The Use of User Defined Elements and Extra Degrees of Freedom

Kelly S. Carney[1], Paul DuBois[1], and Thomas Borrvall[2]
[1]*Forming Simulation Technology LLC*
[2]*DYNAmore GmbH*

Abstract

The capabilities of the LS-DYNA® User Defined Features, allowing for modifications to LS-DYNA analyses, are powerful. It would be impractical to fully exercise their extensive capabilities, and so many of the features have also not been fully documented. Specifically, several undocumented actions are required to use User Defined Elements, when using extra degrees of freedom. This brief paper will explain the additional steps required for the LS-DYNA integrator to update the extra degrees of freedom, when using User Defined Elements. Additional details of extra degrees of freedom usage will also be given.

Introduction

The LS-DYNA User Defined Features allow for extensive modifications to multiple capabilities of LS-DYNA solutions. The provided resource allows for powerful tailoring of the delivered software to solving specific research challenges; both academic and industrial. The User Defined Features are documented in Appendix A of the Keyword User's Manual [1]. An overview of the wide variety of these user defined features in LS-DYNA was presented by Erhart [2]. The most popular user defined module amongst these features is the User Defined Material Model. Another powerful module, which is not as widely used, is the User Defined Element. In addition to Appendix A of the User's Manual, the User Defined Element is also described by Borrvall and Erhart [3].

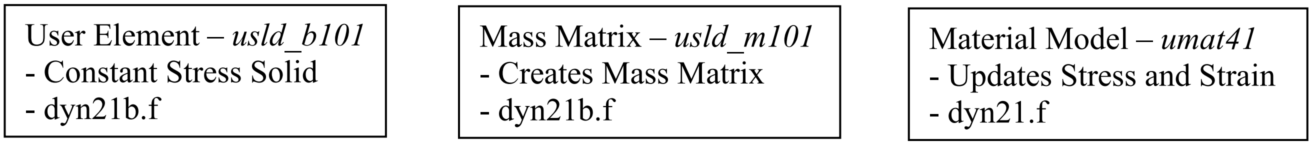
The User Defined Element module enables complete definitions of either shell or solid elements. Additional degrees of freedom at each element node may also be defined. By using the extra degrees of freedoms feature, higher-order or exotic elements can be implemented. Another advantage of defining extra degrees of freedom, is that typically internal element parameters may be passed to connected elements. So alternate uses of extra degrees of freedom, such as tracking thickness in a shell element or hydrostatic pressure for special case analyses, is possible. In addition, using extra degrees of freedom may be used for non-structural, multi-physics applications.

The User Defined Features are so extensive that it would be impractical to fully exercise them in demonstrations, and so many of the features have also not been fully documented. User Defined Elements with extra degrees of freedom are in the category of features that have not been fully documented. There are several required steps to enable the LS-DYNA integrator to update the values of the extra degrees of freedom. These actions are not discussed in the manual or the reference papers, and they include modifications to subroutines other than the user element subroutine. These required steps will be presented in this paper, as well as details of extra degrees of freedom usage, including input and output.

In this paper, the demonstration example will be an 8-node solid element. A multi-physics problem will be assumed, and the extra degrees of freedom will not represent structural-mechanical properties. As a result, individual terms of the isoparametric gradient-displacement (b) matrix and the Jacobian (g) matrix will not be altered in this example. However, the locations of these terms within the gradient-displacement matrix, and the size of the gradient-displacement matrix, will be modified to accommodate the extra degrees of freedom. The LS-DYNA version used for this work was R8.1.0, and some characteristics described may differ from other versions.

Subroutine Overview

The subroutines which can be altered, compiled. and linked to an object version of LS-DYNA are provided in two files; namely dyn21.f and dyn21b.f. The information necessary for obtaining and utilizing these files is provided in the User's Manual, and only details relevant to extra degrees of freedom will be reviewed. The primary subroutine which will be modified is the constant stress example in dyn21b.f, *usld_b101*. The corresponding mass matrix subroutine, *usld_m101*, also needs to be modified, and it is also in dyn21b.f. For outputting the values of history variables, subroutine *umat41* in dyn21.f will also be modified. In each discrete timestep, the *usld_b101* subroutine is called before the *umat41* subroutine. These subroutines are summarized in Figure 1.



