

Conjugate Heat Transfer in LS-DYNA[®]: An Update of the ICFD-Structure Coupling Capabilities for Hot Stamping

Iñaki Çaldichoury¹ Rodrigo Paz¹, Facundo Del Pin¹, Chienjung Huang¹
¹Livermore Software Technology, an Ansys company, Livermore, CA, USA

Abstract

Hot stamping is an essential process in the long and complex assembly chain that will lead to the manufacturing of a complete vehicle. The three main steps consist in a metal sheet being heated, formed and rapidly cooled, yielding parts with high strength and light weight. The main focus of research in that area resides in reducing cycle times, i.e. the time it takes for the tools to cool down, while simultaneously maintaining a high integrity of the formed workpieces. To that effect, the design of cooling systems has drastically increased in complexity and new guiding models are needed for the engineer to be able to identify and then correct potential 'dead flow zones', 'hot spots' and other problematic areas.

Simulation is increasingly viewed as the most general and versatile tool to tackle those challenges. The physics require a coupled thermal, fluid and often mechanical simulation for which a Multiphysics code is needed. Within LS-DYNA, the ICFD solver offers such capabilities and efforts have been continuous over the years to improve existing capabilities and add new ones in conjunction with user feedback [1] [2]. This paper will offer an overview of the existing capabilities, reveal some best practice approaches as well as introduce the latest developments, with a special focus on the fluid-structure interface and how turbulent effects may affect the heat transfer.

The ICFD Solver for Conjugate heat transfer: Principles

LS-DYNA is increasingly recognized for its multiphysics capabilities. For the current application, both the incompressible CFD and the solid mechanics thermal solvers will be coupled together. The ICFD solver starts by solving the Navier Stokes equations using a fractional step approach (Ref [3]). Thanks to the incompressibility hypothesis, the heat equation is solved separately. Its expression in the fluid is similar to the solid with the addition of a convection term representing the fluid's velocity effects.

$$\frac{\partial T}{\partial t} + \vec{V} \vec{\nabla} T = q_v + \alpha \Delta^2 T \quad (1)$$

When conjugate heat transfer is involved, the two domains are combined into one single system resulting in a monolithic solve. The interface between the fluid and the solid is typically handled by a constraint method with the fluid nodes 'seeking' the closest solid face before being projected on it and creating a new constraint condition as described on Figure 1. This method is considered robust and allows for an accurate representation of an ideal thermal contact between two interfaces. It should therefore be understood as being the preferred modelling choice. It is however also important to note a few repercussions which follow this approach. The first comes from its numerical implementation. Since it is the fluid nodes that are projected on the solid faces, it is recommended for the fluid mesh to be at least as fine as the solid mesh in order to ensure that no free node is left on the interface that could potentially lead to a loss of accuracy. This criteria is often met in practice since an accurate representation of the fluid commonly requires a finer mesh than for solid mechanics. Secondly, the fact that a projection point is used to determine the constraint system may lead to some inaccuracies in the temperature flow across the interface in highly non-conforming mesh cases and when the material's diffusion coefficients are low.

Lastly, this method only allows for an ideal contact to be modelled across the interfaces without any temperature losses. In order to tackle those two last points, an alternative contact algorithm based on the mortar penalty method is currently being investigated and will be discussed in a following section.

