_{i=0}^{m}\sum_{j=0}^{n} b_{ijk}\hat{\beta}^i\hat{\eta}^j + u_k^{\mathrm{tool}}|_{\mathrm{S}}$$

$$+ \hat{d}^{\mathrm{n}} n_k(\hat{\beta},\hat{\eta}) - x_k^{\alpha} = 0_k, \quad k = 1,2,3, \;\; \hat{\beta},\hat{\eta} \in [0,1], \;\; \hat{d}^{\mathrm{n}} \in [\Re] \tag{33}$$

As for Equation (26), the projection vector and the outward normal to the surface are non-linear functions of the parametric co-ordinates $(\hat{\beta},\hat{\eta})$; the system defined in the former equation is non-linear in the same variables. To its solution is necessary to resort to the Newton–Raphson iterative algorithm, which can be summarized, for iteration I, as follows:

$$\hat{a}_k^{\,I} = \hat{a}_k^{\,I-1} - \frac{F_j^{\text{Proj}}|^{I-1}}{\bar{\nabla}_j F_k^{\text{Proj}}|^{I-1}}, \quad j,k = 1,2,3 \tag{34}$$

where $\hat{\mathbf{a}}^{I-1} = [\hat{\beta}, \hat{\eta}, \hat{d}^{\text{n}}]$ presents the solution vector at iteration $I-1$, and $\bar{\nabla}_j F_k^{\text{Proj}}$ is the Jacobian matrix of Equation (33).

As in the case of the impact algorithm, the convergence criterion is based on the simultaneous satisfaction of the two following conditions:

$$\begin{cases} |\hat{a}_k^{\,I} - \hat{a}_k^{\,I-1}| \leqslant \varepsilon_a^{\text{Conv.}} \\ |F_k^{\text{Proj}}|^{I-1}| \leqslant \varepsilon_F^{\text{Conv.}} \end{cases} \Rightarrow \text{Convergence in iterations } I \tag{35}$$

where $\varepsilon_a^{\text{Conv.}}$ and $\varepsilon_F^{\text{Conv.}}$ are some predefine limit values.

In case of multiple solutions the algorithm selects the one with minimum normal distance d^{n}. In fact, this parameter represents warrantable geometrical measure to distinguish the various solutions [7, 9]. It is important to notice that, after convergence, the first two lines of $(\bar{\nabla}_j F_k^{\text{Proj}}|^{\text{Conv.}})^{-1}$ define the transformation matrix from the curvilinear system to the Cartesian system. This transformation matrix is required in the calculus of the gradient of the outward normal vector to the obstacle. The third line defines the gradient of the normal distance. Both gradients are needed to the elemental and global finite element ‘stiffness’ matrices (Equations (17) and (19)) [7, 9]). The quality and accuracy of the solution of this algorithm is crucial for the global convergence of the contact problem.

5. NUMERICAL EXAMPLE

In this section, the influence of the global contact search on the performance of the local contact search algorithms is evaluated.

The Newton–Raphson method used to solve the two-presented local contact search algorithms is known to present quadratic convergence rate in the vicinity of the solution. So, a good selection of the first trial solution is very important. When the global contact search is performed with the first algorithm, the trial solution used is the middle point of the patch, as it was this geometrical measure that allowed to assort the patches. The use of the second algorithm for the global contact search allows the choice of the trial solution between the middle point and one of the vertices of the patch. The use of the second algorithm also allows an improved prediction of the perturbation introduced in the Impact algorithm, in case of tools with no displacement. In fact, the outward normal vector used in the prediction of the impact direction vector is, in this case, the one corresponding to the nearest point of the patch selected between the middle point and the vertices, which can be very important in case of tools with complex geometries.

Previous works indicate that the impact and projection algorithms converge in almost any situation, with a number of iterations that are function of the patch curvature. The impact algorithm for plane patches normally uses two iterations, and for a patch with strong curvature it takes about six iterations. The projection algorithm, if the degree of the patch is high $(m, n > 4)$, may require between four and eight iterations [9].