User Element – *usld_b101*
- Constant Stress Solid
- dyn21b.f

Mass Matrix – *usld_m101*
- Creates Mass Matrix
- dyn21b.f

Material Model – *umat41*
- Updates Stress and Strain
- dyn21.f

Figure 1. Primary subroutines modified.

There are both user element history variables and user material model history variables. These are separate from each other, and only the user material model history variables can be output into the d3plot file. The user material model history variables must be written from subroutine *umat41*. Output of user element history variables into the d3plot files is not currently an option. So, if element values are desired as d3plot output, then those selected values must be copied into subroutine *umat41*, and written into the user material model history variables there. They can then be output to the d3plot file by setting NEIPH or NEIPS, on the *DATABASE_EXTENT_BINARY keyword, as is typical.

Subroutine *usld_b101* is called by the subroutine *usrsld_b*, which in turn is called by subroutine *usrsld*. In order to conveniently pass information, these upstream subroutines are also modified. These and other subroutines called by *usrsld,* which do not necessarily have to be modified, are shown in Figure 2. The subroutine *usld_m101* is called from outside of dyn21b.f, and is not called by *usrsld*. Since subroutine *umat41* is in dyn21.f, it is also not called by *usrsld*.

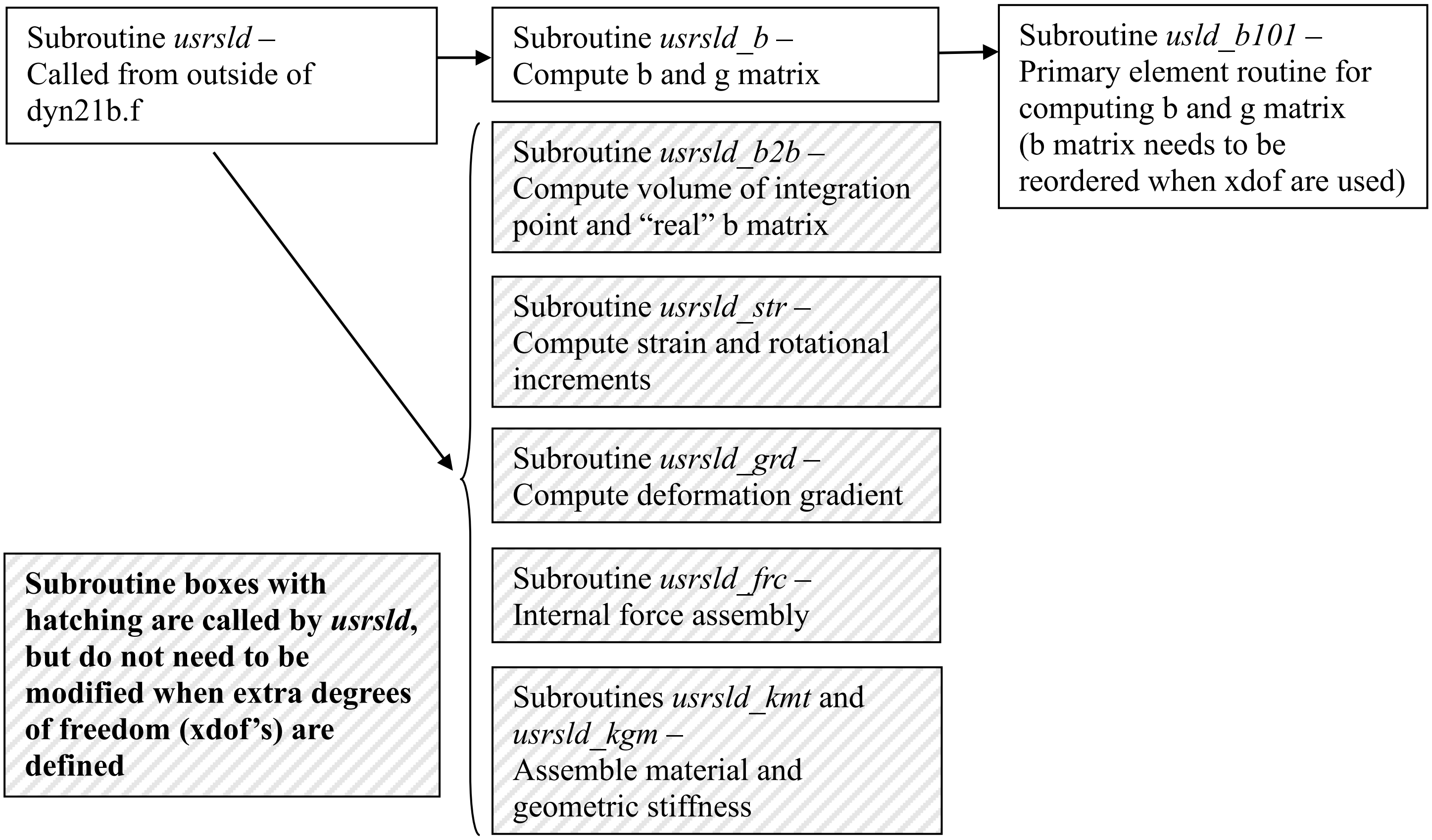

Subroutine usrsld – Called from outside of dyn21b.f
Subroutine usrsld_b – Compute b and g matrix
Subroutine usld_b101 – Primary element routine for computing b and g matrix (b matrix needs to be reordered when xdof are used)
Subroutine usrsld_b2b – Compute volume of integration point and "real" b matrix
Subroutine usrsld_str – Compute strain and rotational increments
Subroutine usrsld_grd – Compute deformation gradient
Subroutine usrsld_frc – Internal force assembly
