New Developments in *CONSTRAINED_BEAM/SHELL/SOLID_IN_SOLID

Hao Chen, Ian Do

Livermore Software Technology Corporation

Abstract

**CONSTRAINED_BEAM_IN_SOLID card was developed originally only to solve rebar enforced concrete. It is to replace the problematic *CONSTRAINED_LAGRANGE_IN_SOLID ctype 2 implementation.*

During the past two years, new enhancements and capabilities have been added accompanied by the expanding usage of this card. Initially rebars, then composite materials. Later it even finds its usage in medical devices.

*In this paper new developments in *CONSTRAINED_BEAM_IN_SOLID card, in additional to two new keywords in the same family: *CONSTRAINED_SHELL_IN_SOLID and *CONSTRAINED_SOLID_IN_SOLID are introduced.*

Two New Keywords

There were two new keywords added recently: *CONSTRAINED_SHELL_IN_SOLID and *CONSTRAINED_SOLID_IN_SOLID. They deal with embedded shell and solid structures inside a solid part. They share the same algorithm and most of the code with *CONSTRAINED_BEAM_IN_SOLID and could be seen as two variants of the *CBIS card. Same as *CONSTRAINED_BEAM_IN_SOLID, the default coupling scheme is constraint based. Penalty formulation is invoked when the optional "_PENALTY" is added after "*CONSTRAINED_SHELL/SOLID_IN_SOLID" card. Their keyword formats are as follows.

*CONSTRAINED_SHELL/SOLID_IN_SOLID(_PENALTY)							
SLAVE	MASTER	SSTYP	MSTYP				
START	END				PSSF		

"SLAVE" and "SSTYP" give us the embedded structure Part/PartSet ID; "MASTER" and "MSTYP" the solid Part/PartSet ID. "START" and "END" mark the starting and end time.

"PSFF" is used to input the scale factor when calculating penalty stiffness. It is only valid for penalty formulation. The penalty stiffness calculation is based on bulk modulus of SLAVE/MASTER parts and similar to that of "soft=0" in CONTACT.

Implicit Enhancements

A big improvement in *CBIS is its capability to relax the constraint along axial direction. This, in addition to the AXFOR feature, which is to apply slip force based on slip distance between beam and solid, provided a powerful tool in simulating problems like pretensioners in concrete, fiber enforced composites. But in early versions, implicit implementation failed to consider the relaxed constraint along axial direction. Instead, the linear constraint matrix was generated at all three global axes. This works if the relative motion between the beam and solid nodes are not big. But for general problems, even if the run could continue the convergency would be too good.

Also, the initial implicit implementation did not take account into the slip force. We know that in nonlinear implicit analysis, forces needs to be linearized as a product of stiffness matrix and displacements. However, this was not done for the slip force option and this also causes bad convergency.

Efforts were spent to fix the above two problems in order to achieve smooth implicit analysis. First, the linear constraint is applied in a local coordinate system at two local directions perpendicular to the beam axial direction. Now the constraint along beam axis is released. Beam could move freely along axial direction, of course, if no slip force applied. Secondly, in this local coordinate system, a stiffness matrix is generated between the beam node and up to 8 solid nodes. The local stiffness matrix is constructed with nonzero components only along beam axial direction; the other two directions all components should be zero. Then the local stiffness is transformed into global stiffness via a rotation. The final stiffness matrix is of 27 by 27.

These two enhancements brought a much better convergency in users' problems.

Thermal Support

Temperature is defined at node, same as velocity. To enable the coupling between beam and solid nodes, we need to add an extra set of equations for temperature, in addition to the 3 velocities. The problem is that the thermal solver is an implicit solver independent from the general mechanical solver. So we have to build a separate linear constraint matrix for temperature and load it into the global thermal solver matrix. The early version of thermal support was incomplete and only worked on limited cases. This problem has been fixed and we now have full thermal support for *CBIS.

Conclusions

New capabilities in *CONSTRAINED_BEAM_IN_SOLID family are discussed in this paper. The developers at LSTC are committed to continually work with our users to improve.