

LS-DYNA Software Revision 12.0.0

Release Notes



Oasys

LS-DYNA ENVIRONMENT

Contents

Contents	1
1 LS-DYNA Release R12.0.0.....	3
1.1 Introduction to R12.0.0.....	3
1.2 License.....	3
1.3 Documentation	3
1.4 Release notes for R12.0.0	3
1.4.1 *AIRBAG	4
1.4.2 ALE.....	5
1.4.3 *BATTERY	7
1.4.4 *BOUNDARY.....	8
1.4.5 BLAST.....	9
1.4.6 CESE (Compressible Fluid Solver).....	9
1.4.7 *CHEMISTRY.....	9
1.4.8 Dual CESE Compressible Fluid Solver	10
1.4.9 CONTACT.....	11
1.4.10 *CONSTRAINED.....	15
1.4.11 *CONTROL.....	17
1.4.12 Discrete Element Method.....	19
1.4.13 EFG (Element Free Galerkin)	19
1.4.14 *ELEMENT	19
1.4.15 EM (Electromagnetic Solver)	22
1.4.16 *FATIGUE	23
1.4.17 Forming Analysis	24
1.4.18 *FREQUENCY_DOMAIN	25
1.4.19 ICFD (Incompressible Fluid Solver)	28
1.4.20 Implicit (Mechanical) Solver.....	30
1.4.21 *INITIAL.....	35
1.4.22 Isogeometric Analysis (IGA)	37
1.4.23 *LOAD	38
1.4.24 *MAT and *EOS	39
1.4.25 MPP.....	50
1.4.26 Output.....	50
1.4.27 Restarts	52
1.4.28 *SENSOR	53

1.4.29	SPG (Smooth Particle Galerkin).....	54
1.4.30	SPH (Smooth Particle Hydrodynamics).....	54
1.4.31	Thermal Solver	57
1.4.32	XFEM (eXtended Finite Element Method).....	59
1.4.33	Miscellaneous	59

1 LS-DYNA Release R12.0.0

1.1 Introduction to R12.0.0

This file constitutes revision 1 of the release notes for LS-DYNA version R12.0.0.

Changes in revision 1 are:

- Fixed typos.
- Added introductory sections I, II, and III which were inadvertently left out in revision 0.

1.2 License

The string "REVISION 12" must appear in the LS-DYNA license file in order to run version R12.0.0.

1.3 Documentation

Until such time the R12.0 User's Manual is posted at www.lstc.com/download/manuals, the DRAFT User's Manual at www.lstc.com/download/manuals may be used as a temporary substitute, but please be aware that there may be features mentioned in the DRAFT User's Manual which are not implemented in version R12.0.0.

1.4 Release notes for R12.0.0

The remainder of this file provides a brief description of new features and enhancements in version 12.0.0. Some bug fixes are also described, some of which may also have been included in other recent releases.

The items are arranged by category. Understand that in many cases, a particular item may pertain to more than one category, but in the interest of brevity, each item is listed only once, under a single category.

Excluding the "Miscellaneous" category, the categories are arranged alphabetically.

The categories are:

*AIRBAG

ALE

BATTERY (Electrochemistry Solver) <<< NEW category for R12.0.0

*BOUNDARY

Blast

CESE Compressible Fluid Solver

*CHEMISTRY

Dual CESE Compressible Fluid Solver <<< NEW category for R12.0.0

CONTACT

*CONSTRAINED

*CONTROL

Discrete Element Method

EFG (Element Free Galerkin)

*ELEMENT

EM (Electromagnetic Solver)

*FATIGUE

Forming Analysis
*FREQUENCY_DOMAIN
ICFD (Incompressible Fluid Solver)
Implicit (Mechanical) Solver
*INITIAL
Isogeometric Analysis (IGA)
*LOAD
*MAT and *EOS
MPP
Output
Restarts
*SENSOR
SPG (Smooth Particle Galerkin)
SPH (Smooth Particle Hydrodynamics)
Thermal Solver
XFEM (eXtended Finite Element Method)
Miscellaneous

1.4.1 *AIRBAG

Fix a bug for the output of area and leakage information for all parts constituting a control volume airbag. The bug occurred when an airbag was comprised of more than 10 parts.

*AIRBAG_HYBRID_CHEMKIN: fix an MPP bug introduced in r117881 that results in incorrect airbag pressure.

Remove the interaction between *AIRBAG_REFERENCE_GEOMETRY and *INITIAL_FOAM_REFERENCE_GEOMETRY so that they can only be used to define the reference geometry of shell and solid elements respectively.

*AIRBAG_PARTICLE (CPM):

- New keyword *DEFINE_CPM_NPDATA to support more part-specific input for *AIRBAG_PARTICLE. Invoked by NPDATA>0 and STYPEH = 2 or 3. Among other things, this new feature allows user to control smoothing algorithm for applying particle to fabric impulse.

- Add new limit checking and self-adjusting algorithm for 4th order polynomial nonmonotonic function of nonlinear CP to avoid incorrect result.

- Support inflator mass flowrate curve (LCTi) using *DEFINE_CURVE, *DEFINE_CURVE_FUNCTION and *DEFINE_FUNCTION.

- New feature for *DEFINE_CPM_VENT for pushout vent to allow user to apply the ambient pressure when internal parts extend out from the external vent.

- Support C23 (discharge coefficient) as function of vent area.

- Support Autoliv porous leakage model (FVOPT=-1,-2) under CPM and CPM+UP switch capabilities.

- Add tire inflation capability under CPM method to maintain the target tire pressure during the initial setup.
- Support inflator volume evaluated from current geometry while using user-defined inflator chamber.
- Fix bug for airbag with solid parts inside. The volume from those parts are excluded from bag volume.

New keyword `*CONTROL_AIRBAG` for CV (Control Volume) closed volume check.

Support multiple airbags when not all airbags have a reference geometry (`*AIRBAG_REREFERENCE_GEOMETRY`).

1.4.2 ALE

`*INITIAL_ALE_MAPPING`: Add a parameter `SYM` to apply specific mapping rules to elements and nodes outside the mesh of the previous run that wrote the mapping file to be used in the current run.

`*ALE_MAPPING`: Add new keyword to map data during a run (not just initially like with `*INITIAL_ALE_MAPPING`). A particular state from the mapping file is read.

`*INITIAL_VOLUME_FRACTION_GEOMETRY` and `CNTTYP=7`: Allow the user to define the geometry using `*DEFINE_FUNCTION`.

`*INITIAL_DETONATION`: If `PID<-1`, `|PID|` is the ID of a `*SET_PART`.

`*INITIAL_HYDROSTATIC_ALE`, `*ALE_AMBIENT_HYDROSTATIC`: Account for the compaction to compute the initial and ambient hydrostatic pressure for `*EOS_MIE_GRUNEISEN`.

`*ALE_PRESCRIBED_MOTION` and `*BOUNDARY_AMBIENT`: Add a new parameter `SIDR` in both keywords to control their use during the dynamic relaxation phases (similar to `SIDR` in `*DEFINE_CURVE`).

`*BOUNDARY_AMBIENT`: Set birth and death times with the first and last abscissa of `*DEFINE_CURVE` for the internal energy and relative volume time curves (`LCID1` and `LCID2`).

`*CONTROL_ALE`: Add a new variable `BNDFLX` to select only the ALE groups (`*SET_MULTI-MATERIAL_GROUP_LIST`) that can flux in when these groups are in the ALE elements along the mesh boundaries. Set `BNDFLX=-1` to forbid any influx along free mesh boundaries.

`*ALE_BURN_SWITCH_MMG`: Implement this keyword to allow the user to implement his own burn models.

`*ALE_MESH_INTERFACE`: Implement this new keyword to mesh material interfaces with triangular shells (the material volume can also be meshed with tetrahedra).

*CONSTRAINED_LAGRANGE_IN_SOLID: Implement a 2D version of the implicit thermal ALE coupling.

*CONSTRAINED_LAGRANGE_IN_SOLID_EDGE with CTYPE=5: Renew the coupling interface along the outside shell edges involved in the ALE coupling after shell erosion.

*CONSTRAINED_LAGRANGE_IN_SOLID with $4 \leq CTYPE \leq 6$: Write in and read back from d3full the FSI relative displacements and other coupling arrays.

*CONTROL_SEGMENTS_IN_ALE_COUPLING: New keyword that deactivates segments in the ALE penalty coupling (CTYPE=4,5,6 in *CONSTRAINED_LAGRANGE_IN_SOLID) if the segments are face to face. If variable SYM=1, a segment normally constrained is excluded from the coupling (similar to ISYM in *CONTROL_CONTACT).

*DATABASE_BINARY_FSIFOR and *CONTROL_MPP_DECOMPOSITION_NUMPROC: Prevent fsifor from being corrupted by a MPP job starting right after MPP decomposition.

*DATABASE_BINARY_FSIFOR and *SECTION_ALE2D: Output the interface force file (fsifor) in 2D (meshfringe in LS-PrePost should be used to see the coupling force and pressure distributions along the segment edges).

*DATABASE_FSI: Output the center of pressure at fx-lc,fy-lc,fz-lc (or gx,gy,gz) in dbfsi if a node set (NDSETID) is provided in *DATABASE_FSI.

