A User-Defined Element Interface in LS-DYNA v971

Thomas Borrvall
Engineering Research Nordic AB
Linköping, Sweden

Abstract

The user-defined features in LS-DYNA are powerful tools that allow users in academia or industry to verify research results in the context of general and complicated finite element applications. Implementation work concerns only the special field of interest, and there is no need for the comprehensive task of developing and maintaining the complete finite element software. One of the new user-defined features in LS-DYNA v971 is the possibility to define structural solid and shell elements. Up to a total of ten element formulations can be implemented in a single LS-DYNA executable both for explicit and implicit analyses. A high abstraction-level interface is in particular provided for numerically integrated elements, and stabilization schemes can easily be incorporated. There is also the option to implement discrete elements, and property parameters and history variables can be associated with the element. The interface is equipped with additional features that facilitates research on element technology, but also makes it perfectly suited for educational purposes. An overview of the procedure of implementing an element in the new interface as well as invoking it from the keyword input file will be presented.

Introduction

Finite element technology is an active area of research and the topic of a considerable amount of produced papers every year. The driving force is to develop elements with better performance, in particular for large-scale calculations, and apparently there will never be a perfect element. Typically the nature of a particular application determines what type of element is called for, and the difficulties facing the finite element developer include the alleviation of volumetric locking for (nearly) incompressible materials, the alleviation of shear locking for nonlinear shell elements and the delicate trade-off between computational speed and accuracy. Mixed or hybrid elements, assumed strain elements and reduced integration elements with stabilization are examples of element classes that have shown to overcome many of the mentioned difficulties and sustained the test of time.

Key properties of a finite element such as stability and completeness can on one hand be derived from the mathematical formulation, but this is often intricate and patch tests are preferred in practice. Patch tests are simple models that are set up to test the ability of a finite element to reproduce specified strain fields and in a sense reveal the soundness of the formulation. Since only a few finite elements are involved, computational efficiency is not an issue and they are easily implemented in a high-level programming language. Often however, a newly developed element is intended for a particular application, e.g., sheet metal forming, and it is of interest to test it out also in this context. This requires that the element is implemented together with other important features such as material models, contact algorithms, boundary conditions and a finite element solver, and computational efficiency and robustness becomes a major concern. Needless to say, this is an untenable situation and a researcher should only be concerned with the development in his/her own field and should not have to deal with complete development and maintenance of the finite element software.

User-defined features in LS-DYNA [1] allow users to create their own FORTRAN or C subroutines and link them with the LS-DYNA objects to form a tailor-made LS-DYNA executable. In this way LS-DYNA can be customized for particular applications, which makes it a powerful tool in research and development. Throughout the years, the list of available user-defined features has grown and currently includes the following items

- Structural materials
- Structural elements
- Thermal materials
- Equation of state
- Airbag sensors
- Solution control, including far-reaching control of output of solution results
- Friction
- Interface control, which enables control of contacts on a low level
- Weld failure
- Loads
- Control over output of shell stresses and history variables to the d3plot database
- Material failure for commonly used material models for metals
- Mesh refinement for 3D-shell adaptivity

Among the recently introduced features, user-defined structural element is the topic that this paper is devoted to. To implement a user-defined element, an object version of LS-DYNA must be retrieved from your local LS-DYNA distributor. Furthermore, a FORTRAN compiler is needed to compile the code and must be bought separately. In the LS-DYNA object distribution there is a FORTRAN text file containing all user-defined element subroutines. You will need to edit subroutines according to the particular task that you want to accomplish. Once this is done, the code is compiled and linked with the rest of the LS-DYNA objects typically by executing a makefile script either from command prompt or from a graphical user interface. The result is an LS-DYNA executable ready to be used.

The paper is organized as follows. First a theoretical background is given on element technology after which an overview of the implementation procedure is given. Information on how to invoke the user-defined element is given and the paper ends with some conclusions.

Formulation

A general formulation of a structural element can be rather involved and is out of the scope of this paper, see for instance [2] for a more comprehensive treatment. We will here present a framework that is representative for the user-defined element interface in LS-DYNA, and that can be used as a reference throughout the rest of the paper.