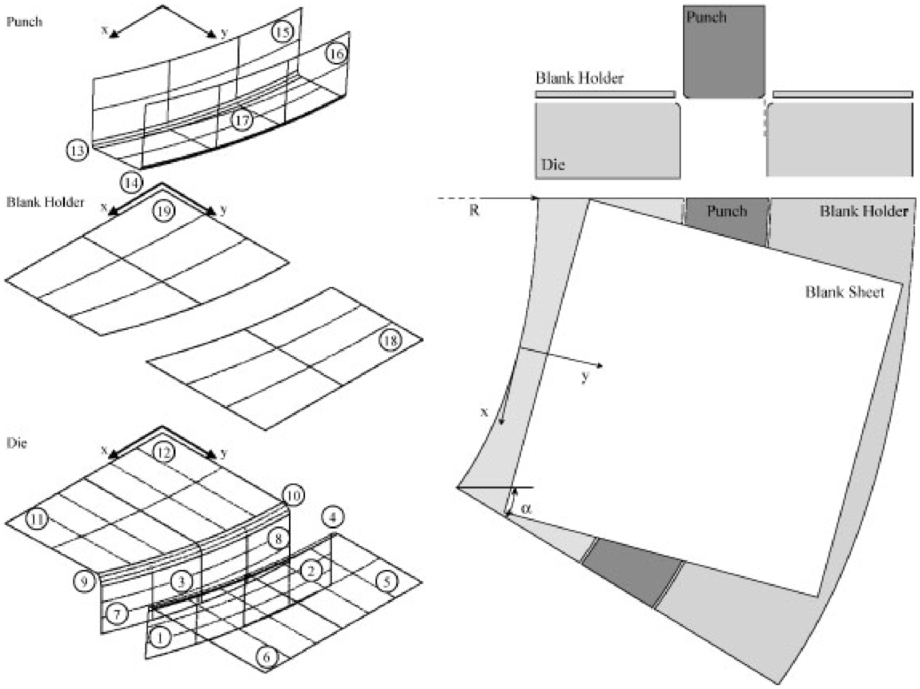


Figure 4. Forming tools geometry, patches definition and numeration for each tool.

Table I. Summary of the input data for the numerical simulation of a rail.

Initial sheet geometry	
Width/2	150 mm
Length	300 mm
Thickness	1 mm
Material data	
Young's modulus, Poisson ratio	211 000 MPa, 0.3
Isotropic hardening	$\sigma = 553.5\ (\varepsilon + 0.00163)^{0.227}$ (MPa)
Process parameters	
Friction coefficient	0.1
Blank holder constant force	90 000 N

To compare the presented global contact search algorithms, the simulation of the deep drawing of a rail was performed with the first algorithm (Equations (27) and (28)), and with the second algorithm (Equations (29) and (30)). The selected deep drawing problem is presented in Figure 4 [19]. The process conditions used are summarized in Table I.

This problem requires three different tools: the punch, the blank holder and the die, as presented in Figure 4. They were all defined with Bézier surfaces. Owing to symmetry

Table II. Summary of the numerical data used for the presented numerical simulations.

		Impact algorithm			Projection algorithm	
Maximum number of iterations		20			20	
$\varepsilon_a^{\text{Conv.}}$ (Equations (28) and (35))		1×10^{-10}			1×10^{-10}	
$\varepsilon_F^{\text{Conv.}}$ (Equations (28) and (35))		1×10^{-10}			1×10^{-10}	
Global contact search algorithm		1st	2nd		1st	2nd
Trial solution	$\bar{u}^0$	0.5	$u^{\text{closest vertex}}$	$\hat{u}^0$	0.5	$u^{\text{closest vertex}}$
	$\bar{v}^0$	0.5	$v^{\text{closest vertex}}$	$\hat{v}^0$	0.5	$v^{\text{closest vertex}}$
	T^0	0.0	0.0	d^{n^0}	0.0	0.0

conditions only one half of the tools were modelled using CAD software, resulting in twelve surfaces for the die, five surfaces for the punch and two for the blank holder. The degree of the patches that describe the punch and die shoulder is six in longitudinal direction and five in the radial direction. This is kept equal in adjacent patches in order to guarantee an equal number of characteristic points for the common patch curves assuring G_0 continuity [17]. The patches that define the die were cut along the middle plane and split in two, in order to evaluate the influence of the size of the patches in the impact and projection algorithms (Figure 4).

