

# Mortar Contact Algorithm for Implicit Stamping Analyses in LS-DYNA<sup>®</sup>

Thomas Borrvall  
Engineering Research Nordic AB  
Linköping, Sweden

## Abstract

*A challenging task for the static implicit nonlinear solver in LS-DYNA is to accurately and robustly solve contact problems, especially is this needed for stamping simulations. This paper aims at investigating the benefits of a mortar segment-to-segment contact algorithm by Puso and Laursen [1,2] when compared to the traditional node-to-segment approach. A penalty based version of the algorithm is implemented in LS-DYNA, meaning that the contact tractions are proportional to both the penetration as well as the overlapped area of segments in contact. This allows for the nice property that the resulting global contact force is continuous with respect to deformation and thus makes the approach intuitively suitable for implicit analyses. Further measures for smoothing the response are implemented in the method and the first tests indicate that the method is advantageous at least for a certain class of problems, but how great the overall impact will be remains to be seen.*

## Introduction

Mechanical contact is a common occurrence in real-life events and industrial applications, and hence its associated algorithms play an important role in computer simulation software tools such as LS-DYNA. Many years of academic research and industrial software development have lead to efficient and robust contact algorithms for successful use in simulating a wide range of complex processes. But in LS-DYNA in particular, the existing algorithms are first and foremost developed for explicit time integration where the mechanical configurations change very little between discretization steps, and thus allows for various efficiency optimizing measures. The explicit time integration also puts less requirements on the continuity properties of forces and moments compared to implicit integration, and is therefore well suited for simulating the non-smooth nature of contacts. In general however, explicit time integration requires mass inertia in the simulated system and is the preferred scheme for dynamical events such as crash, blast and penetration, but for quasi-static events such as forming and stamping the Courant stability criterion on the time step results in either very long simulation times or undesired spurious dynamic effects due to artificial mass scaling. Although the mass scaling approach has been widely used, and with much success too, it still requires great caution in order to not add too much dynamic effects that would deteriorate the results. Users tend to not want to play with this feature and worry about the influence of inertia, but instead convert to implicit analysis where this effect can be completely removed. A potential advantage herein lies in the fact that implicit integration is unconditionally stable, thus the step size is restricted only by convergence and accuracy considerations. The main disadvantage is that the scheme requires the solution of a large linear system of equations typically several times per step, which makes each step computationally time consuming when compared to that of an explicit analysis. To benefit from

converting from explicit to implicit analysis, large enough steps for overcompensating the expense of the linear system solves are required, but this in turn requires accurate and smooth algorithms accompanied with a good representation of the tangent stiffness. Herein lies a great challenge regarding contact algorithms, to obtain a formulation that is well suited for implicit analysis all in terms of speed, robustness and accuracy.

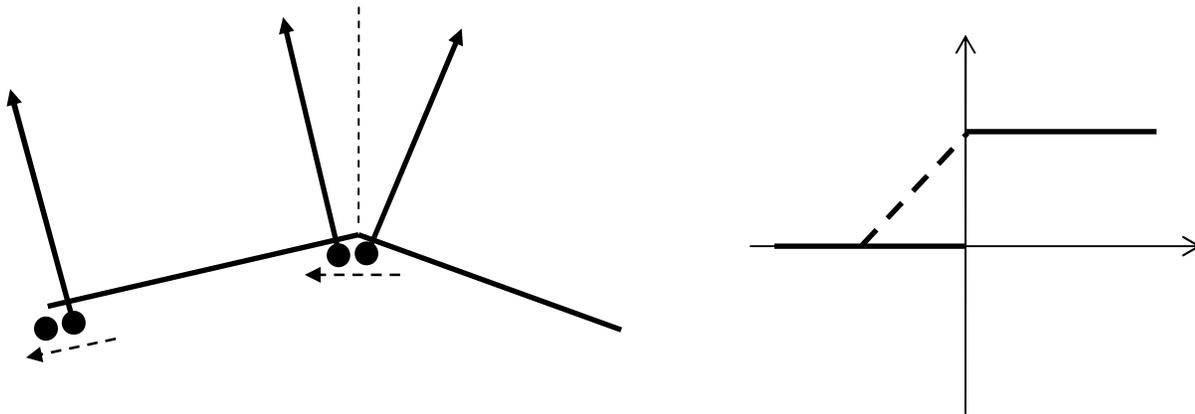


Figure 1 Illustration of contact force jumps in node to surface contact (left) and contact stiffness as function of penetration for IGAP=1 (dashed line) and IGAP=2 (solid line).