We assume that the element consists of N nodes and that each node has n degrees of freedom. If not explicitly stated, lower-case letters i, j, k, l represent degrees of freedom and are in the range 1 to n, lower-case letters p, q, r, s, t represent space dimensions and are in the range 1 to 3 and upper-case letters I, J represent node numbers and are in the range 1 to N. Also, the Einstein summation convention is understood so repeated indices are summed over their range.

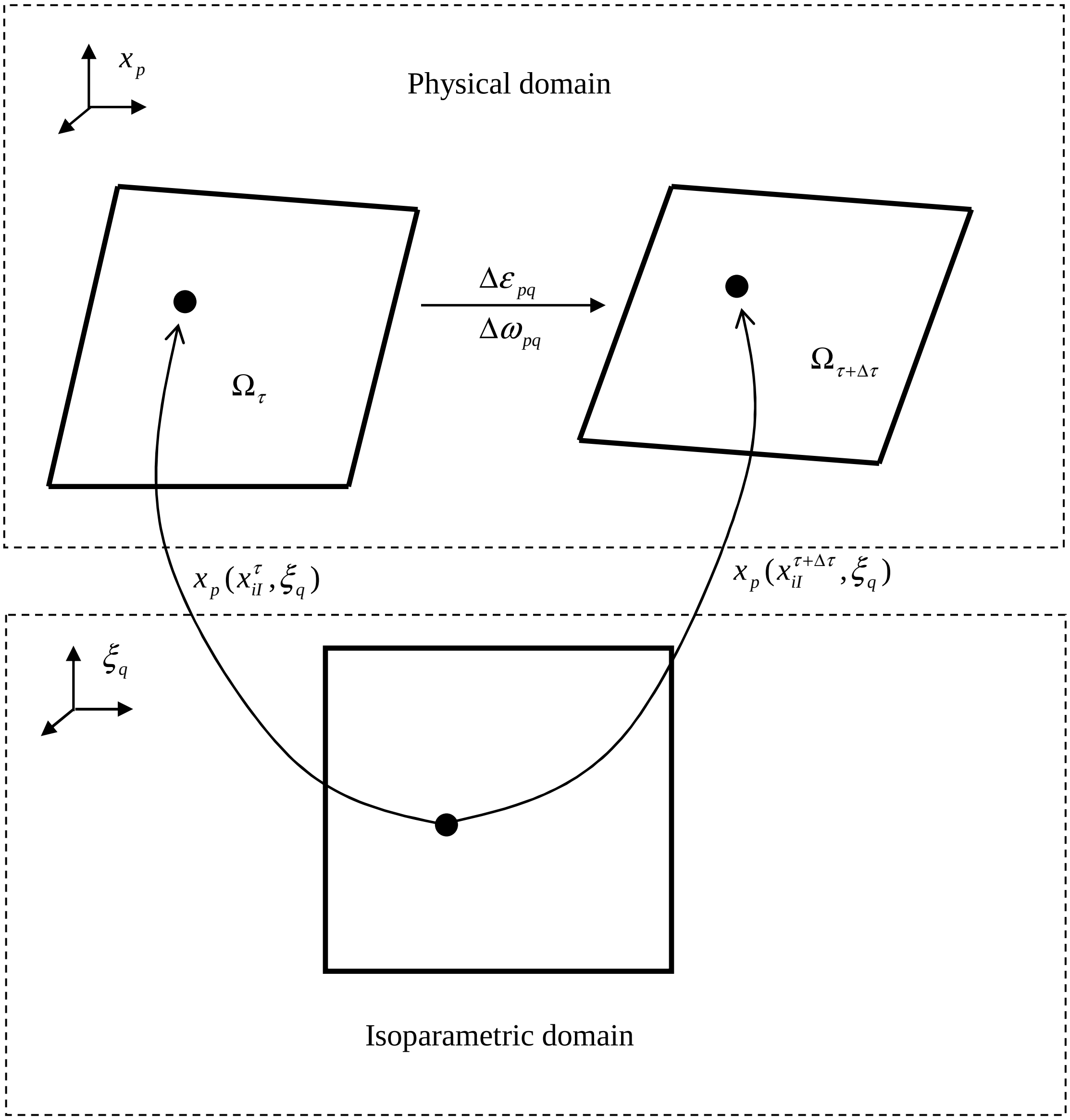

Figure 1: Element configurations

The configuration of the element is at time τ described by the nodal coordinate vector x_{iI}^{τ} and an interpolation scheme that is characteristic for the element formulation. For shells, a co rotational formulation is used where a local coordinate system is attached to and rotates with the element. The coordinate vector may or may not be expressed in this system depending on the user's choice. The degrees of freedom in the nodal coordinate vector may represent translational, rotational and possibly additional degrees of freedom. The aim is to compute the internal force vector $f_{iI}^{\tau+\Delta\tau}$ for the configuration represented by $x_{iI}^{\tau+\Delta\tau}$, where $\Delta\tau$ is the time step size. The internal force vector gets contributions from physical stress in the material and from additional stabilization stress so we can write it as

$$f_{iI}^{\tau+\Delta\tau} = {}^{S}f_{iI}^{\tau+\Delta\tau} + {}^{H}f_{iI}^{\tau+\Delta\tau}.$$

Furthermore, in implicit analysis it is necessary to compute the tangent stiffness matrix denoted $K_{iIjJ}^{\tau+\Delta\tau}$. This matrix is expressed as

$$K_{iIjJ}^{\tau+\Delta\tau} = {}_{\mathrm{mat}}^{S}K_{iIjJ}^{\tau+\Delta\tau} + {}_{\mathrm{geo}}^{S}K_{iIjJ}^{\tau+\Delta\tau} + {}^{H}K_{iIjJ}^{\tau+\Delta\tau},$$