*SECTION_ALE1D and *DATABASE_EXTENT_BINARY BEAMIP>0: Access auxiliary and history variables in d3plot for ALE 1D elements (like NEIPH for solids and NEIPS for shells).

*DATABASE_TRACER and TRACK=2: The tracer moves with the ALE mesh.

*DATABASE_TRACER and ALE mapping: Output the tracer final locations in a keyword format (to be included in the next run initialized by a mapping).

A new ellipsoid geometry option is added in

*INITIAL_VOLUME_FRACTION_GEOMETRY.

*INITIAL_HYDROSTATIC_ALE now supports ALE single material with void formulation, that is, ELFORM 12 in *SECTION_SOLID.

Structured ALE (S-ALE):

- Support *EOS_MURNAGHAN to model weakly incompressible water.

- *ALE_STRUCTURED_MESH_VOLUME_FILLING implemented to fill ALE fluids into the initial S-ALE mesh and sidestep *INITIAL_VOLUME_FRACTION_GEOMETRY.

- *ALE_STRUCTURED_FSI implemented to perform ALE fluid-structure interaction with structured ALE mesh. It could effectively stop the leakage often observed when using *CONSTRAINED_LAGRANGE_IN_SOLID.

1.4.3 *BATTERY

Updates to the *INTERFACE_LINKING_NODE<_SET>_LOCAL option so that the coordinate system transformation behaviour makes more sense. (See also "Battery module Release" in the "EM (Electromagnetic Solver)" section.)

The *BATTERY family of keywords invokes the new electrochemistry solver, which is only available in double precision executables. *BATTERY keywords are documented in Volume III of the LS-DYNA Keyword User's Manual.

The keywords starting with *BATTERY refer to and control the problem set up for detailed one-dimensional electrochemistry modelling of battery cells. This is intended to be used for battery-thermal-structure-interaction problems.

Two Lithium Ion Battery (LIB) models are available via two one-dimensional full cell models.

- Single insertion which has this structure:

 - Anode Lithium metal electrode/Separator/Cathode composite electrode

- Dual insertion which has this structure:

 - Anode composite electrode/Separator/Cathode Composite electrode

These two cell models are based on an electrochemical model with various anode, separator, and cathode material properties. Therefore, the user can select a material property for their anode and cathode including an open-circuit potential (OCP). For the separator, the user can also select a material property among the different electrolytes which include all of the transport properties necessitated in battery simulation. Note that all transport properties are uniquely designed for each battery type.

The selection of the single insertion or dual insertion model is done in keyword *BATTERY_ECHEM_CONTROL_SOLVER. Here, users can select the battery run mode such as Galvanostatic or Potentiostatic, the number of cycles, and termination controls. Please see Vol III of the keyword manual for more details.

With the *BATTERY_ECHEM_CELL_GEOMETRY keyword, users can choose the length of each layer in a single cell including both sides of the current collectors, and the corresponding total number of mesh elements can also be input.

After this is done, then users need to set up their own values of all variables in the material cards for all layers: anode, cathode, and separator. The three main keywords are,

- *BATTERY_ECHEM_MAT_ANODE

- *BATTERY_ECHEM_MAT_CATHODE

- *BATTERY_ECHEM_MAT_ELECTROLYTE

Here, first the OCP ID must be carefully selected (please refer to the keyword manual) and then Coulomb capacity, initial state of charge, thermodynamics data, and porous media data, respectively.

In the *BATTERY_ECHEM_THERMAL keyword card, the user can set the initial temperature and other thermal properties. However, please note that this card is ignored when the battery solver is coupled to the thermal and thermal-mechanical solvers.

Please note that the purpose of these 1D solvers is give the user a tool to test their OCP, material properties, and transport battery properties. For coupling the electrochemical battery solver to the thermal-mechanical solver, users must design their own mesh system with a different part number and must assign the individual part numbers for each solver. For example, if the electrochemical battery solver needs to solve within a specific part (all battery regions), then the user must assign that part number in the first variable column in each of these material keywords:

*BATTERY_ECHEM_MAT_ANODE,
*BATTERY_ECHEM_MAT_CATHODE, and
*BATTERY_ECHEM_MAT_ELECTROLYTE.

The same part number should be assigned for use by the structural thermal and mechanical solvers. Structural parts outside of the battery would have other part numbers. Please note that the 1D electrochemistry solver solves all elements assigned to the part numbers for the battery, and also note that this battery solver has its own 1D mesh system in each battery structural element, while the thermal-mechanical solver has its own 3D mesh system.

Currently however, only the dual insertion model is coupled with the LS-DYNA thermal-mechanical solvers. This model covers all Lithium-Ion batteries used by the battery manufacturers of cellular phones, and automotive industries batteries.

So, in version R12.0.0, users can simulate how the Lithium-Ion battery responds thermally and/or mechanically when external heat sources or impacting forces are applied to the battery pack. The battery model can be a single battery cell, a cell stack which has multiple cells connected (over 100 cells), and even multiple cell stacks connected in parallel. By controlling the solution scheme of the structural solvers, users can calculate the structural effects for a certain period of time of battery operation such as initial, middle, and end of state of battery discharging or charging.

For more detail about electrochemical battery theory, please refer to the LS-DYNA multiphysics theory manual. For keyword setup, please refer to the multiphysics (Vol III) keyword manual. In addition, for interested users, please request some sample keyword decks from LS-DYNA technical support, or email directly to kyoungsu.im@ansys.com.

1.4.4 *BOUNDARY

Fix bug in *BOUNDARY_PRESCRIBED_MOTION which could cause velocity boundary conditions to be incorrectly handled for dof= \pm 4 or 8 if the node has more than one velocity boundary condition, e.g., one during dynamic relaxation and another during transient analysis.

Corrected MPP communication error associated with Implicit loading constraints for *BOUNDARY_PRESCRIBED_MOTION_FINAL_GEOMETRY.

For convection, flux and radiation boundary conditions, the parameter PSEROD on *BOUNDARY_... specifies a part set for which new segments exposed to the environment due to solid element erosion will inherit the boundary condition data.

New option BNDOUT2DYNAIN on *BOUNDARY_PRESCRIBED_MOTION_RIGID, allows for output of reaction force to dynain for use in subsequent simulations.

SPC2BND on *CONTROL_OUTPUT which will convert constraints on *MAT_RIGID to prescribed motions, for access to reaction forces in bndout.

Fix *BOUNDARY_NON_REFLECTING to have the correct velocity averaging and force redistribution for triangular segments.

*CONTROL_ADAPTIVE: Boundary conditions *BOUNDARY_RADIATION_SET and *BOUNDARY_CONVECTION_SET are updated in accordance with mesh changes due to 3D tet adaptivity under the condition that the segment set(s) are defined using *SET_SEGMENT_GENERAL with OPTION=PART.

1.4.5 BLAST

Fix blast wind velocity field for *LOAD_BLAST_ENHANCED BLAST=4. Previously velocity was always the ground-reflected wave. Now, if a segment is not in the Mach stem region, the blast wind comes from both the incident-only and ground-reflected waves.

Fix a bug which could potentially affect results for models which contained both *LOAD_BLAST and *LOAD_BLAST_ENHANCED with TBO.ne.0. The time offsets could get mixed up or ignored.

1.4.6 CESE (Compressible Fluid Solver)

For the solvers not performing chemistry calculations, a switch has been made to a positivity-preserving method. For most problems, this will lead to a smaller time step, depending upon the shape of the smallest mesh element. Note that this method guarantees that density and internal energy remain positive.

Several bug fixes were made that make these solvers more stable.

For the CESE moving mesh conjugate heat transfer solver, fixed several bugs, including some MPP bugs that depended upon the mesh decomposition.

1.4.7 *CHEMISTRY

The inflator model can be extended up to a 5-chamber model. Previously, it was limited to only a 3-chamber model which consists of the combustion chamber, gas chamber, and tank. In this version, we added two more gas plenum chambers to control both the speed of the gas and pressure into the tank.

Resolved issues with back flows when the pressure of the downstream gas chamber is higher than the upper chamber by correcting the pressure equation.

Updated the procedure for computing/loading an initial blast profile as an initial condition of a detonation or deflagration.

Fixed some problems in the axisymmetric combustion solver.

Chemically reacting FSI flow problems using the Euler equation solver now work for any problem up to 60 species in the combustion gas. FSI with this Euler solver is strong and stabilized.

1.4.8 Dual CESE Compressible Fluid Solver

The *DUALCESE family of keywords invokes the new dual Conservation Element/Solution Element (dual CESE) compressible fluid solver. This solver is only available in double precision executables.

The *UNIT family of keywords is introduced which provides a coherent way to specify units in a LS-DYNA problem. For now, these keywords only work with the dual CESE solver but may be extended to other solvers in the future.

Dual CESE Solver Characteristics:

- Explicit
- Double precision
- Dynamic memory handling
- SMP and MPP
- 3D solver / special case 2D solver and 2D axisymmetric solver

Note: 2D axisymmetric solver is only done for the CFD solver, not yet for FSI-ibm or FSI-mmm solver cases.