The numerical parameters defined for the local contact search algorithms are summarized in Table II. The blank sheet is modelled with a uniform coarse mesh of average element size of 10 millimetres in the plane, and two layers in thickness. This mesh is too coarse to evaluate process conditions. However, this discretization is sufficient for the evaluation of the behaviour of both global contact search algorithms.

Figure 5 shows the geometrical configuration obtained for both algorithms after a punch displacement of 10 mm. In this figure the tools are described by triangular elements only for visualisation post processing purposes. The difference between the contact zones in the punch predicted from the first and second algorithm of the global contact search is evident. In fact, it is possible to observe penetration between the punch and the blank sheet near the symmetry plane, when the global contact search is performed with the first algorithm. Such situation only occurs because a 'gap' status is assigned to the nodes with penetration after the local contact search algorithm. This incorrect prediction of the contact status is not observed when dealing with nodes contacting the die surfaces, in both algorithms. The incorrect solution results from an erroneous selection of the candidate patches performed by the first algorithm, since it does not occur for the contact with the punch predicted by the second algorithm. The first algorithm also seems to be more sensitive to the dimension of the patches, since no incorrect local contact search solution appears when the nodes establish contact with the die. One should remember that the patches that define the die were cut along the middle plane and split in two, so they are smaller in area than the punch patches.

In order to evaluate this penetration problem with more detail, two nodes contacting the punch were selected for analysis. The positions of the selected nodes (A and B) are presented in Figure 5, being A in the symmetry plane. The resume of the results obtained for the impact

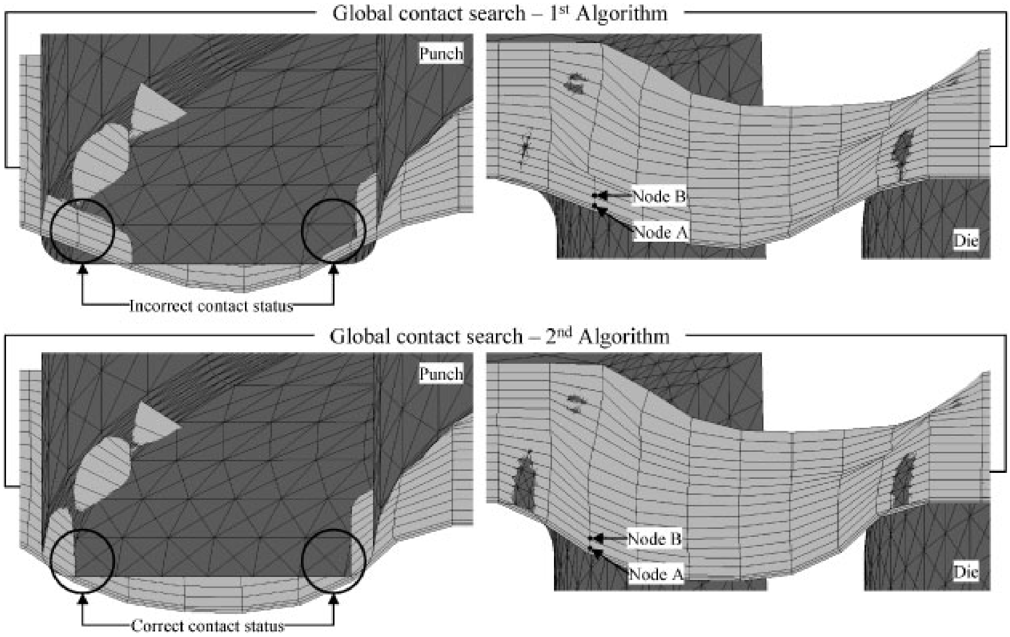


Figure 5. Contact areas with the punch and die after a punch displacement of 10 mm, with the two presented global contact search algorithms.

and projection algorithms are shown in Table III, but only for patch 13, which corresponds to the correct solution for the reference position. As mentioned earlier, the degree of this patch is six in the punch shoulder direction and five in the other direction. From Table III the influence of the trial solution in the solution and convergence rate of the impact and projection algorithms is clear.