i.e., as the sum of the material and geometric stiffness matrix from the physical stress and the stiffness matrix from the additional stabilization stress. When necessary, the force vector and stiffness matrix are transformed to the global coordinate system before assembly.

In the following the two force contributions are elaborated on in more detail.

Numerical integration

For the force contribution from physical stress, we use numerical integration and an isoparametric formulation. We introduce ξ_q to be the isoparametric coordinate that is independent of time and refer to Figure 1 for the following. At time τ, the physical position of a point x_p^τ in the element is given by

$$x_p^\tau = x_p(x_{iI}^\tau, \xi_q),$$

that defines a mapping from the isoparametric to the physical domain. For shells the physical point is expressed in the local coordinate system. The velocity of the physical point is given by

$$\dot{x}_p^\tau = \frac{\partial x_p^\tau}{\partial x_{iI}^\tau}\dot{x}_{iI}^\tau = \frac{\partial x_p^\tau}{\partial x_{iI}^\tau}R_{ij}^\tau v_{jI}^\tau,$$

where R_{ij}^τ is a transformation matrix between solution velocities v_{iI}^τ and coordinate velocities $\dot{x}_{iI}^\tau$ that is independent of the isoparametric coordinate. As an example, the transformation matrix could represent the relationship between fiber vector velocities and rotational velocities.

The velocity gradient increment matrix ΔL_{pq} is computed according to

$$\Delta L_{pq} = B_{pqiI}^{\tau+\Delta\tau/2} d_{iI},$$

where

$$B_{pqiI}^\tau = \tilde{B}_{priI}^\tau \left(g^\tau\right)_{rq}^{-1}$$

is the physical gradient displacement matrix at time τ and d_{iI} is the displacement vector obtained as the solution from the finite element solver. We have introduced

$$\tilde{B}^{\tau}_{pqiI} = \frac{\partial^2 x^{\tau}_p}{\partial \xi_q \partial x^{\tau}_{jI}} R^{\tau}_{ji}$$

and

$$g^{\tau}_{pq} = \frac{\partial x^{\tau}_p}{\partial \xi_q}$$

as the isoparametric gradient displacement matrix and jacobian matrix from the isoparametric to the physical domain, respectively.