In LS-DYNA, the available contact algorithms are penalty based and the best suited for implicit stamping analysis is (in the opinion of the author) the forming surface to surface contact. This is a double sided node to surface contact, meaning that the nodes on the slave side may not penetrate the segments on the master side and vice versa. Nevertheless, a contact algorithm based on nodes penetrating segments will yield a force that is discontinuous with respect to motion, for instance when nodes slide over adjacent segments that are not co-planar or when a node slides off the edge of a segment, see **Figure 1**. The practical consequence of this is difficult to say, the effect should be marginal for fine enough discretization, but it is still not a nice property when considering an implicit solution procedure. What would be best is if both force and stiffness were continuous functions, but this is difficult to accomplish for a penalty based contact algorithm. The contact stiffness for a node typically jumps from zero to a high penalty value when going in to contact and drops back to zero when going out of contact, this is what is usually causing convergence difficulties as nodes tends to oscillate in and out of contact. An attempt to remedy this has been made to all contacts by ramping the penalty stiffness from zero up to the value at contact during a finite interval in the vicinity of the contact surface, this is illustrated in **Figure 1**. This has a positive effect on convergence, if just looking at the number of iterations, but is negative for accuracy. Once a slave node is in that vicinity of a master segment, this node will tend to stick at that very distance from the segment and not going either in to contact or out of contact. The effects of this will be illustrated in the examples section.

The mortar contact algorithm [1,2] is a penalty based segment to segment contact where the contact tractions are proportional to the penetration of segments as well as their overlapped area. This means that there will be a continuous transition of forces when a slave segment slides across adjacent master segments or when it slides off the edge of a master segment, and this is an attractive property for implicit analysis. Although the effect may not be obvious in practice, a quadratic form of the penalty function is used that makes the tangent stiffness continuous. The

mortar contact algorithm is presented in more detail in the next section, starting with a short description of its usage. The paper ends with some examples highlighting the pros and cons of the new method compared to the traditional forming surface to surface contact and some concluding remarks.

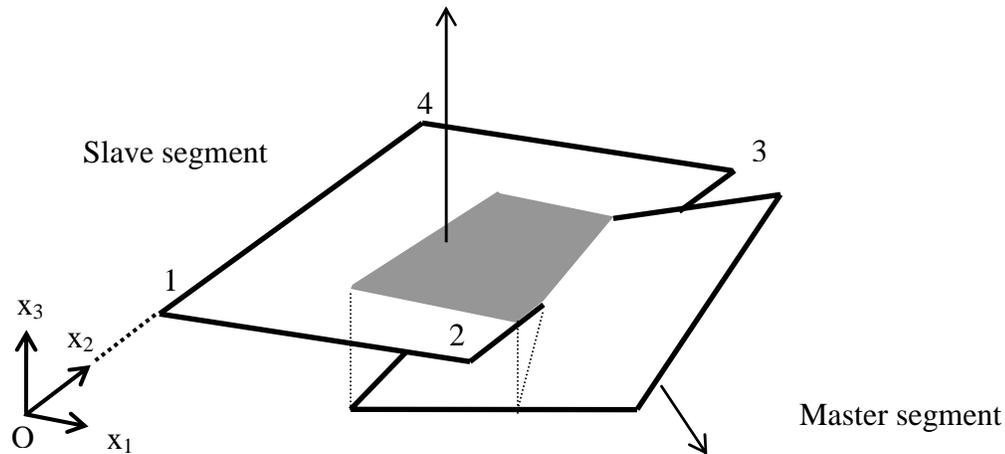


Figure 2 Mortar segment-to-segment based contact. The shadowed surface illustrates the overlapped area and the space between the segments indicated by dashed lines is the penetrated volume.

### The mortar contact algorithm

The mortar contact is assumed to occur between two sets of triangular and/or quadrilateral segments, one denoted slave set and the other master set. The slave set is for the time being restricted to consist of shell elements only, this is a consequence of the algorithm being first and foremost intended for forming and stamping applications. The common situation is that the slave set is the collection of segments representing the surface of the blank in a forming/stamping simulation. The master side can consist of either solid or shell segments, and typically represents the surface of one of the rigid tools operating on the blank. If the master side consists of shells, the user must explicitly orient the mesh such that the shell normals point towards the blank, because this is how the algorithm is set up to work. The orientation of the slave side is arbitrary but contact can occur on one side only, LS-DYNA will automatically detect on which side the master segment set is located and take the appropriate actions. This means that several contact interfaces must be created for a single application, but this is the general procedure for these types of problems anyway and should not be an issue. A contact situation is illustrated in **Figure 2**.