- Automatic coupling with the LS-DYNA structural solver
- Eulerian fixed mesh or moving mesh (Either type input with *DUALCESE_ELE2D or *DUALCESE_ELE3D cards, or via *MESH cards for a tetrahedral mesh)

Dual CESE Solver Main Features:

- CFD solver is based upon the dual CESE (Conservation Element / Solution Element) method that is a new version of CESE technology with enhanced accuracy and robustness
- Highly accurate shock wave capturing
- Embedded (immersed) boundary approach or moving mesh (fitted) approach for FSI problems
- Mesh can be broken up into regions/parts with a different solver per region, with the intent to minimize the region where a moving mesh FSI technique is used, or where an immersed boundary FSI method is used.

This capability is intended to help optimize solver performance.

- Complex fluid equations of state (EOS) are now available through the REFPROP and COOLPROP EOS libraries. This is only supported with *DUALCESE models.
- Bi-cubic table look-up systems are now available for the REFPROP and COOLPROP libraries to dramatically speed up the evaluation of thermodynamic quantities.

Dual CESE Solver Applications (Non-exhaustive) :

- FSI problems
- Shock wave capturing
- Shock/acoustic wave interaction

Planned future features of Dual CESE Solver:

- FSI with material erosion (as done in the *CESE immersed boundary FSI solver)
- Coupled stochastic fuel spray solver (See *STOCHASTIC keywords)
- Coupling with chemistry (See *CHEMISTRY keywords) solver
- Multi-species transport
- Conjugate heat transfer
- Cavitation model

1.4.9 CONTACT

Fix old issue for *CONTACT_ENTITY type 9 to properly handle rotations when the entity is not centred on (0,0,0).

Fix for spotwelds improperly deleted due to rigid body conflict when an IPBACK *CONTACT interface is in effect. The inability of the constraint-based contact interface to tie should not cause the weld to be deleted if the penalty side ties (MPP only).

Fix SMP *CONTACT_2D_NODE_TO_SOLID. Friction was being ignored.

Added a warning that *CONTACT_TIEBREAK_NODES_TO_SURFACE_ID is not supported for implicit computations.

*CONTACT_2D_..._THERMAL is now available in MPP.

Fix an SMP bug for ERODING contact of solid elements, which could result in erroneous contact thickness.

Fix seg fault or incorrect frictional behaviour when using _THERMAL_FRICTION option for *CONTACT_AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE and *CONTACT_AUTOMATIC_SURFACE_TO_SURFACE (SMP only).

Fix instability when using *CONTACT_TIEBREAK_NODES_TO_SURFACE (SMP only).

Fix incorrect *CONSTRAINED_TIE-BREAK behaviour when the master node is the last node in the input after sorting.

Fix failure to detect contact between beam and shell edge when using *CONTACT_AUTOMATIC_BEAMS_TO_SURFACE and the beam diameter is large compared to the segment size (SMP only).

Fix *CONTACT_AUTOMATIC_GENERAL for spot weld beams when using SSID=0, i.e. all parts included in the contact, and CPARAM8=2.

Implement unloading curve, UNLCID, for options FCM=2/3 in *CONTACT_RIGID_(OPTION).

FTORQ=1 and 2 (Opt. Card E in *CONTACT), affecting transmittal of moments, are now implemented in SMP for

*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK and *CONTACT_AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK.

FTORQ=2 is now implemented in SMP for

*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE and *CONTACT_ONE_WAY_SURFACE_TO_SURFACE.

Added contact energy density to the intfor database for segment-based (SOFT=2) contact. This option is invoked by NENG=1 on *DATABASE_EXTENT_INTFOR.

Added support for ORTHO_FRICTION option for segment-based (SOFT=2) contact. This option is available for all versions, SMP, Hybrid, and MPP with and without the groupable option, and for all supported CONTACT keywords.

Added options to limit the scope the DPRFAC option of segment-based (SOFT=2) contact. When DPRFAC is set to a value greater than or equal to 1.0, then DPRFAC is the ID of either a shell set, a segment set, or a part set. When both segments are in the set, then the DPRFAC option is active for that segment pair. If either segment is not in then set, the DPRFAC is inactive. When searching for the set, the search will proceed in the order of shell set, then segment set, and finally part set. The first set that is found with a matching ID will be used. The set attribute DA1 will be read and used as the DPRFAC value.

Added support for VSF parameter on Card 3 of *CONTACT when using SOFT=2 contact.

Added support for FTORQ=2 (*CONTACT, Card E) for SMP and MPP segment-based (SOFT=2) contact. Setting FTORQ=2 adds torque to master nodes to counteract the moment that is created by frictional contact forces due to the contact surface being offset from the mid-plane of the element. This added torque reduces the net moment from contact to zero.

Added keyword *DEFINE_FRICTION_SCALING which allows shell segments to have friction coefficients on the inner and outer face (SOFT=2 contact only). Independent scale factors on the inner and outer face scale the nominal friction coefficient. This was motivated by airbags which have liners on the inside of the bag thus creating a smoother (slicker) surface on the inside as compared to the outside.

Added a new variable EDGEK on Card C of *CONTACT which scales edge-to-edge contact stiffness for SOFT=2 contact having DEPTH=5, 15, 25, or 35.

Added control of thick segment checking using the SNLOG flag on *CONTACT. A thick segment check has been done quietly throughout the history of segment-based (SOFT=2) contact. A search along the contact surface is made looking for thick pairs that are too close, and those pairs are removed from contact. This is done to prevent possible unstable behaviour that could occur if the offsets of nearby thick segments come into contact at a bend in the mesh. When SNLOG=0 or 1, the thick segment check will be done quietly as before. Set to 2, the check will be done and a warning 21465 will be written to report the segment pairs that are removed from contact. Set to 3, the thick segment check will be omitted.

Enabled segment-based contact (SOFT=2) to work with segments attached to cohesive elements with zero volume.

Enabled *CONTACT_ENTITY to work with thick shell elements. It was previously error terminating in the input phase.

Enabled SINGLE_SURFACE segment-based (SOFT=2) contact to output gap values to the intfor output database. Previously, output of the gap values worked only for the non-SINGLE_SURFACE contact types.

Enabled the SPOTHIN and SWRADF variables on *CONTROL_CONTACT to work when spot welds are modelled by sharing nodes with the shell parts that are welded. This has not

worked until now because the search for affected shells was based on tied contact, and tied contact is not needed when nodes are shared.

Enabled `*CONTACT_2D_AUTOMATIC` to work with erosion of 2D solid elements that use `*MAT_081` or `*MAT_082`.

Enabled multiple instances of `*CONTACT_2D_AUTOMATIC_TIED` to be reliably used in a model.

Improved the `PSTIFF` option of segment-based contact to make the segment mass better match the nodal masses. The main effect is to increase the stiffness for segments on the edge of the mesh. For a regular mesh with no mass scaling or lumped nodal masses, the `PSTIFF` option now calculates the same segment mass as the default method based on segment volume and density.

Fixed segment-based contact for adaptive remeshing and full deck restarts.

Fixed the coefficient of restitution option (`ICOR`) in segment-based contact when it is used along with `DPRFAC` (Depth of Penetration Reduction FACTor).

Fixed `*CONTACT_2D_FORCE_TRANSDUCER` with both slave and master sides defined when they are used with `*CONTACT_2D_AUTOMATIC` in MPP. A memory clobber was likely and also output was incorrect.

Fixed MPP wear calculation in segment-based contact. A mix-up of ID numbers means that the wear was only working when contact interfaces were numbered sequentially from 1.

Fixed the `THERMAL` option of `*CONTACT` in MPP when some MPP partitions do not participate in all the contact definitions.

Fixed `*CONTACT_2D_AUTOMATIC_TIED` when used with selective mass scaling, which is made active by setting `IMSL=1` and `DT2MS<0` on `*CONTROL_CONTACT`. The problem occurred when selective mass scaling was applied to the nodes that are tied. Prior to the change, kinematic constraints were failing.

Fixed MPP `*CONTACT_2D_AUTOMATIC` when beams were in contact, or when solids were used with nonzero values of solid surface offset(s) (variables `SLDSOS` or `SLDSOM`). Both cases could lead to using invalid memory space and a segmentation fault.

Add new option `ICNEP=1` to `*DEFINE_FRICTION` with which those lines with non-existent parts or part sets will be ignored.

Variable `ENGOUT` on `*CONTROL_OUTPUT` added to write minimum (wherever occurring) contact energy densities to `d3plot` (MORTAR contact only).

Variable `NENG` on `*DATABASE_EXTENT_INTFOR` added to write contact energy densities to `intfor` (MORTAR contact only).

Variable `PENOUT` on `*CONTROL_OUTPUT` added to write maximum (wherever occurring) penetrations across interfaces to `d3plot` (MORTAR contact only).

Variable NPEN on *DATABASE_EXTENT_INTFOR to write absolute and relative penetrations to intfor (MORTAR contact only).

Variable NTWELD on *DATABASE_EXTENT_INTFOR writes user tied weld history variables to intfor file (MORTAR contact only).

Variable VC on *CONTACT_... supported for MORTAR contact.

Added variable TDPEN on

*CONTACT_2D_AUTOMATIC_SURFACE_TO_SURFACE_MORTAR. This is the analogue to interference option MPAR1 for IGNORE=3 in 3D automatic MORTAR contact.

Mortar contact supports erosion, that is, any solid segment or shell edge segment belonging to parts involved in the contact definition will become active in the contact when exposed to the environment due to erosion.