The strain and rotational increment matrices are obtained as the symmetric and skew-symmetric parts of the velocity gradient increment matrix, respectively,

$$\begin{aligned} \Delta\varepsilon_{pq} &= \frac{1}{2}(\Delta L_{pq} + \Delta L_{qp}) \\ \Delta\omega_{pq} &= \frac{1}{2}(\Delta L_{pq} - \Delta L_{qp}) \end{aligned} .$$

The stress matrix at time τ is denoted σ^{τ}_{pq} and we emphasize that both the strain increment matrix and stress matrix is expressed in the local coordinate system for shells. The constitutive update is written in general terms as

$$\sigma^{\tau+\Delta\tau}_{pq} = \sigma_{pq}(\sigma^{\tau}_{rs}, \Delta\varepsilon_{rs}, \Delta\omega_{rs}, ...),$$

meaning that it depends not only on the previous stress state and the strain increment matrix, but also on material properties and history. The rotational increment matrix is used for solids to rotate the stress for objectivity. The update of history variables is implicitly understood and omitted in writing.

The expression of virtual work results in the following expression of the internal force vector

$${}^{S}f^{\tau+\Delta\tau}_{iI} = \int_{\Omega_{\tau+\Delta\tau}} B^{\tau+\Delta\tau}_{pqiI} \sigma^{\tau+\Delta\tau}_{pq} d\Omega ,$$

where Ω_{τ} is the physical element domain at time τ. This integral can be transformed to the isoparametric domain and numerically integrated using e.g. Gauss integration. This results in the following approximate expression for the internal force vector

$${}^{S}f^{\tau+\Delta\tau}_{iI} \approx \sum_{m=1}^{M} B^{\tau+\Delta\tau}_{pqiI}(\xi^m_t)\sigma^{\tau+\Delta\tau}_{pq}(\xi^m_t) J^{\tau+\Delta\tau}(\xi^m_t) w_m / 8 .$$

Here ξ^m_t is the isoparametric coordinate of integration point m, M is the total number of integration points, w_m is the weight at integration point m, and

$$J^{\tau} = \det\left(g^{\tau}_{pq}\right)$$

is the scalar jacobian, or relative volume, between the element and isoparametric domain at time τ.

For implicit analysis, the tangent stiffness matrix is derived by taking the variation of the internal force vector. The result is that

$${}^{S}_{\text{mat}}K^{\tau+\Delta\tau}_{iIjJ} \approx \sum_{m=1}^{M} B^{\tau+\Delta\tau}_{pqiI}(\xi^{m}_{t})C^{\tau+\Delta\tau}_{pqrs}(\xi^{m}_{t})B^{\tau+\Delta\tau}_{rsjJ}(\xi^{m}_{t})J^{\tau+\Delta\tau}(\xi^{m}_{t})w_{m}/8$$

and

$${}^{S}_{\text{geo}}K^{\tau+\Delta\tau}_{iIjJ} \approx \sum_{m=1}^{M} B^{\tau+\Delta\tau}_{rpiI}(\xi^{m}_{t})\sigma^{\tau+\Delta\tau}_{pq}(\xi^{m}_{t})B^{\tau+\Delta\tau}_{rqjJ}(\xi^{m}_{t})J^{\tau+\Delta\tau}(\xi^{m}_{t})w_{m}/8$$

are the expressions of the material and geometric stiffness contributions, respectively. We have in the above introduced C^{τ}_{pqrs} as the material tangent modulus at time τ and emphasize that the geometric contribution is in general only approximate and may depend in a complicated manner on the formulation of the element. The geometric stiffness matrix is used primarily for linear buckling analysis.

Stabilization

Stabilization forces are needed if the force from the physical stress is incapable of creating an element of full rank, i.e., without zero energy modes. A reduced integration element is a typical example of this, and the stabilization force is expressed in general terms as

$${}^{H}f^{\tau+\Delta\tau}_{iI} = f_{iI}(x^{\tau+\Delta\tau}_{jJ},...).$$

The force typically depends on the geometry and material properties and is nonzero only for modes that get no contribution to the internal force from physical stress. The stiffness matrix contribution is determined from the variation of this force, i.e.,

$${}^{H}K^{\tau+\Delta\tau}_{iIjJ}\delta x_{jJ} = \delta\,{}^{H}f^{\tau+\Delta\tau}_{iI}.$$

We also consider the special case of resultant or discrete elements, i.e., when ${}^{S}f^{\tau+\Delta\tau}_{iI} = 0$. In this case the stabilization force contribution represents the total internal force.