Subroutines usrsld_kmt and usrsld_kgm – Assemble material and geometric stiffness
Subroutine boxes with hatching are called by usrsld, but do not need to be modified when extra degrees of freedom (xdof's) are defined

Figure 2. Subroutines called by subroutine *usrsld*.

Several matrices and parameters which are required for the use of extra degrees of freedom are not provided in the delivered example subroutine *usld_b101*. There are several other matrices not provided which are also convenient to have in *usld_b101*. Some of these matrices are available in *usrsld*, and some may be gained access to by adding common blocks. The *usrsld_b* call statement in subroutine *usrsld* was modified as follows;

```
      call usrsld_b(ietyp,xiipt,etaipt,zetaipt,lft,llt)
```

was changed to

```
      call usrsld_b(ietyp,xiipt,etaipt,zetaipt,lft,llt,
     + NXDOF,NHSV,CM,LMC)
```

where NXDOF is the number of extra degrees of freedom, NHSV is the number of user material history variables, CM is the material constants array, and LMC is the length of the material constants array. (The capitalized matrices and parameters in the call statement indicate an alteration to the delivered example subroutine.) The subroutine *usrsld_b* statement was also correspondingly augmented.

If the user material model history variables are desired in the user element, they may be introduced into the subroutine *usrsld_b* by adding the following common blocks:

```
        common/bux13loc/ eps(6),sig(6),hsv(NHISVAR),temps(nlq)
        common/aux14loc/sig1(nlq),sig2(nlq),sig3(nlq),sig4(nlq),sig5(nlq),
   +    sig6(nlq),epsps(nlq),hsvs(nlq,NHISVAR)
```

User element history variables are not being accessed in the provided demonstration user element in dyn21b.f. To utilize them, this common block can also be added to *usrsld_b*:

```
        common/bel8loc/uehisv(nlq,2*(NHISVUE+45))
```