IGNORE=4 on *CONTACT_..._MORTAR supports a curve of relative interference reduction vs. time. When $MPAR1 < -1.0$, $|MPAR1|$ is the curve ID.

*LOAD_THERMAL_CONSTANT is supported in mortar contact, in those situations where the contact properties depend on temperature.

In eigenvalue analysis, nonzero mortar contact stress will not influence rigid body modes, i.e., you should see the proper rigid body modes among the available output.

TEMP<0 on *CONTACT_..._MORTAR_TIED_WELD calls a user tied weld interface, allowing implementation of "arbitrary" tied condition in weld simulations.

*CONTACT_TIED_..._THERMAL supported for (spotweld) beams.

The THERMAL option can now be used for the following tied contacts:

*CONTACT_TIED_NODES_TO_SURFACE_OFFSET_THERMAL

*CONTACT_TIED_NODES_TO_SURFACE_CONSTRAINED_OFFSET_THERMAL

*CONTACT_TIED_NODES_TO_SURFACE_THERMAL

*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_OFFSET_THERMAL

*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET_THERMAL

*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_CONSTRAINED_OFFSET_THERMAL

*CONTACT_TIED_SHELL_EDGE_TO_SURFACE_THERMAL BUT the slave set type can NOT be a node set, at least not yet.

Now implemented for MPP are

*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK and

*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIED_WELD.

Segment-based contact (SOFT=2) now supports groupable and non-blocking features.

Option FTORQ for transmittal of moments across the contact interface is implemented in the groupable node-to-surface contacts, surface-to-surface contacts, and tied contacts.

*CONTACT_AUTOMATIC_GENERAL supports options SOFT=1 and SRNDE for shell edge treatment.

*CONTACT_AUTOMATIC_GENERAL supports options SOFT=1 and SRNDE for the shell edge treatment.

A new orthogonal friction model is developed for the constraint and penalty

*RIGIDWALL_ORTHO. A RIGIDWALL_ORTHO with equal friction coefficients in all directions now behaves as a regular RIGIDWALL with a single friction coefficient.

*CONTACT_RIGID_TO_RIGID is enhanced to support unloading curve, FCM=1/2/3.

*CONTACT_CONSTRAINT_NODES_TO_SURFACE is enhanced to support variable PENSF on MPP Card 2 of *CONTACT and recover contact forces that existed before solid remeshing associated with 3D solid adaptivity.

*CONTACT_AUTOMATIC_GENERAL is improved to stabilize short beams with small rotational inertias.

ERODING contacts are now supported in full deck restarts.

*CONTACT_AUTOMATIC_BEAMS_TO_SURFACE supports beam to shell_edge contact and detects the contact between shell edges and phantom nodes.

A new built-in variable IDRFLG added to *DEFINE_CURVE_FUNCTION,

*DEFINE_FUNCTION. IDRFLG carries a value of 1 during dynamic relaxation and 0 in the transient phase.

Multiple *DEFINE_FRICTION tables with the same ID are merged.

Effect of variable I2D3D is corrected in MPP to avoid duplicated contact segments when the attached shell and solid are not assigned to the same processor.

ERODING contact in MPP is fixed to compute the stiffness of newly exposed solid nodes and segments and apply the optional thickness SLDTHK to the newly exposed solid segments.

The contact thickness in *PART_CONTACT is applied to

*CONTACT_AUTOMATIC_GENERAL.

*CONTACT_AUTOMATIC_GENERAL is corrected by reducing the contact stiffness to half.

1.4.10 *CONSTRAINED

Fix ordering issue during initialization of *CONSTRAINED_INTERPOLATION in MPP that could have resulted in incorrect "colinear" warnings or possibly deadlock.

Fix MPP message passing error that could occur if a node involved in

*CONSTRAINED_SHELL_TO_SOLID is shared between more than 2 processors.

Enhanced processing of an exactly singular constraint matrix for
*CONSTRAINED_INTERPOLATION.

Corrected error in the checking for massless *CONSTRAINED_INTERPOLATION
constraints in single precision.

Enhance *CONSTRAINED_INTERPOLATION in the explicit solver so the linear algebra of
the constraint equation processing always operates in double precision to reduce loss of
precision in single precision executables.

Add implicit support to *CONSTRAINED_SHELL_IN_SOLID.

*CONSTRAINED_BEAM_IN_SOLID:

- Add support for IDIR=1. This means the beam is allowed to freely slip longitudinally inside
solid elements.

- Add implicit support for the AXFOR option. Debonding between beams and solids may now
be simulated using the implicit solver.

- Add implicit support for CONSTRAINED_BEAM_IN_SOLID_PENALTY.

Fix incorrect *CONSTRAINED_INTERPOLATION motion of the dependent nodes if the
number of constrained dof is less than the total dof, i.e., DDOF.ne.123456.

Fix ineffective PNODE in *CONSTRAINED_NODAL_RIGID_BODY when PNODE<0.

Fix kinetic energy dependence on MPP core count when using
*CONSTRAINED_JOINT_(OPTION) with LMF=1 in *CONTROL_RIGID.

Added swforc file output for welds modelled by *CONSTRAINED_GENERALIZED_WELD.
Output values are the brittle failure forces, axial and shear, and the failure function.

Improved temperature-dependent failure curve option of spot welds defined by
*CONSTRAINWED_SPOTWELD. Curve interpolation was using the wrong segment of the
curve. Also, changed swforc output to show zero axial and shear forces after a weld has failed.

Add correct torsion to SPR3 (*CONSTRAINED_INTERPOLATION_SPOTWELD with
STIFF4>0).

Small modification for PIDVB<0 of *CONSTRAINED_SPR2/SPR3: Beams get deleted after
failure (they were just separated before).

Fix for *CONSTRAINED_SPR2/SPR3: part id of internally generated beams has to take Nodal
Rigid Bodies into account.

Ctl

Add error trap if INTPERR=1 is set on *CONTROL_SHELL and data interpolation would be
done with *INITIAL_STRESS_SHELL.

*CONSTRAINED_JOINT_STIFFNESS_CYLINDRICAL now available to model cylindrical/revolute connections with play

*CONSTRAINED_COORDINATE now supports part set ID in addition to part ID.

*CONSTRAINED_COORDINATE: When IDIR is negative, the constraint is applied at the nearest node, not at the coordinate.

For nodes which are not attached to any elements, turn off the automatic global constraint normally applied to such nodes if those nodes are involved in

*CONSTRAINED_MULTIPLE_GLOBAL.

1.4.11 *CONTROL

*CONTROL_REFINE_... with NTOTRF=-1: Refine solids or shells with elements dynamically added during the run (as opposed to NTOTRF>0 for which the user must estimate the number of child elements required for the refinement and these elements are added during the initialization).

*CONTROL_REFINE_SHELL: Support contact for the case of NTOTRF=-1, i.e., dynamically add contact segments due to mesh refinement.

In *CONTROL_REFINE_... keywords:

- If CRITRF<0, reverse the refinement conditions
- If CRITRM<0, reverse the coarsening conditions

*CONTROL_STAGED_CONSTRUCTION:

- Added support for *ELEMENT_SOLID ELFORMs 4, 13, 16, 17.

- Added new input field IDYNAIN to suppress output of dynain file at end of each stage. This can be done for all stages using IDYNAIN on *CONTROL_STAGED_CONSTRUCTION, or for individual stages using IDYNAIN on *DEFINE_CONSTRUCTION_STAGES.

- Fix bug affecting *MAT_021 and *MAT_021 solids with Staged Construction (*DEFINE_STAGED_CONSTRUCTION_PART). When a part became active, the stresses were wrongly initialized. Bug affected solid elements only.

*CONTROL_PORE_FLUID:

Fixed bug in internal energy calculation for solid element types with more than one integration point. This affected only models containing *CONTROL_PORE_FLUID.

*CONTROL_ADAPTIVE: Small change to 2d adaptive remeshing that should improve behaviour for meshes with bad boundaries.

Fix problem with 2d adaptivity and boundary merging. Some boundary points between materials weren't merged in some cases, depending on where the program thought the 2d contours started. This only applies for *CONTROL_ADAPTIVE adpopt=8 with mmm2d=1.

Implement idrflg=-3 for *CONTROL_DYNAMIC_RELAXATION which invokes the version R7 (and earlier) implementation of idrflg=3. Now idrflg=3 means the parts not included in the part set are eliminated from the computation, and idrflg=-3 means all parts are included in the explicit computation but only those in the part set are included in the distortional-kinetic-energy-based convergence check in the dynamic relaxation phase.

Ignore *CONTROL_SPOTWELD between parts that are inactive during implicit dynamic relaxation phase.

Fix a bug whereby SHLTRW in *CONTROL_SHELL was ignored when ISTUPD>0.

Add variable MLKBAG to *CONTROL_OUTPUT to include accumulated airbag gas leakage in abstat.

Fix wrong pointer used for section id/properties when generating part for visualization of rigidwall during adaptivity. This cause error message KEY+137.

*CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS:

- Add a stiffness option (ISTFNS) to *CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS.

EQ.1: no stiffness is added.

EQ.2: only spin softening is added which keeps stiffness symmetric.

EQ.3: both spin softening and gyroscopic effects are added which makes the matrix skew symmetric. Default is 3.

- Add the whirling direction to the output eigout .