The mortar contact method is implemented in LS-DYNA v.971 R4 and supports the more important contact options but some remains to be implemented. It is currently available in SMP and the MPP implementation is undertaken. It is activated in the keyword input deck by simply appending the optional suffix MORTAR to the keyword

\*CONTACT\_FORMING\_SURFACE\_TO\_SURFACE, i.e.,  
 \*CONTACT\_FORMING\_SURFACE\_TO\_SURFACE\_MORTAR, and takes the same input parameters as the traditional forming surface to surface contact. We now turn to its implementation and theoretical description.

The hierarchical contact search starts with a bucket sort, where the slave segments are gathered into buckets to make the remainder of the contact search more efficient. All buckets are of the same dimensions, aligned along the global axes, and just large enough so that no slave segment (including thickness offsets and displacements during step) can intersect more than  $2 \times 2 \times 2 = 8$  buckets. Then a sequence of refined contact searches takes place to find pairs of slave and master segments that are such that the master segment is in the shadow of the slave segment. This is to say that the master segment should intersect the infinite extrusion of the slave segment in the direction of its normal, and this defines the potential contact state. Contact occurs when parts of the slave and master segments penetrate as shown in **Figure 2** and the penalty force introduced to separate them is proportional to the penetrated volume. The details are as follows.

We assume that the slave and master segments are quadrilaterals, the generalization to triangles is more or less trivial. Furthermore, sum over repeated indices is implicitly assumed unless otherwise stated. Without loss of generality we assume that the segments are plane, a warped segments is made plane by projecting its nodes onto its mean plane.

Let  $x_{ii}^S$  and  $x_{ii}^M$  denote the  $i$ th ( $i=1,2,3$ ) coordinate of node  $I$  ( $I=1,2,3,4$ ) of the slave and master segment, respectively. The slave and master segments are parameterized using bilinear shape functions as

$$x_i^S = x_{ii}^S N_I(\xi, \eta) \quad x_i^M = x_{ii}^M N_I(\xi, \eta)$$

where  $(\xi, \eta)$  belongs to the isoparametric domain

$$\Pi = \{ (\xi, \eta) : -1 \leq \xi \leq 1, -1 \leq \eta \leq 1 \}$$

and

$$N_1(\xi, \eta) = \frac{1}{4}(1 - \xi)(1 - \eta) \quad N_2(\xi, \eta) = \frac{1}{4}(1 + \xi)(1 - \eta)$$

$$N_3(\xi, \eta) = \frac{1}{4}(1 + \xi)(1 + \eta) \quad N_4(\xi, \eta) = \frac{1}{4}(1 - \xi)(1 + \eta)$$

The node numbering convention is shown in **Figure 2**. We let  $\tilde{\Pi} \subset \Pi$  be the set of points  $(\tilde{\xi}, \tilde{\eta})$  for which there exists a projection from  $\tilde{x}_i^S = x_{ii}^S N_I(\tilde{\xi}, \tilde{\eta})$  on the slave segment along the slave segment normal denoted  $n_i^S$  onto a corresponding

$$\bar{x}_i^M = x_{ii}^M N_I(\bar{\xi}, \bar{\eta})$$

on the master segment. The domain  $\tilde{\Pi}$  is further reduced to  $\tilde{\Pi}_+$  by intersecting it with the points for which  $(\tilde{x}_i^S - \bar{x}_i^M)n_i^S \geq 0$ , that is, we only consider slave segment points that have penetrated the master segment. Let the domain  $A_+$  be the image of  $\tilde{\Pi}_+$  on the slave segment, this is a polygon that in practice is obtained using a clipping algorithm and represents the overlapped area of the two segments in contact and is illustrated in grey in **Figure 2**.