In *usrsld_b*, the modified call statement which includes the number of extra degrees of freedom, the user material input, and the history variable matrices and variables is:

```
call usld_b101(cmtrx,hmtrx,hmtrx,
   1     xi,eta,zeta,
   2     n1,n2,n3,n4,n5,n6,n7,n8,
   3     dn1dxi,dn2dxi,dn3dxi,dn4dxi,
   4     dn5dxi,dn6dxi,dn7dxi,dn8dxi,
   5     dn1deta,dn2deta,dn3deta,dn4deta,
   6     dn5deta,dn6deta,dn7deta,dn8deta,
   7     dn1dzeta,dn2dzeta,dn3dzeta,dn4dzeta,
   8     dn5dzeta,dn6dzeta,dn7dzeta,dn8dzeta,
   9     x1,x2,x3,x4,x5,x6,x7,x8,
   +     y1,y2,y3,y4,y5,y6,y7,y8,
   +     z1,z2,z3,z4,z5,z6,z7,z8,
   +     xdof,dxdof,
   +     UEHISV,IUEHUP,NHSV-NADD1,
   +     HSVS,NHISVAR,NXDOF,FRC,
   +     CM,LMC,
   +     lft,llt)
```

with the three lines,

```
   +     UEHISV,IUEHUP,NHSV-NADD1,
   +     HSVS,NHISVAR,NXDOF,FRC,
   +     CM,LMC,
```

having been added. So, in subroutine *usrsld_b* user element history variables matrix is named UEHISV, and IUEHUP is the number of user element history variables. The subroutine *usld_b101* statement was correspondingly modified, but with several matrices being renamed:

```
      subroutine usld_b101(bmtrx,gmtrx,gjac,
        :
   + HSV,IHSV,NHSV,
   + HSVS,NHISVAR,NXDOF,FRC,
   + CM,LMC,
```

with HSV being the user element history variables, and IHSV is their number. In dyn21b.f, HSVS is the user material history variable matrix, and NHISVAR is their number. The FRC is the force matrix, and non-zero values in the force matrix must be defined for the extra degrees of freedoms to be updated by the LS-DYNA integrator.

Modifications to the *usld_b101* Subroutine

As described in the User's Manual, the gradient-displacement (b) matrix and the Jacobian (g) matrix are calculated in the subroutine *usld_b101.* When extra degrees of freedom are present, the size of the b matrix expands, and the location of the terms within the b matrix is altered. The b matrix is defined as

$$\mathrm{bmtrx}(*,\mathrm{i},\mathrm{j},\mathrm{k}) = b_{ijk}, \tag{1}$$

where

$$b_{ijk}u_k = \frac{\partial v_i}{\partial \zeta_j}\Delta t \tag{2}$$

and where u_k is the generalized local displacement, v_i is the velocity vector, ζ_i is the isoparametric coordinate, and Δt is the current time step. The i and j indices refer to translational directions in element coordinates. When extra degrees of freedom are defined the index k becomes,

$$k = (n + n_{xdof})(m - 1) + d \tag{3}$$

where n is 3 if only translational nodes are present, n_{xdof} is the number of extra degrees of freedom (NXDOF), m is the local node number, and d is the degree of freedom index. For example, the first eight terms of b_{ijk} (bmtrx), which are associated with the element x coordinate, are given in the example subroutine as

```
bmtrx(i,1,1,1)  = dn1dxi
bmtrx(i,1,1,4)  = dn2dxi
bmtrx(i,1,1,7)  = dn3dxi
bmtrx(i,1,1,10) = dn4dxi
bmtrx(i,1,1,13) = dn5dxi
bmtrx(i,1,1,16) = dn6dxi
bmtrx(i,1,1,19) = dn7dxi
bmtrx(i,1,1,22) = dn8dxi
```

where dn1dxi, dn2dxi, dn3dxi, dn4dxi, dn5dxi, dn6dxi, dn7dxi, dn8dxi are the nodal shape function derivatives with respect to the first parental coordinate. If one extra degree of freedom is defined, then theses first eight terms must be defined as

```
bmtrx(i,1,1,1)   = dn1dxi
bmtrx(i,1,1,5)   = dn2dxi
bmtrx(i,1,1,9)   = dn3dxi
bmtrx(i,1,1,13) = dn4dxi
bmtrx(i,1,1,17) = dn5dxi
bmtrx(i,1,1,21) = dn6dxi
bmtrx(i,1,1,25) = dn7dxi
bmtrx(i,1,1,29) = dn8dxi
```