DISPLAY option in *RIGIDWALL_PLANAR will be ignored for stationary rigid walls when SKIPRWG=1 in *CONTROL_CONTACT.

Added options to the ERODE parameter on *CONTROL_TIMESTEP. The original options were 0 and 1 where 1 causes solid and thick shell elements to be eroded if their time step dropped below TSMIN*DT0 where TSMIN is specified on *CONTROL_TERMINATION, and DT0 is the solution time step that is calculated in the first cycle. The new options are to check and delete shell elements and beam elements having small time steps. The control of shells is in the 'tens' place and the beams is in the 'hundreds' place. The valid options for ERODE are now 1, 10, 11, 100, 101, 110, and 111. For ERODE=111, beams, shells, and solids are all checked. For ERODE=11, shells and solids are checked. The original options of 0 and 1 still behave exactly as they did.

Added support of *CONTROL_SUBCYCLE for segment-based (SOFT=2) contact. Contact forces are calculated for all nodes of a segment pair only if any node of the pair is due to have a force calculated.

Fixed some issues so that solid elements and thick shell elements will work as advertised with respect to DTMIN on *CONTROL_TERMINATION, ERODE on *CONTROL_TIMESTEP, and PSFAIL on *CONTROL_SOLID. These variables control whether we terminate the analysis or fail elements when their time step becomes small or their volume becomes negative.

Add an option TET13V on *CONTROL_SOLID to choose between the efficient and a more accurate version of the tet type 13 implementation.

Lagrangian multiplier joints, LMF=1 on *CONTROL_RIGID, are supported for consistent mass scaling (RBSMS=1 on *CONTROL_RIGID).

***CONTROL_ADAPTIVE:**

Fix bug in 3D tet adaptivity when there are rigid parts with a tetrahedral mesh and ELFORM=10. Before the bug fix, the rigid parts with ELFORM=10 were not properly written out during adaptive restart, resulting in an error termination.

***CONTROL_REMESHING:**

3D tet adaptivity is able to handle the remeshing of multiple bodies in a single adaptive part. Preserving of feature lines is controlled by the variable SEGANG.

1.4.12 Discrete Element Method

New keyword *DEFINE_DE_FLOW_DRAG to support DES interacting with external flow field.

Add a moving system with birth/death times in *DEFINE_DE_ACTIVE_REGION.

Fix bug of generation failure and uneven distribution of injected DES particles (*DEFINE_DE_INJECTION) when the radius of injected DES particles is very small.

Fix segmentation fault when using simple and small restarts in DEM.

Add the keyword *DEFINE_DE_COHESIVE to enable the capillary force in specified DES node set or parts.

Add the keyword *INITIAL_STRESS_DES to initialize stress in discrete element spheres (DES).

Output the DES initial stress and geometry information into dynain_ini and dynain_geo files (FSPLIT=1 in *INTEFACE_SPRINGBACK_LS_DYNA).

Fix incorrect behaviour when using *DEFINE_DE_ACTIVE_REGION with itype=1, i.e. BOX, with *DEFINE_TRANSFORMATION. The box was not rotated.

1.4.13 EFG (Element Free Galerkin)

***SECTION_SOLID_EFG:**

Volume of 6-noded solid elements using element formulation 41 (EFG) was calculated incorrectly. That bug is now fixed.

1.4.14 *ELEMENT

For triangular shells with thickness stretch (*SECTION_SHELL, ELFORM=27) the option IDOF=1 is made available allowing a continuous thickness field between them and quadrilateral shells with thickness stretch, ELFORM=25,26.

For the Belytschko-Tsay shell with thickness stretch (*SECTION_SHELL, ELFORM=25) an improved representation of stresses over the thickness may be obtained with the new option IDOF=11 instead of IDOF=1 (for continuous thickness fields) or with the new option IDOF=12 instead of IDOF=2 (for discontinuous thickness fields).

Changed *DEFINE_ELEMENT_EROSION to be more compatible with the original FE options and added support for IGA elements via *DEFINE_ELEMENT_EROSION_IGA.

Added calculation of implicit tangent for shell formulation 46, a 2D cohesive element.

*ELEMENT_DIRECT_MATRIX_INPUT:

- Enhanced the reading of superelement files to support connections using nodes, scalar nodes, internal dofs, and the label of "0" for scalar nodes.
- Added MPP logic for collecting nodes associated with superelements for use in decomposition for the reading of binary formatted files for superelements.
- Fix an error associated with using superelements in MPP explicit. Two arrays should have been initialized to zero but the bug did not manifest itself until the user used 12 processes. The arrays are now properly initialized to zero.
- Corrected the initialization of superelements involving nodes with SPC constraints.

*ELEMENT_SEATBELT_PRETENSIONER: make type-8 pretensioner available for 2D belt.

*ELEMENT_SEATBELT_SENSOR: add type-5 sensor for tracing retractor payout.

Restructure nodal volume data communication for solid tet formulation 13 for better parallel performance.

Added new more conservative time step calculation (independent of shell thickness) for improved stability in cohesive shell formulation 29.

Fix incorrect stress output to d3plot and ASCII files when using tetrahedron solid types 10 and 13 with orthotropic materials and when CMPFLG=1 in *DATABASE_EXTENT_BINARY.

Fixed a bug that caused axisymmetric elements (shell ELFORMs 14,15) to instantly fail when DTMIN>0 in *MAT_ADD_EROSION.

Shell elements with *MAT_NULL can be deleted by TIME in *DEFINE_ELEMENT_DEATH_SHELL.

Added a check for invalid ELFORM values on *SECTION cards. Invalid values were leading to terminations without clear error termination messages.

Added an optional modification to the time step calculation that is done for solid elements. See variable IHDO in *CONTROL_TIMESTEP. Setting IHDO to 1 causes the time step calculation to be modified such that it does not have a discontinuity between expansion and compression. With the default time step calculation, the solution time step can jump up or down about 6% when an element controlling the time step switches from compression to expansion.

The size of the jump depends on the linear bulk viscosity coefficient. These jumps may not occur if multiple elements are controlling the time step as it is unlikely for all to be expanding at the same time, so the smaller time step is used consistently. Setting IHDO to 1 prevents this discontinuity from occurring.

Added support for shell form 23 in output file eloutdet (*DATABASE_ELOUT). When shell form 23 is used, the code can now output stress and strain at 2x2 or 3x3 integration points and extrapolate to all 8 nodes.

Added a bulk viscosity option for thick shell (TSHELL) elements (See variable TSTYPE on *CONTROL_BULK_VISCOSITY).

Enabled plane strain and axisymmetric elements (*SECTION_SHELL ELFORM 13, 14, 15) to include stress initialization by *INITIAL_STRESS_SHELL for materials that use equations of state.

Enabled solid spot weld assemblies to use 8 node and 6 node cohesive elements with *MAT_240. There is no assembly failure calculation, but the resultant forces and moments are calculated and can be output to the swforc file. This works with solid element formulations 19, 20, 21, and 22.

Enabled variables CDL and TDL on *SECTION_DISCRETE to be used when one of the discrete beam nodes is constrained by an SPC.

Improved the accuracy of pressure loading on fully integrated, volume-weighted axisymmetric solid elements (ELFORM=15/NIP=4 on *CONTROL_SHELL), particularly along the axis of symmetry.

Improved the post-failure behaviour of integrated beam elements by adding an option (NAUDP on *SECTION_BEAM) to update the neutral axis when one or more integration points fail so that nodal moments are correctly calculated. This option is available for the integrated beam formulations 1, 9, 11, and 14 when used with *MAT_003, *MAT_098, *MAT_100, *MAT_124, and *MAT_158.

Improved mass scaling of beam element formulations 4 and 5. A dimensional problem was causing erratic behaviour which could cause unstable behaviour.

Fixed bug affecting simultaneous use of ERODING contact with cohesive pentahedral solids, which through use of ESORT=1, are changed to formulations 21/22. Prior to this fix, this combination of keywords could have led to error termination due to negative volume.

Add check if two or more parts use solid element formulation 13 (tetrahedron with pressure averaging) and those parts share some nodes. Write a warning in that case because such a condition could lead to instabilities.

COHOFF on *SECTION_SOLID allows for adjusting the reference layer for cohesive element formulations 20 and 22.

Fixed bug in plane strain, 8-node shell formulation 55 to show correct stresses when 4 in-plane points are required for d3plot output. Restored writing out stress components at all integration

points in elout. Shell formulation 55 is for implicit analysis only and requires 4, 9 or 16 integration points (default 4) since it is meant to simulate the singular stress field around a crack tip.

For thick shell (TSHELL) formulations 1 and 2, fixed incorrect stress output in elout when used for composite materials and CMPFLG=1.

1.4.15 EM (Electromagnetic Solver)

*EM_SOLVER_BEM: Option PRECON=4 (line 1 field 4), LLT factorization of the BEM matrix, can now be used in MPP. It could only be used in serial/SMP up to now.

Eddy current/inductive heating solver:

- Addition of monolithic solver for FEM+BEM solve (higher time steps and improved stability for ferromagnetic materials) (*EM_SOLVER_FEMBEM).
- Optimization of memory/CPU cost in BEM solve.
- Improved accuracy of solution on prismatic elements.
- Addition of LLT factorization as a preconditioner for BEM solver in MPP (*EM_SOLVER_BEM).
- Added 2D and 2D axi capability for RSW and other applications (*EM_CONTROL and *EM_MAT_004).
- Support of eroding conductors (*EM_CONTROL_EROSION and *EM_MAT_...).
- Improved accuracy of thermal coupling for tetrahedral elements.