We define the penetrated volume of the segment pair  $(S, M)$  as

$$d_{(S,M)} = \int_{A_+} n_i^S (\tilde{x}_i^S - \bar{x}_i^M) dA = n_i^S (x_{il}^S c_I^{SM} - x_{il}^M c_I^{MS})$$

where

$$c_I^{SM} = \int_{A_+} N_I(\tilde{\xi}, \tilde{\eta}) dA \approx |A_+| N_I(\tilde{\xi}_C, \tilde{\eta}_C)$$

$$c_I^{MS} = \int_{A_+} N_I(\bar{\xi}, \bar{\eta}) dA \approx |A_+| N_I(\bar{\xi}_C, \bar{\eta}_C)$$

and subscript  $C$  indicates the coordinate in  $\tilde{\Pi}_+$  that is associated with the centroid of  $A_+$ . With this notation we define the accumulated sliding as

$$s_S^j = s_S^j(t_-)_+ + {}_s t_i^j(t_-) \sum_M (x_{il}^M(t_-) c_I^{MS} - x_{il}^S(t_-) c_I^{SM})$$

where  ${}_s t_i^j$  refers to the tangential basis of the slave segment ( $j=1,2$ ) and the argument  $t$  means that the evaluation of the corresponding parameter is from the previous implicit step. The accumulated penetration is defined as

$$d_S = \sum_M d_{(S,M)}.$$

The contact pressure is given as a function of the penetrated volume

$$p_{(S,M)} = K_S \varepsilon \varepsilon_p g_d \left( \frac{d_{(S,M)}}{\varepsilon T_S A_S} \right)$$

where  $A_S = c_1^{SM} + c_2^{SM} + c_3^{SM} + c_4^{SM}$ ,  $\varepsilon = 0.03$ ,  $T_S$  is the slave segment thickness,  $K_S$  is the slave segment bulk modulus and  $\varepsilon_p$  is the penalty scale factor. The function  $g_d$  is given by

$$g_d(x) = x^2 / 4.$$

The frictional stress is given by

$$p_{(S,M)}^j = \mu p_{(S,M)} \frac{s_S^j}{s_S} g_s \left( \frac{s_S}{\mu d_S} \right)$$

where  $\mu$  is the frictional coefficient,  $s_s = \sqrt{s_s^j s_s^j}$  and

$$g_s(x) = \begin{cases} x & x \leq 1 - \varepsilon \\ 1 - \varepsilon g_d \left( \frac{1 + \varepsilon - x}{\varepsilon} \right) & 1 - \varepsilon < x \leq 1 + \varepsilon \\ 1 & 1 + \varepsilon < x \end{cases}$$

The penalty functions are chosen to obtain continuity in the stiffness with respect to deformation, in hope that this will result in a more robust implicit solution procedure. The parameter  $\varepsilon$  is introduced to represent the degree of smoothing of the involved functions. For large values there is lots of smoothing but more penetration, and the other way around for smaller values.

The force on slave segment  $S$  due to interactions with master segment  $M$  is given by

$$f_{il}^S = -(n_i^S p_{(S,M)} + s t_i^j p_{(S,M)}^j) c_I^{SM}$$

and the corresponding reaction force on the master segment is given by

$$f_{il}^M = (n_i^S p_{(S,M)} + s t_i^j p_{(S,M)}^j) c_I^{MS}.$$

The derivation of the true stiffness matrix is a delicate matter, and the result would be non-symmetric. Although LS-DYNA recently added support for non-symmetric matrix contributions we have implemented a symmetric approximation that we feel is sufficient for the time being. The result is

$$\begin{aligned} K_{ijj}^{SS} &= \frac{\partial f_{il}^S}{\partial x_{jj}^S} \approx \frac{K_S \varepsilon_P}{T_S A_S} \left( g'_d \left( \frac{d_{(S,M)}}{\varepsilon T_S A_S} \right) n_j^S c_J^{SM} n_i^S c_I^{SM} + g'_s \left( \frac{s_S}{\mu d_S} \right) \hat{s}_S^j c_J^{SM} \hat{s}_S^i c_I^{SM} \right) \\ K_{ijj}^{SM} &= \frac{\partial f_{il}^S}{\partial x_{jj}^M} \approx -\frac{K_S \varepsilon_P}{T_S A_S} \left( g'_d \left( \frac{d_{(S,M)}}{\varepsilon T_S A_S} \right) n_j^S c_J^{MS} n_i^S c_I^{SM} + g'_s \left( \frac{s_S}{\mu d_S} \right) \hat{s}_S^j c_J^{MS} \hat{s}_S^i c_I^{SM} \right) \\ K_{ijj}^{MS} &= \frac{\partial f_{il}^M}{\partial x_{jj}^S} \approx -\frac{K_S \varepsilon_P}{T_S A_S} \left( g'_d \left( \frac{d_{(S,M)}}{\varepsilon T_S A_S} \right) n_j^S c_J^{SM} n_i^S c_I^{MS} + g'_s \left( \frac{s_S}{\mu d_S} \right) \hat{s}_S^j c_J^{SM} \hat{s}_S^i c_I^{MS} \right) \\ K_{ijj}^{MM} &= \frac{\partial f_{il}^M}{\partial x_{jj}^M} \approx \frac{K_S \varepsilon_P}{T_S A_S} \left( g'_d \left( \frac{d_{(S,M)}}{\varepsilon T_S A_S} \right) n_j^S c_J^{MS} n_i^S c_I^{MS} + g'_s \left( \frac{s_S}{\mu d_S} \right) \hat{s}_S^j c_J^{MS} \hat{s}_S^i c_I^{MS} \right) \end{aligned}$$