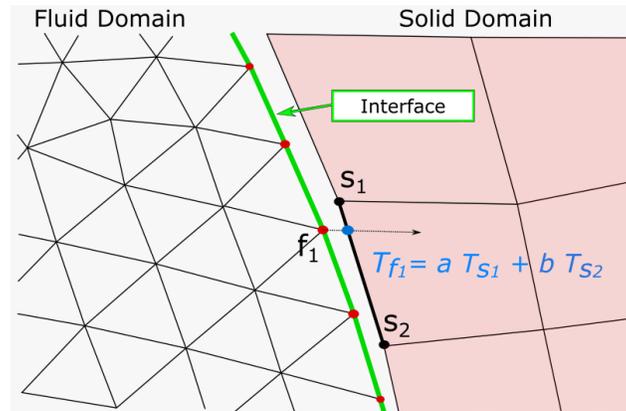


Figure 1 Establishment of a new constraint at the fluid structure interface

Another item to consider is which solver and pre-conditioner to use. Currently two approaches are available for the user. A direct method (solver type 20 in *CONTROL_THERMAL_SOLVER) and a GMRES iterative solver with Diagonal preconditioner (Solver type 17). Given the typical sizes of the problems, the iterative solver will mostly preferred in Conjugate Heat Applications (See Next Section).

Best practice approach for tool cooling simulations

One of the challenges of hot stamping simulations is dealing with the numerical cost associated with solving the complete coupled CFD thermal analysis. In a fully transient simulation, the required timestep for the Navier Stokes equations solve is often in the vicinity of 1.e-4 seconds while the total process can take several dozens of seconds or minutes. For this reason a decoupled approach is preferred. Since for this application the influence of temperature change on the fluid quantities can be considered negligible, the fluid velocity and pressure can be solved separately at the beginning of the run. Then, once a steady state has been achieved, the conjugate heat transfer problem can be solved in a transient manner using the results from the steady state analysis to assemble the forced convection term for the heat equation. By proceeding in such a way, calculation times can be greatly improved since after steady state has been reached, only the heat equation remains to be solved. But also, greater timesteps can be used, often in the order of 1.e-2 or higher which further improves solve times.

Finally, in many cooling scenarios, the mechanical deformations or displacements are few. For such configurations it may not be needed to reassemble the thermal matrix from scratch at every timestep. This matrix reassembly frequency can be controlled in the fifth flag of the keyword *CONTROL_THERMAL_SOLVER is Post-R12 versions.

A mock example showing an industrial tool cooling process simulation will be introduced for illustration purposes (See Figure 2) consisting of 1.3M elements for the fluid and roughly 350 000 for the solid part. In the first part of the calculation, the first order steady state solver is used in order to solve for the velocity and pressure fields in the pipe. The standard k-epsilon model is being used to model the turbulent flow and the boundary layer mesh parameters have been chosen accordingly to ensure a Y^+ between 20 and 100 at the wall.

After steady state has been reached (5000 iterations, 2.5 seconds per iteration), the coupled thermal problem is allowed to proceed until time reaches 30 seconds. The runs were conducted using 24 CPUs (2x12 cores per node) on an early 2010s cluster Intel Xeon E5645 @2.4GHz. As such, it is important to keep in mind that in all likelihood, newer generations of CPUs allow for faster solve times. The interconnection is Mellanox InfiniBand.

The results for the various options are being shown in Table 1 while Figure 3 shows some temperature fringes results. Unsurprisingly, the iterative solver offers the best run times. However, it is also interesting to note that the time per iteration drops by a couple of orders of magnitude with the direct solver when skipping the matrix assembly. For those iterations, the direct solver performs even better than the iterative solver. This is a logical result since, for the direct solver, the matrix assembly steps performs a factorization which is then saved for the subsequent steps which only need to perform a triangular solve. This results in very cheap calculation costs. Conversely, the gains per step between reassembling the matrix and not reassembling are not so extreme with the iterative solver. This is an interesting result since it implies that while the iterative solver is the obvious choice in cases where frequent matrix reassemblies are needed (cases with moving pieces for instance), the direct solver becomes relevant again to use in configurations where this is not the case. If the thermal stiffness matrix remains unchanged during the entire run, the assembly would need to be performed only once at the beginning of the run.