EM contact:

- Optimize contact search and robustness => reduced calculation times and memory cost
- Added capability in RSW to specify electric contact law (*EM_CONTACT_RESISTANCE).

Battery module Release:

- Equivalent circuit model (ECM): Implementation of Randles circuits as simplified electro-chemistry model
- Dynamic EM-mechanical-thermal coupling for external and internal short circuit (battery crush) (see in particular *EM_RANDLES_SHORT).
- Bath loading, coupling mono-bi and bath in same domain.
- Purkinje network (*EM_EP_MECACOUPLING).

- Current defined stimuli (*EM_EP_TENTUSSCHER_STIMULUS).

1.4.16 *FATIGUE

Added using *DEFINE_TABLE to define SN curves for multiple mean stress.

Added new mean stress correction method (METHOD=5): Morrow equation (SN curve).

Extended thick shell element fatigue analysis (including multiaxial) to MPP.

Extended solid element fatigue analysis (including multiaxial) to MPP.

Extended shell element fatigue analysis (including multiaxial) to MPP.

Fixed a bug in running *FATIGUE_SUMMATION. This function is now working as a standalone module. Previously it was used inside fatigue analysis.

Added warning message MSG_FTG+5 in case the calculated stress value is higher than the stress of the first point on SN curve, to remind user to check the SN curve or stress calculation. If the stress value is higher than the stress of the first point on SN curve, extrapolation is used and this may not be accurate, especially if the first point is the UTS (Ultimate Tensile Strength).

Added a parameter (DMGMIN in *FATIGUE) to allow user to define base damage ratio for elements with zero stress or strain.

Added a new option _D3PLOT to allow running time domain fatigue analysis based on d3plot.

Implemented *FATIGUE_FAILURE to remove failed elements from model.

Implemented *FATIGUE_MULTIAXIAL to run multiaxial fatigue analysis.

Implemented *FATIGUE_LOADSTEP to run fatigue analysis with multiple load steps.

Added warning message MSG_FTG+4 if fatigue properties are not defined for the current element under fatigue analysis.

Implemented definition of initial fatigue damage ratio by part or part set (PID/SID in *INITIAL_FATIGUE_DAMAGE_RATIO).

Added semi-log SN curve definition. This can be used both in time domain and frequency domain fatigue analysis.

Added exposure time TEXPOS for fatigue analysis, in case it is different from endtim in *CONTROL_TERMINATION.

Added an option D3PLOT to *INITIAL_FATIGUE_DAMAGE_RATIO to allow using damage variables from transient preload cases as initial fatigue damage ratio.

Fixed a bug in defining mean stress correction methods. The UTS/yield strength should be defined on material model, not part ID, according to the Manual.

1.4.17 Forming Analysis

*DEFINE_BOX_NODES_ADAPTIVE now supports more than 2 levels of mesh refinement for tube adaptivity. Maximum level is specified by variable LEVEL.

ONESTEP simulations with triangular elements used to produce misleading effective plastic strain results in regions with high curvatures. This is now fixed.

Fix bug in ONESTEP method that produces false minor and major strain in the output file onestepresult

Fix bug in *CONTROL_FORMING_ONESTEP, OPTION=6.

Fix MPP bug for ONESTEP output files repositioned.k and onestepresult.

Fixed the problem of lacking temperatures at the lancing line in MPP lancing simulations.

Fixed a memory access problem which resulted in error termination of simulations of unflagging process.

*CONTROL_FORMING_MAXID is supported by 3d tet adaptivity (*CONTROL_ADAPTIVE).

*INCLUDE_STAMP_PART: Improvement to support *DEFINE_TRANSFORMATION.

*DEFINE_CURVE_TRIM_3D: Projection of curve to blank is now allowed when the trimming curve is far away from the blank. This was previously disallowed.

*CONTROL_FORMING_ONESTEP:

- Add a CPU time report for onestep method to record the time spent on initial unfolding.
- Allow user to output specific part to the file of 'onestepresult'.
- The fracture curve LCSDG can be automatically found from *MAT_ADD_EROSION.

*CONTROL_FORMING_PRE_BENDING(_LOCAL): New option allows user to define a vector based on a local coordinate system. If the LOCAL option is used, then coordinate system ID is input on Card 2.

*ELEMENT_LANCING: Allow multi-curve lancing with different starting times. Previously, only one starting time was allowed. With this new feature, all the lancing curves will be used to cut at the same time, but the nodes will not be able to separate until the lancing time is reached.

*CONTROL_FORMING_AUTOPOSITION_PARAMETER:

- Support 2D elements.
- When CID is a negative value, its absolute value refers to a vector ID (*DEFINE_VECTOR) rather than a coordinate system ID.

***DEFINE_FORMING_ONESTEP_MASTER:** This new keyword allows a second blank to be welded to a master blank. The two blanks are connected by using spot welds.

Add keyword ***CONTROL_FORMING_TRIM_SOLID_REFINEMENT** to homogeneously refine elements along a trim curve. Supports ***DEFINE_CURVE_TRIM_2D** and ***DEFINE_CURVE_TRIM_3D**.

***DEFINE_CURVE_TRIM_2D** allows refinement of a sandwiched part along trimming curves. The sandwich core comprised of multiple layers of solid elements can now be refined. In addition, the trimming curve can be in any direction (2D trimming only).

***CONTROL_FORMING_ONESTEP_ORTHO** uses 2 nodes in the final part to define the rolling direction of the anisotropic material.

***DEFINE_FIBERS:**

- User can perform carbon fiber simulation without the matrix (fibers only).
- No need to define material property for the matrix.

1.4.18 ***FREQUENCY_DOMAIN**

Correction to rotational inertia for ***FREQUENCY_DOMAIN_SSD**.

***con**

Extend logic for implicit constraint handling to tied contact with **_CONSTRAINED_OFFSET** to skip any SPC constraints on the rotational dofs of the SPC slave nodes. We already have such logic to skip the constraints on the translational dofs for the slave nodes for all tied contacts.

***loa**

Correct the issue where use of ***LOAD_BODY** is applied to a model with rigid bodies. If the vector of nodes having the load applied belong to a rigid body then the vector of elemental stiffness matrices is null. So logic was added to skip the call to the implicit matrix assembly subroutine for that case.

***DATABASE_FREQUENCY_BINARY_{OPTION}:**

- Added a new option **SUMMATION** to ***DATABASE_FREQUENCY_BINARY_D3PSD** and ***DATABASE_FREQUENCY_BINARY_D3RMS** to get total PSD and RMS response from multiple loading resources.

- Update to support ***INCLUDE_TRANSFORM**.
- Added **NFSPACE=3** to define output frequency as eigenfrequencies.

- Added an option (**BINARY=2**) to dump out individual mode response to **d3spcm** (***FREQUENCY_DOMAIN_RESPONSE_SPECTRUM**).

***DATABASE_FREQUENCY_ASCII_{OPTION}:**

- Update to support *INCLUDE_TRANSFORM.
- Added NFSPACE=3 to define output frequency as eigenfrequencies.

*FREQUENCY_DOMAIN_ACOUSTIC_BEM:

- Update to support *INCLUDE_TRANSFORM.
- Added radiated power computation to Rayleigh method.

*FREQUENCY_DOMAIN_ACOUSTIC_FEM:

- Update to support *INCLUDE_TRANSFORM.
- Update to use mixed elements (tet, hex and pentahedrons).
- Fixed a bug in running with combined boundary conditions (transient vibration and impedance).
- Added a flag (RESTRT) to run restart from transient analysis results. LS-DYNA will read an existing binary database to extract the velocity boundary condition.
- Added defining impedance boundary condition by *DEFINE_CURVE.
- Added a flag "idump" in *FREQUENCY_DOMAIN_ACOUSTIC_FEM_EIGENVALUE to dump out acoustic stiffness and mass matrices.

*FREQUENCY_DOMAIN_FRF:

- Update to support *INCLUDE_TRANSFORM.
- Enabled using structural damping (*DAMPING_STRUCTURAL).

*FREQUENCY_DOMAIN_LOCAL:

- Implemented this keyword to run frequency domain analysis on part of model.
- Implemented options SOLID_SET, SHELL_SET, BEAM_SET and TSHELL_SET.
- Implemented option EXCLUDE to exclude specified sets of elements or nodes.
- Enabled using this feature on response spectrum analysis (*FREQUENCY_DOMAIN_RESPONSE_SPECTRUM).

*FREQUENCY_DOMAIN_MODE:

- Implemented option LOAD_PROJECTION, to select the normal modes based on load projection ratio.
- Implemented option MODAL_COEFFICIENT to select the participating modes based on relative significance of the modal coefficients.

- Enabled this keyword to FRF (*FREQUENCY_DOMAIN_FRF) and SSD (*FREQUENCY_DOMAIN_SSD).

*FREQUENCY_DOMAIN_RANDOM_VIBRATION:

- Update to support *INCLUDE_TRANSFORM.