where

$$\hat{s}_S^i = \frac{s_S^j s t_i^j}{s_S}$$

represents the direction of sliding in the global coordinate system.

## Examples

All examples in this section are simulated with three contact options, the traditional forming surface to surface contact with (IGAP=1) and without (IGAP=2) stiffness smoothing and the mortar contact. All use implicit integration with BFGS updates, with stiffness reformation every 7<sup>th</sup> iteration and a maximum of 5 stiffness reformations during each step. The automatic time stepping option is turned on with appropriate input parameters.

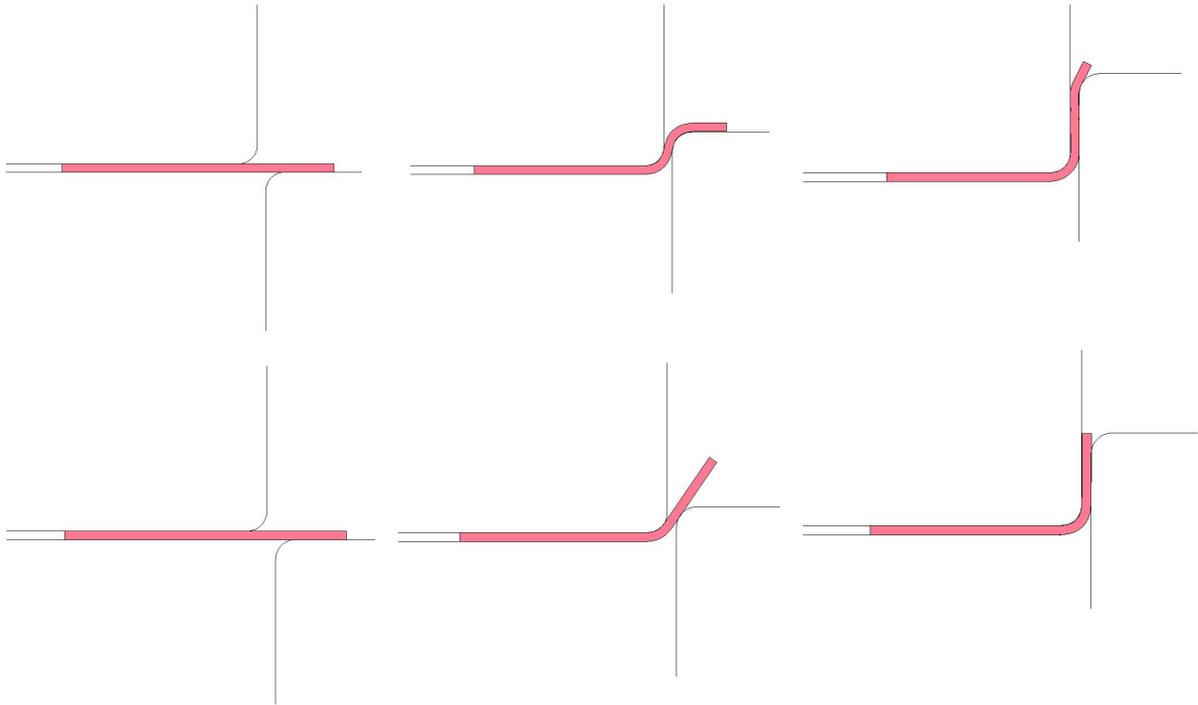


Figure 3 Flanging problem solved with forming contact with stiffness smoothing (above) and with mortar contact (below). The result from forming contact without stiffness smoothing is very similar to the one for mortar contact and is not shown.