Implementation

User-defined elements in LS-DYNA can be implemented for hexahedral solids and quadrilateral shells. Up to a total of ten elements can be implemented in a single LS-DYNA executable for both explicit and implicit analyses.

For each user-defined element the interface consists of two subroutines, one routine for numerically integrated elements and one for any additional force and stiffness contributions (hourglass, drilling etc). Property parameters and history variables can be connected to the element by defining this in the keyword input file. Furthermore, extra degrees of freedom per node can be specified if the element formulation so requires. These extra degrees of freedom could for instance represent shell thickness or hydrostatic pressure.

For the integrated elements the user only needs to implement the gradient displacement matrix and jacobian, leaving LS-DYNA to take care of stress updates and force and stiffness assembly. This allows the user to take advantage of the rich supply of standard materials in LS-DYNA and also relieves him/her from the tedious work of assembling the global finite element matrices. It is of course possible to combine the user-defined element with a user-defined structural material in a usual fashion. The integration points can be placed at arbitrary locations within the element to avoid any forms of restrictions in this context. If the element requires stabilizing forces, these may be implemented in a separate routine to complete the definition of the element, or one could simply use the standard LS-DYNA stabilization (hourglass) routines.

For other element types, the global force vector and stiffness matrix must be implemented in a single routine. While this puts little or no limits in terms of what type of elements that can be defined, extra amount of work may be needed to define the element in question. For users especially concerned with the computational efficiency, even numerically integrated elements can be implemented in this way, although it may be somewhat awkward.

Subroutines

For the physical stress contribution the interfacing subroutine is

```
subroutine uXXX_bYYY(bmtrx,gmtrx,gjac,…
dimension bmtrx(nlq,3,3,*),gmtrx(nlq,3,3),gjac(nlq)
```

where `XXX` is substituted for `shl` or `sld` depending on the element type in question, and `YYY` is substituted for a number between `101` and `105`. This number should correspond to the value of ELFORM given on the *SECTION_... card, see below. The number `nlq` is a vector length parameter for the LS-DYNA vectorization and corresponds to the number of elements processed in a vector block. The output from the routine consists of the isoparametric gradient displacement matrix `bmtrx` corresponding to $\tilde{B}^{\tau}_{pqiI}$ in the theory presented above, and the jacobian matrix `gmtrx` corresponding to g^{τ}_{pq}. As an option, the `bmtrx` may correspond to the physical gradient displacement matrix B^{τ}_{pqiI}, and in this case the scalar jacobian `gjac`, mathematically corresponding to J^{τ}, must be computed. For shells the coordinate vector is expressed in the local coordinate system by default, but as an option it is expressed in the global system. The gradient displacement matrices must for this case be modified accordingly. Additional arguments to the subroutine include nodal coordinates, isoparametric coordinates, shape function values and derivatives, shell thicknesses, nodal fiber vectors and the coordinate transformation matrix computed by LS-DYNA. The output from this routine is, according to the theory, sufficient for LS-DYNA to completely determine the internal force contribution from the physical stress.

For the stabilization stress, the subroutine is

```
     subroutine uXXX_eYYY(force,stiff,ndtot,…
     dimension force(nlq,*),stiff(nlq,ndtot,*)
```

where again `XXX` and `YYY` is substituted as described for the previous subroutine above. The output from the routine consists of force vector `force,` mathematically corresponding to ${}^{H}f_{iI}^{\tau+\Delta\tau}$, and the stiffness matrix `stiff`, mathematically corresponding to ${}^{H}K_{iIjJ}^{\tau+\Delta\tau}$. For shells the force vector and stiffness matrix are expressed in the local coordinate system, but if the option of using the global representation of the coordinate vector is used, these must be expressed in the global system as well. Additional arguments to the subroutine are user defined property parameters and history variables. Otherwise the arguments are the same as for the previous subroutine.

Usage

A user-defined structural element is invoked on the *SECTION_... card in the keyword input file. The element formulation ELFORM in card 1 is specified to a number between 101 and 105 and the corresponding user-defined element subroutines are called for computing the internal force and tangent stiffness. Additional cards are used to define parameters associated with the interface, and the following is an extract of the keyword input for user-defined elements. The meaning of most parameters defined below comes naturally when reading the theory and implementation sections above, but for a complete description the reader is referred to the LS-DYNA Keyword User's Manual [1].