- Added parameters ICOARSE and TCOARSE in *FREQUENCY_DOMAIN_RANDOM_VIBRATION to reduce points in PSD curve to save CPU time.

- Added parameter TOL_INTG to allow user to define the relative tolerance for integration. The default is 1.e-5.

- Updated PSD curve input so that loading types 9 (base velocity), 10 (base displacement), 11 (enforced acceleration), 12 (enforced velocity), and 13 (enforced displacement) can be defined through time history curve.

- Added structural damping (*DAMPING_STRUCTURAL) to random vibration analysis.

*FREQUENCY_DOMAIN_RESPONSE_SPECTRUM:

- Enabled running response spectrum analysis with CASE.

- Added beam elements stress/strain components in elout_spcm when applicable. Previously only resultant forces and moments are provided.

- Fixed a bug in reading d3eigv family files for multipoint response spectrum analysis.

- Update to support *INCLUDE_TRANSFORM.

- Fixed a bug in running response spectrum with base excitation defined in time domain.

*FREQUENCY_DOMAIN_SSD:

- Fixed a bug in reading d3eigv for restart option (RESTMD=1) when both 4-noded shells and 8-noded shells (like ELFORM 23) are present in the model.

- Update on SSD to use the current geometry of model if SSD is performed after dynamic analysis. Previously SSD is always performed based on the initial geometry.

- Implemented radiated acoustic power computation with Rayleigh integral.

- Added radiation efficiency computation for ERP (*FREQUENCY_DOMAIN_SSD_ERP).

- Enabled sense switch for direct SSD (ctrl-c).

- Updated *FREQUENCY_DOMAIN_SSD_FATIGUE by adding scale factor for each load. Added MSG_KEY+1232, MSG_KEY+1233 to remind user about the changes.

- Added torque and rotational dof excitation for SSD (VAD=8, 9, 10, 11), with local damping (DMPFLG=1).
- Update to support *INCLUDE_TRANSFORM.
- Extended *FREQUENCY_DOMAIN_MODE_LOAD_PROJECTION to the computation of nodfor_ssd, nodfor_ssd and elout_ssd.
- Added option _FREQUENCY_DEPENDENT to run direct SSD with frequency dependent material properties.
- Added option _FRF to run SSD with FRF setting. Changed the output filename to d3frf. With this option, one can get fringe plot for FRF (previously only xyplot results were given by FRF).
- Added rotational dof output in nodout_ssd, when applicable.
- Added new loading options vad=12, 13, 14 for direct SSD (*FREQUENCY_DOMAIN_SSD_DIRECT), for prescribed velocity, acceleration, and displacement.

1.4.19 ICFD (Incompressible Fluid Solver)

Navier Stokes solve/free surface flows:

-
- *ICFD_BOUNDARY_FSWAVE: Added Stokes wave of fifth order, solitary wave model and irregular waves using JONSWAP spectrum. Add a complete check of wave generation settings and the validity of the parameters within each wave theory.
*ICFD_BOUNDARY_PRESCRIBED_VEL can be used jointly to model the interaction of any wave theory and any prescribed velocity at inlets.
 - *ICFD_BOUNDARY_PERIODIC: Addition of periodic, reflective and sliding mesh boundary conditions.
 - *ICFD_CONTROL_TIME: Maximum and minimum time step definitions can now be made functions of time.
 - *ICFD_CONTROL_OUTPUT_SUBDOM: Supports output to d3plot (before only supported VTK and VTU formats).
 - *ICFD_DEFINE_SOURCE: Allows the user to specify a generic volumetric force entering the velocity solve. Can be useful in various applications, for example, when coupling ICFD with the EM solver.
 - *ICFD_INITIAL: *DEFINE_FUNCTION can now be used to define an initial velocity profile.
 - *ICFD_INITIAL_LEVELSET: Extended the number of initial surfaces which can be generated; boxes and spheres are now possible on top of surface plane.

- *ICFD_MAT: Variable CA specifies the contact angle for the liquid-vapor interface, for use in surface tension problems.
- *ICFD_MODEL_POROUS: Added new porous models, pmid=11 and pmid=10, for parachute fluid flow. To be used with thin shells (See *MESH_EMBEDSHELL).
- Bug Fix: Removed pressure outlet smoothing technique after noticing it could cause disruption of results. Results of internal flow problems might be especially affected.
- Bug fix: Added more digits to icfd_fluidmesh.key output to avoid round off errors.

Turbulence models:

- *ICFD_CONTROL_TURBULENCE, SUBMOD=2: Added LES Dynamic turbulence model originally proposed by DK Lilly (1991) with localization on coefficient CS by Piomelli and Liu (1995).
- *ICFD_DEFINE_TURBSOURCE: new keyword which allows addition of turbulent source terms for RANS models.
- *ICFD_CONTROL_TURB_SYNTHESIS: now supports *DEFINE_FUNCTIONS.
- Bug fix: Fixed incorrect calculation of stresses in laws of the wall.

FSI, DEM coupling and remeshing:

- *ICFD_CONTROL_ADAPT: Variable KIS may be used to diminish the excessive coarsening that can potentially occur when remeshing by keeping the nodal mesh size distribution of the initial volume mesh.
- *ICFD_CONTROL_ADAPT: Variable VAR gives the user more control on the error calculation triggering a remesh.
- *ICFD_CONTROL_DEM_COUPLING: Variable MAXVEL is a 'max velocity' term for the DEM force calculation intended to avoid spurious high velocities which could cause the fluid solver to diverge.
- *ICFD_CONTROL_DEM_COUPLING: Variable DTYPE=1 invokes option to use Morrison formula for Drag calculation to apply on DEM particles.
- *ICFD_CONTROL_FSI: Added control on the number of FSI fluid subiterations. This avoids the sometimes unneeded excessive number of FSI subiterations when the fluid and very light structures (like parachutes) develop a resonance-like mode inside the FSI subiterations (coupling iterations). See variable NSUB.

Thermal and Conjugate heat transfer (CHT):

- *ICFD_BOUNDARY_FLUX_TEMP: Modified boundary condition so that the prescribed value is equivalent to quantity prescribed in BOUNDARY_HEAT_FLUX.

- *ICFD_BOUNDARY_CONJ_HEAT: Added option to impose a temperature drop between solid and fluid (see variables VAL and SFLCID). Also added an alternative penalty based (mortar) coupling method (CTYPE=1). This can be beneficial in cases with large mesh size discrepancies but is typically associated with higher solve times.
- *ICFD_BOUNDARY_CONVECTION_TEMP: Added keyword to prescribe a convection boundary condition for thermal problems.
- *ICFD_CONTROL_CONJ: TSF feature added in ICFD solver for thermal and CHT problems.
- *ICFD_CONTROL_TURBULENCE: Added thermal law of the wall; see variable TWLAW.
- Bug fix: Previously, all FSI surfaces were taken into account for the thermal and fluid coupling. Now only the boundaries to which the keyword *ICFD_BOUNDARY_CONJ_HEAT is applied will be coupled to the solid.
- Bug fix: HCLCID and TCLCID in *ICFD_MAT were incorrectly read in MPP.
- Bug fix: icfd_thermal.dat now keeps getting output in CHT cases after the Navier-Stokes equations have reached steady state.

1.4.20 Implicit (Mechanical) Solver

New LSOLVR options in *CONTROL_IMPLICIT_SOLVER:

- 17 (SMP only): iterative solver with a preconditioner based on a local incomplete factorization.
- 27 (MPP only): ditto.

*CONTROL_IMPLICIT_ORDERING: Made various improvements to LS-GPart (ORDER=4).

- Significant improvements in MPP;
- Added hybrid parallelism (OpenMP);
- Added a progress report ("heartbeat") for large problems.

New option where ILIMIT in *CONTROL_IMPLICIT_SOLUTION can vary with time. If $ILIMIT < 0$, $|ILIMIT|$ is taken as the curve ID which defines the iteration limit between automatic stiffness reformations (ordinate) vs. time (abscissa).

Implemented an approximate constitutive matrix for implicit analysis of *MAT_187/*MAT_SAMP-1.

New implicit feature for automatic generation of solution key points through examination of load curves for points of particular interest, e.g., times at which maximum load occurs. Set $DTMIN < 0$ in *CONTROL_IMPLICIT_AUTO to invoke this option.

Enhanced the performance of the output to d3* files for *CONTROL_IMPLICIT_EIGENVALUE and *CONTROL_IMPLICIT_MODES. This saves 15% of the wall clock time for large eigenvalue problems where hundreds of eigenmodes are written to d3eigv.

Corrected the use of load curve specifications for the following implicit control variables:

- IAUTO<0 in *CONTROL_IMPLICIT_AUTOMATIC_DYN,_SPR
- IMASS<0 in *CONTROL_IMPLICIT_DYNAMICS_DYN,_SPR
- ISTIF<0 in *CONTROL_IMPLICIT_SOLUTION_DYN,_SPR

*CONTROL_IMPLICIT_INERTIA_RELIEF:

- Corrected the processing of some kinds of contact during the implicit inertia relief computation.
- Disabled use of iterative linear equation solvers in *CONTROL_IMPLICIT_SOLVER when *CONSTRAINED_IMPLICIT_INERTIA_RELIEF is active.
- Needed to account of inertia relief modes in MPP for the computation of resulting forces from constraints.
- Enhance explicit's use of *CONTROL_IMPLICIT_INERTIA_RELIEF by adding a second orthogonalization step for the accelerations.