The first example is the simple flanging problem illustrated in **Figure 3**. Two rigid tools clamp an initially flat plate while a third rigid tool in motion bends the plate to form a 90 degree flange. By looking at **Table 1** it appears that the traditional forming contact with stiffness smoothing is to prefer as it requires the least number of iterations and smallest CPU time to complete. By looking at the results however in **Figure 3**, the impression is quite different. The tendency of contact sticking with the smoothing option is obvious and the results are completely wrong. For the remaining two contact options the results are acceptable, and considering the implicit performance the mortar contact is the preferred one. The same problem was also run using full Newton, and with this option the mortar contact completes in less than half the CPU time compared to both forming contacts considered, see **Table 2**.

Table 1 Performance indices for the flanging problem, BFGS updates

Contact type	Stiffness reformations	RHS evaluations	CPU time normalized
Forming IGAP=1	11	155	1
Forming IGAP=2	38	993	5.7
Mortar	14	278	1.63

Table 2 Performance indices for the flanging problem, full Newton

Contact type	Stiffness reformations	RHS evaluations	CPU time normalized
Forming IGAP=1	156	510	2.42
Forming IGAP=2	168	376	2.94
Mortar	104	246	1

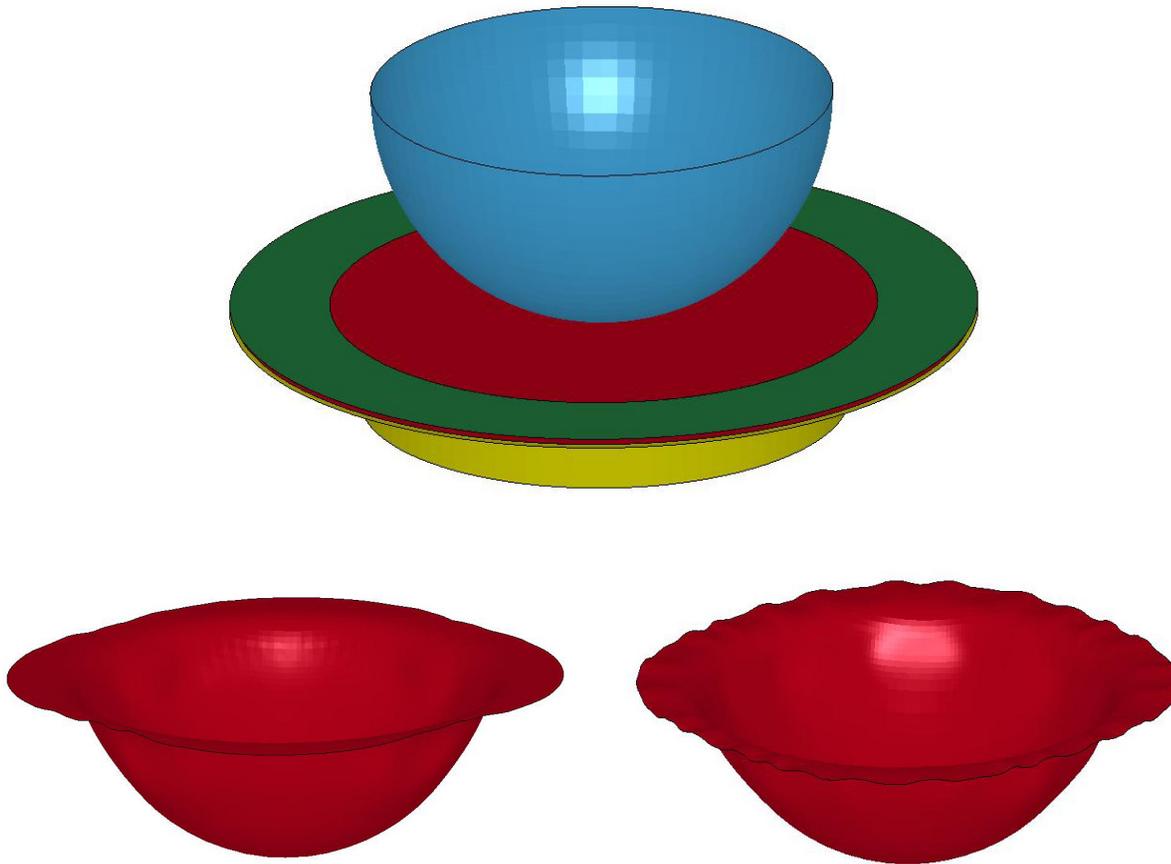


Figure 4 Initial configuration for the forming problem (above) and the resulting cups for forming contact with stiffness smoothing (below left) and mortar contact (below right). The result from forming contact without stiffness smoothing is very similar to the one for mortar contact and is not shown.