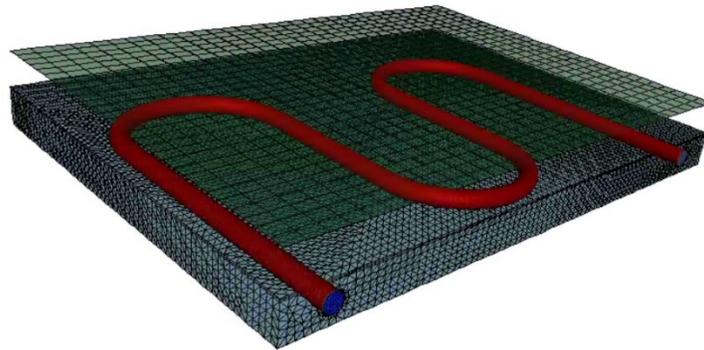


Figure 2 Tool cooling simulation: Example can be found on dynaexample.com

Table 1 Steady solve of N.S equations followed by CHT analysis: the time per iteration only refers to the heat equation solve during the transient phase while the total time includes all contributing factors (mesh assembly, initial fluid solve, etc.).

<i>Tool Cooling</i> <i>(constraints contact)</i>	MF2 direct solver	MF2+matrix update every ten timesteps	GMRES iterative solver	GMRES +matrix update every ten timesteps
Time per CHT iteration	22. seconds	0.25 seconds	1.7 seconds	0.4 seconds
Total run time	13 hours	4 hours 30 minutes	4 hours 20 minutes	3 hours 40 minutes

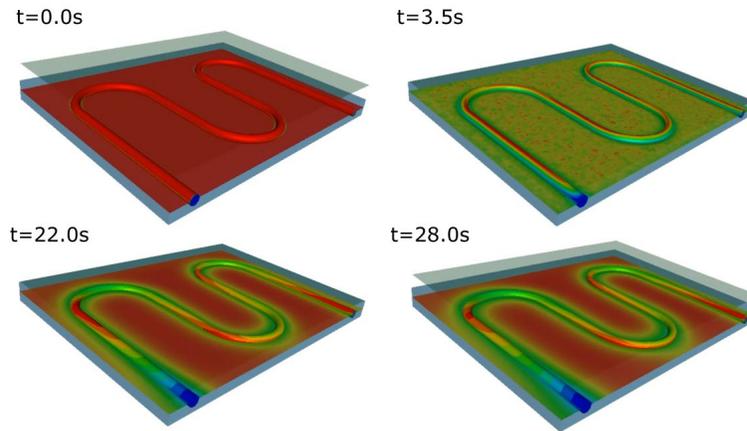


Figure 3 Temperature fringes (using relative scales) showing cooling of the die by the fluid

New coupling approach: mortar contact

A new thermal contact is investigated based on the mortar contact approach. It is a special type of surface to surface contact where the intersection area between the interface solid and fluid faces allows to construct a local stiffness matrix applied to the degrees of freedom (the nodes in this case) belonging to said faces found in contact (See Figure 4). This local stiffness matrix is then added to the global stiffness matrix of the thermal system thus adding a penalty for those specific dofs. This penalty factor can be multiplied by the heat transfer coefficient value between the two domains to symbolize the heat exchange. If that value is high enough, a perfect thermal contact will be established.

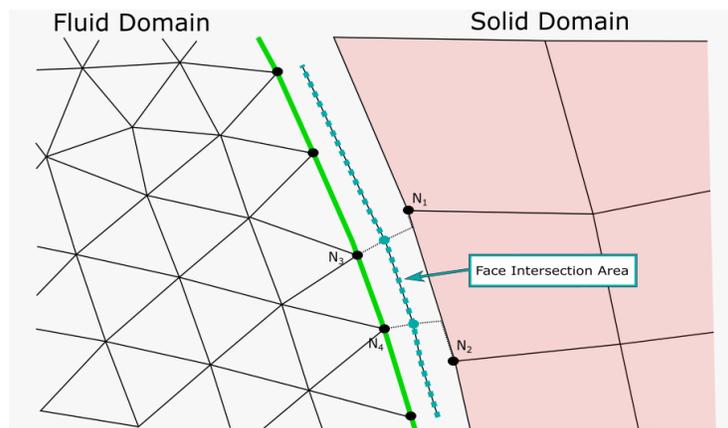


Figure 4 Creation of a new local stiffness matrix between the Nodes N_1, N_2, N_3 and N_4