If two extra degrees of freedom are defined, then these first eight terms must be defined as

```
bmtrx(i,1,1,1)   = dn1dxi
bmtrx(i,1,1,6)   = dn2dxi
bmtrx(i,1,1,11) = dn3dxi
bmtrx(i,1,1,16) = dn4dxi
bmtrx(i,1,1,21) = dn5dxi
bmtrx(i,1,1,26) = dn6dxi
bmtrx(i,1,1,31) = dn7dxi
bmtrx(i,1,1,36) = dn8dxi
```

If three extra degrees of freedom are defined, then theses first eight terms must be defined as

```
bmtrx(i,1,1,1)   = dn1dxi
bmtrx(i,1,1,7)   = dn2dxi
bmtrx(i,1,1,13) = dn3dxi
bmtrx(i,1,1,19) = dn4dxi
bmtrx(i,1,1,25) = dn5dxi
bmtrx(i,1,1,31) = dn6dxi
bmtrx(i,1,1,37) = dn7dxi
bmtrx(i,1,1,43) = dn8dxi
```

and so forth, noting that the terms themselves did not change; only their location within bmtrx. In subroutine *usld_b101*, the locations of all the shape functions must be updated; not just the first eight shown as an example above. With the use of a higher-order element formulation, additional shape function terms must also be defined here.

Terms associated with the extra degrees of freedom must also be included in the force matrix. If these terms are not included, the LS-DYNA integrator will not update the values of the extra degrees of freedom. The dimension and the placement of terms within the force matrix is not the same as in the bmtrx or in the xdof matrix. The xdof matrix is where the time-dependent values of the extra degrees-of-freedom are provided. The xdof matrix has indices,

xdof(i,m,nxdof)

where *m* is the local node number within the element, and nxdof is extra degree of freedom number. The dxdof matrix is also available, and it provides the total deformation, not the incremental deformation. For an 8-node solid brick element, and using 3 extra degrees of freedom, the xdof matrix has dimensions,

xdof(i,8,3)

while the corresponding force matrix has dimensions

$$\mathrm{frc}(i,m*(n + \mathrm{nxdof}))$$

So, for an 8-node solid brick element and 3 extra degrees of freedom, results in a force matrix with dimensions of

$$\mathrm{frc}(i,48)$$

For the same example, the arrangement of terms within one column of the force matrix is as follows:

$$\begin{array}{llllll}
f_{x1} & f_{y1} & f_{z1} & f_{xdof(1)1} & f_{xdof(2)1} & f_{xdof(3)1}, \\
f_{x2} & f_{y2} & f_{z2} & f_{xdof(1)2} & f_{xdof(2)2} & f_{xdof(3)2}, \\
f_{x3} & f_{y3} & f_{z3} & f_{xdof(1)3} & f_{xdof(2)3} & f_{xdof(3)3}, \\
f_{x4} & f_{y4} & f_{z4} & f_{xdof(1)4} & f_{xdof(2)4} & f_{xdof(3)4}, \\
f_{x5} & f_{y5} & f_{z5} & f_{xdof(1)5} & f_{xdof(2)5} & f_{xdof(3)5}, \\
f_{x6} & f_{y6} & f_{z6} & f_{xdof(1)6} & f_{xdof(2)6} & f_{xdof(3)6}, \\
f_{x7} & f_{y7} & f_{z7} & f_{xdof(1)7} & f_{xdof(2)7} & f_{xdof(3)7}, \\
f_{x8} & f_{y8} & f_{z8} & f_{xdof(1)8} & f_{xdof(2)8} & f_{xdof(3)8}
\end{array}$$

with the force values for the translational degrees of freedom and the extra degrees of freedom being interspersed. While the terms within the xdof matrix are arranged as,

$$\begin{pmatrix}
\mathrm{xdof}(1)_1 & \mathrm{xdof}(1)_2 & \mathrm{xdof}(1)_3 & \mathrm{xdof}(1)_4 & \mathrm{xdof}(1)_5 & \mathrm{xdof}(1)_6 & \mathrm{xdof}(1)_7 & \mathrm{xdof}(1)_8 \\
\mathrm{xdof}(2)_1 & \mathrm{xdof}(2)_2 & \mathrm{xdof}(2)_3 & \mathrm{xdof}(2)_4 & \mathrm{xdof}(2)_5 & \mathrm{xdof}(2)_6 & \mathrm{xdof}(2)_7 & \mathrm{xdof}(2)_8 \\
\mathrm{xdof}(3)_1 & \mathrm{xdof}(3)_2 & \mathrm{xdof}(3)_3 & \mathrm{xdof}(3)_4 & \mathrm{xdof}(3)_5 & \mathrm{xdof}(3)_6 & \mathrm{xdof}(3)_7 & \mathrm{xdof}(3)_8
\end{pmatrix}$$

and the translational degrees of freedom are stored in a different matrix.