The next example is a simple forming problem with look-ahead adaptivity, a circular blank is held onto a die by a holder and a spherical punch is used to deform the blank, see **Figure 4** for the initial configuration. Again the forming contact with stiffness smoothing shows the best performance in terms of speed, and visually the negative effect is not as obvious as in the previous example. Looking closely however, the reader may notice that the wrinkling due to the draw-in is not appearing for this contact, this also being a consequence of the tendency of the contact surfaces to stick. Moreover, by looking at the punch force exerted on the blank in **Figure 5**, we see that it is zero for almost the complete process. The punch does not even get into contact with the blank because of the smoothing, thus when combining speed and accuracy the mortar contact is the best choice even in this case.

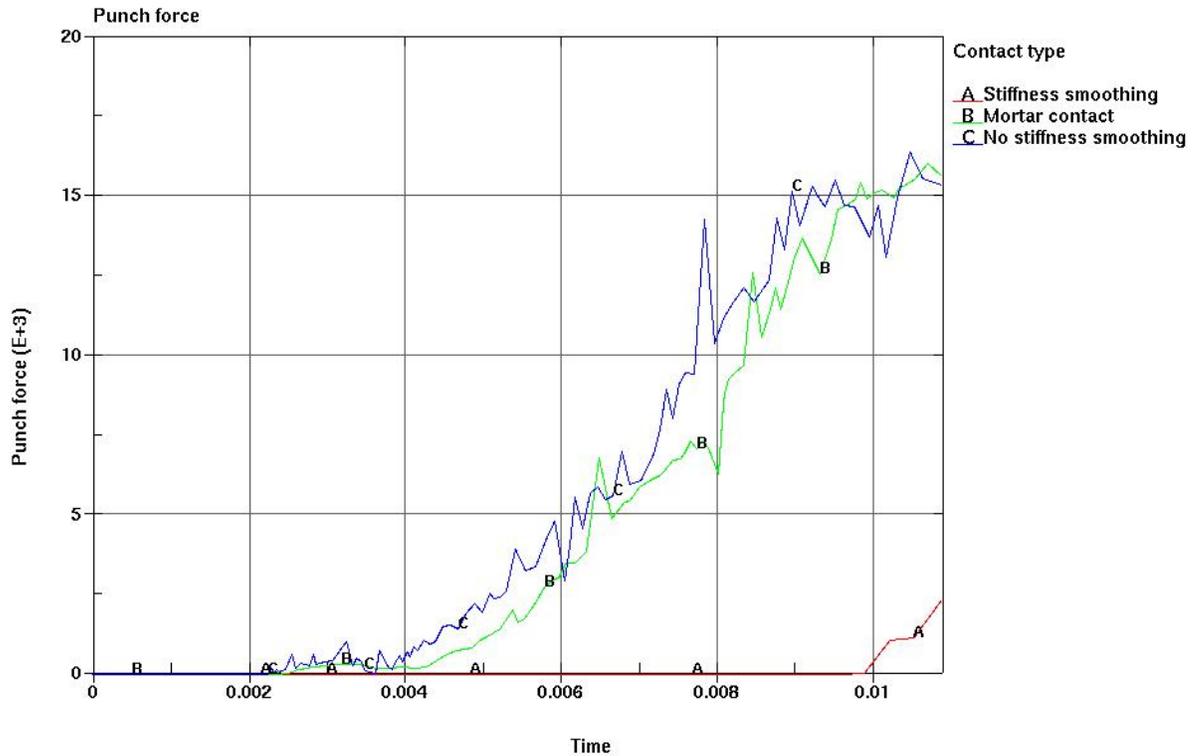


Figure 5 Punch force for the different contact types

Table 3 Performance indices for the forming problem

Contact type	Stiffness reformations	RHS evaluations	CPU time normalized
Forming IGAP=1	63	799	1
Forming IGAP=2	129	6356	3.09
Mortar	93	2579	2.22

The final example is the forming problem illustrated in **Figure 6**. For this problem the forming contact without stiffness smoothing does not even converge and the one with stiffness smoothing actually requires more stiffness reformations and right hand side evaluations than the mortar contact. Note however that the CPU time for the mortar contact is still higher than for the forming contact, indicating that the actual execution of the mortar contact routine is more expensive than for the original forming contact. Although this example is set up to illustrate this in an obvious way and is in a sense to be considered the extreme case, it is concluded that the mortar contact is in general more expensive. The amount by which they differ depends on the type of problem considered, and work is undertaken to identify critical contact situations and to take appropriate measures to improve on this. For large implicit problems however, the effect of this should diminish as the implicit nonlinear and linear part will become more dominant.