In terms of the impact algorithm, the number of iterations necessary to assure the required precision is reduced by a factor of two when the second global contact search algorithm is used instead of the first algorithm. The projection algorithm is more sensitive to the trial solution. In fact, for a not so good trial solution the convergence rate decreases (Node A), or the algorithm can even diverge (Node B). An evaluation of the number of iterations required for the process convergence indicates a reduction factor of three, if the trial solution used is the closest vertex of the patch to the node of the deformable body. These results also confirm that the erroneous geometrical solution predicted by the code with the first algorithm results from an incorrect solution of the projection algorithm. In fact, during the convergence procedure Node A passes through solutions that are clearly out of the patch, and the implemented algorithm abandons that candidate patch before reaching iteration fourteen. Consequently, a 'gap' status is assigned to both nodes A and B, and so they are able to present those values of penetration.

In terms of CPU time, it is not possible to make any direct comparison since the solutions obtained with the distinct global contact search algorithms lead to different algorithm evolutions. Apparently, at least for the present case, the time spent on the selection of the closest

Table III. Reference vector calculation in patch 13 for nodes A and B. Convergence behavior of the local contact search algorithms (Impact and Projection) in function of the global contact search algorithm used.

	Impact algorithm				Projection algorithm			
	Node A		Node B		Node A		Node B	
Iteration	1st	2nd	1st	2nd	1st	2nd	1st	2nd
1	−1.982E + 00 5.234E − 03 −2.020E + 01	1.523E − 02 4.086E − 06 3.335E − 04	−1.663E + 00 4.721E − 02 −1.729E + 01	1.569E − 02 4.289E − 02 3.426E − 04	−8.728E − 01 5.234E − 02 −8.711E + 00	1.523E − 02 4.086E − 06 2.034E − 04	−7.131E − 01 4.721E − 02 −7.457E + 00	1.569E − 02 4.289E − 02 2.090E − 04
2	−1.503E + 00 6.269E − 05 1.676E + 01	1.524E − 02 4.256E − 06 2.384E − 03	−6.130E − 01 4.372E − 02 1.338E + 01	2.498E − 02 4.288E − 02 5.470E − 03	−3.650E + 00 8.833E − 05 −3.734E + 00	1.524E − 02 4.256E − 06 1.454E − 03	−3.097E + 00 4.277E − 02 −2.836E + 00	2.497E − 02 4.288E − 02 3.336E − 03
3	7.437E − 01 6.504E − 06 3.141E + 00	1.524E − 02 4.256E − 06 2.384E − 03	3.142E − 01 4.290E − 02 −6.144E + 00	2.498E − 02 4.288E − 02 6.344E − 03	−3.039E + 00 3.489E − 04 −2.920E + 01	1.524E − 02 4.256E − 06 1.454E − 03	−2.588E + 00 4.152E − 02 −3.793E + 00	2.496E − 02 4.288E − 02 3.867E − 03
4	−7.123E − 01 4.272E − 06 −1.226E + 01	**1.524E−02** **4.256E−06** **2.384E−03**	8.274E − 05 4.288E − 02 −9.380E − 01	**2.498E−02** **4.288E−02** **6.344E−03**	−2.528E + 00 −3.866E − 04 −2.107E + 00	**1.524E−02** **4.256E−06** **1.454E−03**	−2.169E + 00 4.245E − 02 6.521E + 00	2.496E − 02 4.288E − 02 3.867E − 03
5	5.311E − 01 4.262E − 06 −9.455E + 00		2.497E − 02 4.288E − 02 7.727E − 05		−2.122E + 00 2.160E − 04 7.171E + 00		−1.833E + 00 4.263E − 02 9.431E + 00	**2.496E−02** **4.288E−02** **3.867E−03**
6	−1.483E − 01 4.256E − 06 −3.789E + 00		2.498E − 02 4.288E − 02 6.344E − 03		−1.791E + 00 8.913E − 05 9.532E + 00		−1.193E + 00 4.322E − 02 9.500E + 00	
7	1.837E − 02 4.256E − 06 −2.735E − 01		2.498E − 02 4.288E − 02 6.344E − 03		−4.197E − 01 −3.840E − 04 9.324E + 00		−2.684E − 01 4.317E − 02 −5.112E + 00	