The advantages of the mortar approach is that, if the contact parameters are well defined, a continuous contact intersection area is created and, based on it, 'realistic' penalty coefficients or weights are assigned to each degree of freedom. Therefore, this removes the uncertainties caused by the constraint matrix set up in certain configurations and guarantees a smooth transition of the temperature front between the two domains even in highly non-conforming meshes. In summary, the Mortar method offers a unique and well defined way to build the added stiffness matrix, compared to the multiple ways to build constraints (choice of master/slave, ambiguity in the factors when building the constraint matrix, etc ...). Finally, by defining and adjusting the heat transfer coefficient between the two domains, users are allowed to take into account heat losses which can be induced by radiation, external convection or turbulent effects in the fluid (see next section).

Unfortunately, the drawbacks of this method also quickly become apparent. In order to insure a perfect thermal contact, a ‘high’ value of heat transfer coefficient has to be defined. However, those high value terms will often cause the stiffness matrix conditioning to deteriorate. This will lead to higher calculation times, and even prevent convergence of the solution especially for iterative methods. Table 2 shows some calculation time results based on the same problem as in the previous section and using the maximum heat transfer coefficient value allowed by the solver (calculated internally based on model parameters and mesh size) to achieve perfect thermal contact.

Table 2 Steady solve of N.S equations followed by CHT analysis: the time per iteration only refers to the heat equation solve during the transient phase while the total time includes all contributing factors (mesh assembly, initial fluid solve, etc.).

<i>Tool Cooling (mortar contact)</i>	MF2 direct solver	MF2+matrix update every ten timesteps	GMRES iterative solver	GMRES +matrix update every ten timesteps
Time per CHT iteration	35. seconds	0.25 seconds	15. seconds	3.2 seconds
Total run time	18 hours	5 hours 10 minutes	10 hours	5 hours 20 minutes

Investigations are ongoing to improve the calculation speeds using the approach. Post R12, this method is available to users in the *ICFD_BOUNDARY_CONJ_HEAT keyword where the fluid surface mesh part can be defined as using the constraint method or the mortar approach for handling the heat exchange with the structure.

*ICFD_BOUNDARY_CONJ_HEAT

PID	CTYPE	VAL	SFLCID

Where CTYPE is the coupling type, the default value 0 is for the constraint contact whereas CTYPE=1 turns on the mortar approach. VAL allows the user to specify a temperature drop if CTYPE=0 or else gives the Heat transfer coefficient value at the interface between the two domains if CTYPE=1.

Turbulent thermal law of the wall

In CFD law of the walls are frequently employed to estimate the velocity and shear happening at the wall for turbulent flow. They often offer the advantage of being computationally cheap, reasonably accurate and reduce the dependency of the results on the boundary layer mesh. Similar laws of the wall exist for the heat equation and work is in progress on their implementation within the ICFD solver. The final objective is to offer a more accurate heat transfer at the wall which could then be applied to conjugate heat transfer applications using the penalty approach (See Previous section). The current implementation is available for beta testing in Post R12 versions.

In the keyword ICFD_CONTROL_TURBULENCE, flag number 7 and flag number 8 will refer to the choice of thermal law of the wall and flag number 8 will give the critical thermal Y_{+t} . If Y_{+t} is not defined, the solver will automatically estimate its value using:

$$Y_{+tc} = \frac{Y_{+c}}{Pr^{1./3.}} \quad (2)$$

With $Y_{+c} = 11.225$ the critical Y_{+} value and Pr the Prandtl number.

If flag7=0, no turbulent heat flux is calculated, and the heat flux calculated at the wall is equal to:

$$Q = -k \frac{(T_s - T_c)}{y} \quad (3)$$

with T_s the temperature at the wall node and T_c the temperature at the first internal node in the fluid volume.