Table 4 Performance indices for the ribbed plate problem

Contact type	Stiffness reformations	RHS evaluations	CPU time normalized
Forming IGAP=1	18	374	1
Forming IGAP=2	-	-	-
Mortar	10	260	1.40

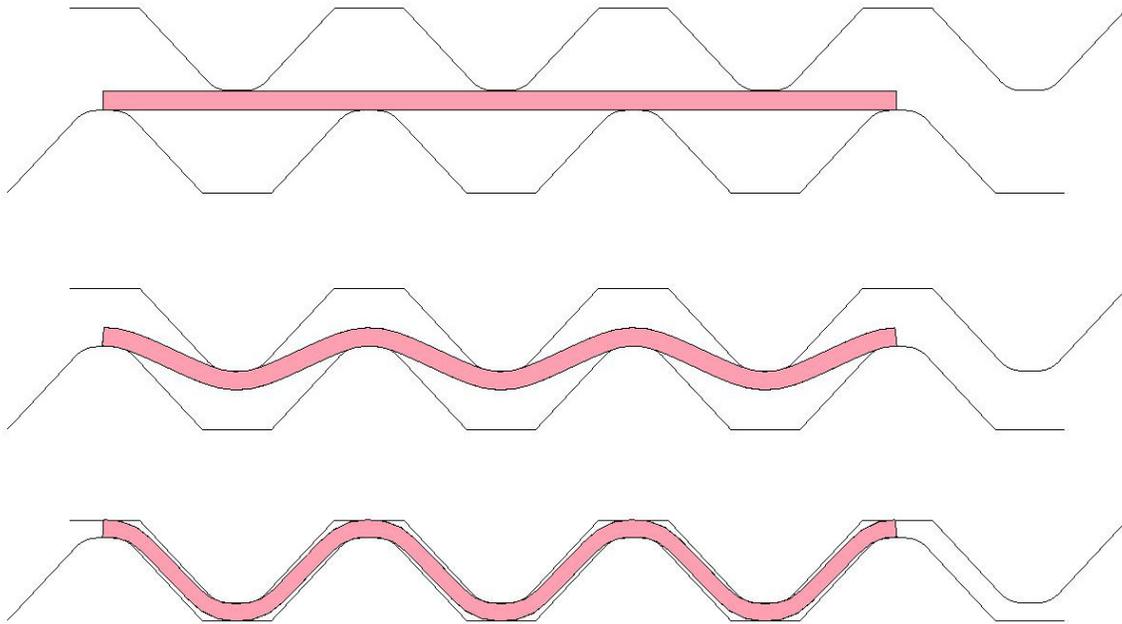


Figure 6 Forming of a ribbed plate

## Conclusions

A segment to segment mortar contact method for stamping analyses is implemented in LS-DYNA and presented in this paper. It is primarily intended for implicit analyses, and possesses nice smooth properties that are intuitively attractive for these types of simulations. Implementation and theoretical details are treated, as well as directions of usage. Some examples are presented that illustrate the advantages of the new method compared to already existing ones, and it is generally concluded that it is more robust and accurate. It is generally not as fast as using the forming contact with stiffness smoothing, but stiffness smoothing tends to stick contact surfaces together and should be used only for problems where no contact release is expected at any point in time and even then the results may be questionable. The mortar contact requires the calculation of the overlapped area between contact segments which makes it more expensive than the node to surface contacts, and there is probably room for efficiency improvements in this direction. Also the MPP version is yet to be implemented but this is underway. In general, the new method seems to have an overall impact on the implicit performance for stamping problems.

## References

1. M.A. Puso and T.A. Laursen, A mortar segment-to-segment contact method for large deformation solid mechanics, *Comput. Methods Appl. Mech. Engrg.* 193 (2004) 601-629.
2. M.A. Puso and T.A. Laursen, A mortar segment-to-segment frictional contact method for large deformations, *Comput. Methods Appl. Mech. Engrg.* 193 (2004) 4891-4913.