Modifications to the *usld_m101* Subroutine

For the LS-DYNA integrator to update the values of the extra degrees of freedom, mass values associated with these degrees of freedom must be defined. However, the example subroutine *usld_m101* is a placeholder, with no coding included to process keyword input. So, defining masses using *ELEMENT_INERTIA or *ELEMENT_MASS keyword input, unlike the implication of the User's Manual, does not add mass values to extra degrees of freedom.

An addition as simple as,

```
do i=1,8
        do j=1,nxdof
        w(j,i)=1.
        enddo
enddo
```

where w(j,i) is the mass matrix for the extra degrees of freedom is sufficient to make the LS-DYNA integrator update the values of the extra degrees of freedom. The physics of the problem being modeled may require a different mass matrix. Integration with the LS-DYNA integrator is now complete.

Input and Output

The keyword input for user elements with extra degrees of freedom is described in the User's Manual. An example, with a single element and accompanying *SECTION and *NODE input cards, is as follows:

```
*SECTION_SOLID
$ SECID  ELFORM
     101       101
$    NIP   NXDOF    IHGF    ITAJ     LMC     NHSV
       1         3       1       0
$     XI        ETA   ZETA     WGT
      0.          0.      0.      8.0
$
*ELEMENT_SOLID_DOF
$ EID   PID
    1     1
$  N1    N2    N3    N4    N5    N6    N7    N8
    1     2    10     9    17    18    26    25
$             NS1   NS2   NS3   NS4   NS5   NS6   NS7   NS8
              101   102   110   109   117   118   126   125
$
*NODE_SCALAR_VALUE
 $ NID  INITIALX   INITIALY   INITIALZ NDOF
   101       4.369      -30.00        298.0      3
   102       4.369      -30.00        298.0      3
   109       4.369      -30.00        298.0      3
   110       4.369      -30.00        298.0      3
   117       0.110      -30.00        298.0      3
   118       0.110      -30.00        298.0      3
   125       0.110      -30.00        298.0      3
   126       0.110      -30.00        298.0      3
```

In LS-PrePost® these scalar node values are displayed at the value of the extra degree of freedom. For example, node 101 in the above example would be displaced initially in the location {4.369, -30.00, 298.00}, which may be a long distance from any finite element mesh. So, these scalar value nodes are not associated with any element in LS-PrePost. As a result, locating and selecting these nodes for post-processing is sometimes a task.

Summary

Using extra degrees of freedom, in association with User Defined Element formulations, is a powerful tool in the LS-DYNA User Features suite. However, some necessary steps for their use are not documented in the LS-DYNA User's Manual. These additional necessary actions are presented in this paper. By following the somewhat arbitrary rules outlined here, usage of extra degrees of freedom can be made relatively more straightforward. The example demonstrating these actions is an isoparametric 8-node solid element, but the procedures can easily be adopted for other element formulations.

Acknowledgement

The authors would like to thank Shriram Santhanagopalan of the National Renewable Energy Laboratory for sponsoring the research that made the investigation into using extra degrees of freedom in User Defined Elements necessary.

References

[1] Livermore Software Technology Corporation, *LS-DYNA Keyword User's Manual*, 2019.
[2] Erhart, Tobias, *An Overview of User-Defined Interfaces in LS-DYNA*, 2010 German LS-DYNA Forum, Bamberg, Germany.
[3] Borrvall, Thomas, and Erhart, Tobias, *A User-Defined Element Interface in LS-DYNA v971*, 2006 German LS-DYNA Forum, Ulm, Germany.