The *SECTION_SHELL card

This is the description that accompanies the information of the *SECTION_SHELL card.

Card 3

Define if and only if ELFORM=101,102,103,104 or 105

Variable	NIPP	NXDOF	IUNF	IHGF	ITAJ	LMC	NHSV	ILOC
Type	I	I	I	I	I	I	I	I
Default	0	0	0	0	0	0	0	0

Define NIPP cards according to the following format.

Variable	XI	ETA	WGT					
Type	F	F	F					

Define LMC property parameters using 8 parameters per card.

Variable	P1	P2	P3	P4	P5	P6	P7	P8
Type	F	F	F	F	F	F	F	F

VARIABLE	*DESCRIPTION*
NIPP	Number of in-plane integration points for user-defined shell (0 if resultant element)
NXDOF	Number of extra degrees of freedom per node for user-defined shell
IUNF	Flag for using nodal fiber vectors in user-defined shell EQ.0: Nodal fiber vectors are not used. EQ.1: Nodal fiber vectors are used
IHFG	Flag for using hourglass stabilization (NIPP.GT.0) EQ.0: Hourglass stabilization is not used EQ.1: LS-DYNA hourglass stabilization is used EQ.2: User-defined hourglass stabilization is used
ITAJ	Flag for setting up finite element matrices (NIPP.GT.0) EQ.0: Set up matrices wrt isoparametric domain EQ.1: Set up matrices wrt physical domain
LMC	Number of property parameters
NHSV	Number of history variables
XI	First isoparametric coordinate
ETA	Second isoparametric coordinate
WGT	Isoparametric weight
PI	Ith property parameter
ILOC	Flag for coordinate system option EQ.0: Pass variables in local coordinate system EQ.1: Pass variables in global coordinate system

The *SECTION_SOLID card

This is the description that accompanies the information of the *SECTION_SOLID card.

Card 2

Define if and only if ELFORM=101,102,103,104 or 105

Variable	NIP	NXDOF	IHGF	ITAJ	LMC	NHSV		
Type	I	I	I	I	I	I		
Default	0	0	0	0	0	0		

Define NIP cards according to the following format.

Variable	XI	ETA	ZETA	WGT				
Type	F	F	F	F				

Define LMC property parameters using 8 parameters per card.

Variable	P1	P2	P3	P4	P5	P6	P7	P8
Type	F	F	F	F	F	F	F	F

VARIABLE	*DESCRIPTION*
NIP	Number of integration points for user-defined solid (0 if resultant element)
NXDOF	Number of extra degrees of freedom per node for user-defined solid
IHFG	Flag for using hourglass stabilization (NIP.GT.0)
	EQ.0: Hourglass stabilization is not used
	EQ.1: LS-DYNA hourglass stabilization is used
	EQ.2: User-defined hourglass stabilization is used
ITAJ	Flag for setting up finite element matrices (NIP.GT.0)
	EQ.0: Set up matrices wrt isoparametric domain
	EQ.1: Set up matrices wrt physical domain
LMC	Number of property parameters
NHSV	Number of history variables
XI	First isoparametric coordinate
ETA	Second isoparametric coordinate
ZETA	Third isoparametric coordinate
WGT	Isoparametric weight
PI	Ith property parameter

Conclusions

A user-defined element interface is available in LS-DYNA from the release of v971. The interface allows users in academia and industry to verify research results on element technology in the context of general and complicated finite element applications. The high level of abstraction makes the interface easy to use and perfectly suited for educational purposes, but may result in some reduction in computational efficiency compared to the standard elements in LS-DYNA. The interface is developed continuously with the aim of improving efficiency and removing any form of restrictions for future releases of LS-DYNA. Moreover, any suggestions by users will be treated seriously in order to retain the competitiveness of LS-DYNA in this field and ensure customer satisfaction.

References

1. LS-DYNA Keyword User's Manual version 971, Livermore Software Technology Corporation, Livermore, 2006.
2. Nonlinear Finite Elements for Continua and Structures, T. Belytschko, W.K. Liu and B. Moran, John Wiley & Sons Ltd., 2000.