But if flag7 is greater or equal to 1, then the turbulent heat flux will be calculated as an additional output variable:

$$Q_t = \rho C_p \frac{U_\tau}{T_+} (T_s - T_c) \quad (4)$$

with ρ the density, C_p the heat capacity, U_τ the shear velocity and T_+ a dimensionless temperature calculated with the thermal law of the wall :

if Flag 7 greater or equal to 1 :

if (Y_{+t} .le. Y_{+tc}) then :

$$T_+ = Pr_t Y_+ \quad (5)$$

else :

If flag 7 =1 , then :

$$T_+ = \frac{Pr_t}{\vartheta} \log(Y_+) + (3.85 Pr_t^{1./3.} - 1.3)^2 + 2.12 \log(Pr_t) \quad (6)$$

If flag7=2, then :

$$T_+ = \frac{Pr_t}{\vartheta} \log\left(\frac{Y_+}{Y_{+t}}\right) + Pr_t Y_{+t} \quad (7)$$

with Pr_t the turbulent Prandlt number taken as a constant here as 0.85 and ϑ the von Karman constant.

The T_+ and the turbulent Heat flux are output as surface variables which LS-PrePost[®] can read.

If the turbulent law of the wall is activated, then the turbulent heat flux will be used when outputting the heat convection coefficient:

$$HTC = \frac{Q_t}{(T_s - T_b)} \quad (8)$$

with T_s the temperature at the wall node, and T_b the bulk temperature, also known as the reference temperature.

To set up this bulk temperature, several options exist in the keyword:

*ICFD DATABASE HTC

OUT	HTC	TB

if HTC = 0, then T_b is calculated the following way, using all the nodes on the pipe cross-section :

$$T_b = \frac{\sum_i T_i V_i}{\sum_i V_i} \quad (9)$$

With T_i the temperature at a given node in the cross section. This method is considered as the costliest.

if HTC = 1, then T_b is a constant defined by the flag TB.

If HTC=2, then T_b is the closest internal node in the volume mesh $T_b = T_c$

If HTC=3, then T_b is taken as the temperature at a certain distance Y_+ from the wall. This Y_+ is specified by the user in the keyword's third flag TB.

If HTC=4, same as above, but now, TB directly gives the distance Y instead of Y_+ .

If HTC=5, then a DEFINE_FUNCTION given by TB is used. Allowed parameters are:

'x', 'y', 'z', 'vx', 'vy', 'vz', 'temp', 'pres', 'shear',
 & 'time', 'dt', 'yplus', 'tplus', 'HTC', 'rhocp', 'alpha', 'hflux',
 & 'hfluxt', 'tclose'

This thermal law of the wall is currently under review and its implementation is yet subject to change. Feedback is welcomed.

Conclusion

Conjugate heat transfer applications are a prime example of pulling various LS-DYNA capabilities together for solving problems involving multiple areas of physics. This paper has shown some of the latest developments in this area. Three axes of research are being continuously investigated. The first one is the linear algebra aspect where different solving methods (direct, iterative solvers) and preconditionners are periodically tested and their performance studied. The second one, closely linked to the first, is implementing different contact methods or other features in order to increase the accuracy or/and calculation speeds. Finally the third aspect is related to physics and how to improve the numerical modelling of the real life phenomenon as shown by the turbulent law of the wall.

References

- [1] T. Kuroiwa, "Tool cooling simulation for hot forming", Proceeding of the 11 th European LS-DYNA User Conference, 2017.
- [2] U. Gohner, B. Boll, I. Caldichoury, T Wicke "Coupled Simulation of the fluid flow and Conjugate heat transfer in Press hardening Processes", Proceeding of the 13 th International LS-DYNA User Conference, 2016.
- [3] Del Pin, F., Huang, C., Çaldichoury, I. *et al.* On the performance and accuracy of PFEM-2 in the solution of biomedical benchmarks. *Comp. Part. Mech.* 7, 121–138 (2020).