8	1.524E − 02 4.256E − 06 2.285E − 03	**2.498E−02** **4.288E−02** **6.344E−03**	−2.757E − 01 9.964E − 06 −1.105E + 00	−2.858E + 01 4.279E − 02 −5.154E − 01
9	1.524E − 02 4.256E − 06 2.384E − 03		8.361E − 02 4.220E − 06 −5.092E − 01	−2.377E + 01 1.341E − 01 1.209E + 06
10	**1.524E−02** **4.256E−06** **2.384E−03**		7.609E − 03 4.256E − 06 −2.736E − 02 (...)	−1.973E + 01 1.400E − 01 5.312E + 02 (...)
14			**1.524E−02** **4.256E−06** **1.454E−03**	−9.308E + 00 1.624E − 01 1.337E + 04 (...)
20				***−3.019E+00*** ***−1.744E+00*** ***−2.513E+01***

vertex of the patch to the contact node is counterbalanced by the reduction in the number of iterations of the Newton–Raphson procedure in local contact search algorithms. The difference in CPU time between the two simulations presented is less then 1.5%.

6. CONCLUSIONS

This paper proposes a new global contact search algorithm for complex 3D geometries modelled with parametric surface patches. For large-scale applications it is fundamental to reduce the number of candidate surfaces to contact with each finite element node of the deformable body. The selection of the closest patches to each node can be performed based on the distance between the node of the deformable body and the middle point of each patch. In complex geometries that require a larger number of patches with different dimensions, it is better to rely on the distance from the node to the middle point and to each vertex of the patch. If the selection of the candidate surface patches is performed based on the patch vertexes, it is also possible to store the information about the closest vertex to the node of the deformable node. This information can be used latter as trial solution for the Newton–Raphson algorithm, which is required for the determination of the reference vector and the implicit co-ordinate system. The comparison between the numerical behaviour of the impact and projection algorithms, with a trial solution either constant or updated in function of the nearest vertex to the node, shows that the latest converges faster and reduces the chances of divergence of the local contact search algorithm. The results also show that the number of iterations indicated in previous works may not be sufficient in case of surfaces presenting large dimensions, and if the trial solution for the Newton–Raphson algorithm is constant and equal to the position of the middle patch vector. However, if the trial solution is obtained from the closest vertex of the patch to the node of the deformable body the number of iterations referred is sufficient.

The proposed global contact search algorithm has contributed for an improvement of the global behaviour of the finite element code DD3IMP. It prevents situations with erroneous contact status associated to nodes of the deformable body that could conduct to convergence problems of the global simulation. Furthermore, the solution in each load step is improved due to the correct contact status. The increase of CPU time associated to the computation of each vertex and middle point of each candidate patch is compensated by the reduction in the number of iterations in the impact and projection algorithms.

ACKNOWLEDGEMENTS

The work presented here was funded by the European Community throughout the Growth Programme (contract G1RD-CT-2000-00104) and by the Portuguese Foundation for the Science and the Technology throughout the Programme POCTI (contract EME/35945/99). The authors are grateful for this support.

REFERENCES

1. Menezes LF, Thuiller S, Manach PY, Bouvier S. Influence of the work-hardening models on the numerical simulation of a reverse deep-drawing process. In *Plasticity*, *Damage and Fracture at Macro*, *Micro and Nanoscales*, *Proceedings of Plasticity'02: The 9th International Symposium on Plasticity and its Current Applications*, Khan AS. Oscar Lopez-Pamies (eds). Neat Press: Maryland USA, 2002; 331–333.

2. Menezes LF, Teodosiu C, Makinouchi A. 3-D solid elasto-plastic elements for simulating sheet metal forming processes by the finite element method. *FE simulation of 3D sheet metal forming process in automotive industry*; Tagungsbericht der VDI-Gesellschaft Fahrzeugtechnik. VDI-Verlag: Zurich, Germany, 1991; 381–403.
3. Menezes LF, Teodosiu C. Three-dimensional numerical simulation of the deep-drawing process using solid finite elements. *Journal of Materials Processing Technology* 2000; **97**:100–106.
4. Alves JL, Menezes LF. Application of tri-linear and tri-quadratic 3-D solid finite elements in sheet metal forming process simulation. In *Simulation of Material Processing*: *Theory*, *Methods and Applications*, *Proceedings of 7th International Conference NUMIFORM 2001*; Ken-ichiro Mori (eds). Toyohashi: Japan, 2001; 639–644.
5. Alart P, Curnier A. A Mixed formulation for frictional contact problems prone to Newton like solution methods. *Computer Methods in Applied Mechanics and Engineering* 1991; **92**(3):353–375.
6. Simo JC, Laursen TA. An augmented lagrangean treatment of contact problems involving friction. *Computers and Structures* 1992; **42**:97–116.
7. Menezes LF. Modelação tri-dimensional e simulação numérica dos processos de enformação por deformação plástica, aplicação á estampagem de chapas metálicas. *Ph.D. Thesis*, Universidade de Coimbra, Portugal, 1995.
8. Menezes LF, Teodosiu C. Improvement of the frictional contact treatment in a single loop iteration algorithm specific to deep-drawing simulations. In *Proceedings of the Conference NUMISHEET'99*; Gelin JC, Picart P (eds). Besançon: France, 1999; 197–202.
9. Heege A. Simulation numerique 3D du contact avec frottement et application à la mise en forme. *Ph.D. Thesis*, Institut National Polythecnique de Grenoble, France, 1993.
10. Heege A, Alart P. A frictional contact element for strongly curved contact elements. *International Journal for Numerical Methods in Engineering* 1996; **39**:165–184.
11. Moreau J.J. Application of convex analysis to some problems of dry friction. In *Trends of Pure Mathematics Applied to Mechanics*; Zorski (eds), vol. II. 1979.
12. Hallquist JO, Goudreau GL, Benson D. Sliding interfaces with contact-impact in large-scale Lagrangian computations. *Computer Methods in Applied Mechanics and Engineering* 1985; **51**(1–3):107–137.
13. Hallquist JO, Schweizerhof K, Stillman D. Efficiency refinements of contact strategies and algorithms in explicit F.E. programming, In *Computational Plasticity*: *Fundamentals and Applications*, Owen DRJ *et al.* (eds). Pineridge Press: Swansea, 1992.
14. Eterovic A, Bathe KJ. An interface interpolation scheme for quadratic convergence in the finite element analysis of contact problems. *Computational Methods in Non-linear Mechanics*, Wriggers P, Wagner W. (eds). Springer-Verlag: Berlin, 1991.
15. Bézier P. Essai de définition numérique des courbes et des surfaces. *Ph.D. Thesis*, Université Pierre et Marie Curie, 1977
16. Degen W. Explicit continuity conditions for adjacent Bézier surfaces patches. *Computer Aided Geometric Design* 1990; **7**:181–199.
17. Oliveira MC, Menezes LF. Optimização da descrição das ferramentas por superfícies de Bézier na simulação do processo de estampagem. In *Proceedings of the V Congresso Métodos Numéricos en Ingeniería*, Goicolea JM, Mota Soares C, Pastor M, Bugeda G (eds). Espanha, 2002 (CD-ROM edition).
18. Oliveira MC. Simulação numérica do processo de estampagem, optimização de um algoritmo implícito. *MSc. Thesis*, Universidade de Coimbra, Portugal, 1999.
19. Col A. First results of the 3DS research project. *Proceedings of the 22nd International Deep Drawing Research Group Congress and Working Group Meetings*, Nagoya, Japan, 2002.