*CONTROL_IMPLICIT_MODES:

- Fixed application of implicit constraint modes in MPP for the case where a constraint mode is also a shared node.
- Added a switch on *CONTROL_IMPLICIT_MODES to not write d3mode file when generating superelements.
- Improved the creation of superelement files in MPP.
- Correct dynamic memory allocation issue that was causing problems with computing the reduced stiffness matrix in MPP for implicit modes.
- Corrected output to d3mode in MPP.
- Added error checking on nodes for *CONTROL_IMPLICIT_MODES to make sure they are deformable.

*CONTROL_IMPLICIT_MODAL_DYNAMICS:

- Allow the use of shell formulation 18 in modal dynamics when reading from d3eigv.
- Correct a typo in the computation of the explicit time step for modal superposition. Also applies to *CONTROL_IMPLICIT_DYNAMICS with imass = 2 or 3.
- Adjusted reading of d3eigv for modal dynamics for the case where the modes are computed and then immediately used -- all with stresses included.
- Correct the implicit modal dynamics stress output for beams using non-default value for BEAMIP.

***CONTROL_IMPLICIT_EIGENVALUE:**

- Freed allocated memory that was not free at the end of subroutine `im_mode_stress`.
- Adjusted the logic for intermittent eigenvalue computation to deal with explicit not quite getting to the end time.

- Properly handle the output of 10 noded tets to the d3eigv database.

- Reduce size of d3eigv by not adding rotational dofs for solid element only models.

- The computation of the i/o address for writing the d3eigv database in MPP for models with 8 noded shell elements was wrong. The d3eigv database would probably fill the i/o system causing an abnormal termination. The i/o address computation is now correct.

- Enable the computation of Campbell diagrams for Implicit Rotational Dynamics as part of the Intermittent Eigenvalue capabilities.

- Enable the use of intermittent eigenvalues during both dynamic relaxation and regular transient phase.

- For eigenvalue analyses where there are lots and lots of zero eigenmodes where the eigensolver fails we will now output the computed modes to the d3eigv database for model debugging purposes.

- Extend the cost logic for Lanczos Eigensolver, both SMP and MPP, to be more aggressive about staying with the current shift. Improves performance for large problems as the cost function is not necessarily monotonic.

- Added shift and scaling to the standard eigenvalue problem to make the solution methodology more robust. This is the same shift and scaling as used for MCMS.

- Corrected output to d3eigv for implicit MPP eigensolver for large number of integration points.

- Add logic so that for some frequency response computations the amount of drilling rotation control is the same as for eigenvalue computations.

- Added Sectoral Symmetry to *CONTROL_IMPLICIT_EIGENVALUE as an implicit eigensolver option for SMP DP. Sectoral Symmetry is an eigensolver for models with a large degree of rotational symmetry such as fan blades.

- Adjust zero shift for Lanczos Eigensolver.

- Fix up MPP storage issues in MPP implementation of Lanczos. Increased minimum storage of eigenvalues from 500 to 2500. Restricted eigenvector checking due to MPI buffer length.

- Add additional tag at end of d3eigv for MCMS that was missing. Normalize MCMS computed eigenvectors to have unit norm.

Correct joint constraint processing in implicit mechanics which should improve the reporting of joint forces (*DATABASE_JNTFORC) in implicit.

Enhance implicit's processing of nodal mass matrices to include the local coordinate transformation.

*CONTROL_IMPLICIT_SOLUTION with NSOLVR = 1:

Fixed problem in which Implicit Linear Analysis used the forces at the end of the linear step to compute resultant forces. This did not match expectations of the users. Computation of resultant forces, such as those in bndout, were changed use the force at the beginning of the linear step.

*CONTROL_IMPLICIT_SOLVER:

- Correct the removal of ordering reuse logic for the MPP implementation of implicit mechanics. This is an important time saving feature for very large models with multiple time steps.
- For intermittent eigenvalue analysis the direct solvers used by the eigenvalue analysis were forced for the entire execution. We now save and restore the solver option selected by the user to use during the non-eigenvalue execution of the run. This enables the use of iterative solvers iterative solvers.
- Fix the implicit logic to properly deal with NEGEV=1 in *CONTROL_IMPLICIT_SOLVER and IAUTO.ne.0 in *CONTROL_IMPLICIT_AUTO.
- Disabled the use of ORDER=1 in *CONTROL_IMPLICIT_SOLVER for MPP as this is a really bad choice in MPP.
- Adjusted output to d3hsp for implicit linear equation solver options to match keyword manual. (*CONTROL_IMPLICIT_SOLVER)
- Adjust implicit logic for dumping matrices (MTXDMP in *CONTROL_IMPLICIT_SOLVER) Correct the dumping of matrices from implicit when MTXDMP > 1.

*CONTROL_IMPLICIT_STATIC_CONDENSATION: Correct indexing of array involved in implicit static condensation.

Echo the input for *CONTROL_IMPLICIT_RESIDUAL_VECTOR to d3hsp.

Fix an issue of the implicit linear algebra filenames being inconsistent Linux and Windows.

Add an implicit-only option for processing *CONSTRAINED_INTERPOLATION which is referred to as force-geometry-mass (fgm). Forces are moved from the dependent node to the independent nodes, geometry of the dependent node is computed from the independent nodes, and the mass on the dependent node is moved to the independent nodes. All without using a constraint matrix or LaGrange Multiplier approach.

No longer automatically increase nodal DOF from 3 to 6 due to the presence of *CONSTRAINED_INTERPOLATION. This is unnecessary for solid element models and increases the cost of large solid element models for implicit by 15% due to huge increase in constraint processing to constrain out all of the rotational dofs.

Correct implicit's handling of SPC constraints on shared nodes in MPP that are also involved in tied contact.

Enhance Implicit constraint processing to accept function definitions for motor joint movement (*CONSTRAINED_JOINT_***_MOTOR).

*CONTROL_IMPLICIT_SOLUTION: Added NSOLVR=-1 so that the initial geometry is used for each step of a multistep linear analysis.

Added implicit time stamps and additional performance tracking for MPP Lanczos eigensolver. Extended command Line Option impcon=1.

Added a command line directive of impcon=1 which tracks events during implicit simulation to analyse performance for large implicit jobs. This is also a useful debugging tool as it helps identify the particular part of implicit is being executed at an abnormal termination.

Single precision executables are now disallowed for implicit mechanics (both MPP and SMP).

Restored final memory report for implicit SMP runs. Logic was based on old memory allocation scheme instead of new dynamic memory allocation.

Turned on Mumps, a new linear solver option for implicit mechanics, to be included for double precision SMP on XEON64 and AMD64 systems.

Corrected an inefficiency for implicit when there are many many AutoSPC constraints in MPP. A linear search through the AutoSPC constraints was replaced with a binary search.

Corrected the collection of resultant forces when using implicit mechanics for tied contact using _OFFSET. Other tied contact such as _CONSTRAINED_OFFSET were OK.

Added SMP implementation of new keyword *CONTROL_IMPLICIT_RESIDUAL_VECTORS.

Apply logic that was in R9 and R10 but had not made it into R11 and R12 to check for incorrect values in nodal inertia array for implicit dynamics and eigenvalue analysis.

Fixed divide by zero in power iteration for buckling problems with inertia relief. (*CONTROL_IMPLICIT_BUCKLING, *CONTROL_IMPLICIT_INERTIA_RELIEF)

Correct problem where solution control was inappropriately returned to implicit after issuing sense switch sw1.

Added *CONTROL_IMPLICIT_RESIDUAL_VECTOR.

Purpose: Activate and control the computations of residual vectors. Residual vectors are the linear response of a model to a specified load which is then orthogonalized to a set of

eigenmodes and any previously computed residual vectors. The eigenmodes can either be computed during the execution or read from an input file. The computation of residual vectors is the same as multi-step linear (NSOLVR = -1 on *CONTROL_IMPLICIT_SOLUTION) but has the additional step of orthogonalization.

Add a stiffness option to *CONTROL_IMPLICIT_ROTATIONAL_DYNAMICS. The stiffness terms are suppressed to keep the stiffness matrix symmetric but the force terms are added.

*MAT_ELASTIC_FLUID/*MAT_001_FLUID usage with
*CONTROL_IMPLICIT_EIGENVALUE:

Warning added to highlight the absence of enforcement of irrotationality in solid elements that use this fluid material model. In eigenvalue analyses this absence will often lead to the prediction of numerous spurious rotational fluid modes. This material model should therefore be used with caution in modal analyses.

Fixed implicit element stiffness of shell elements when used with laminated shell theory, and when also using anisotropic materials *MAT_022, *MAT_034, *MAT_040, *MAT_054, *MAT_055, or *MAT_058. In certain cases, the shear stiffness was too small.

Lagrangian multiplier joints, LMF=1 on *CONTROL_RIGID, are supported for for implicit analysis.

Fixed bug with d3eigv when using TET10S8=1 on *CONTROL_OUTPUT.

1.4.21 *INITIAL

*INITIAL_LAG_MAPPING:

- Thick shell data from a previous run can be mapped to a new thick shell mesh.
- *INITIAL_LAG_MAPPING with NELANGL=-3: For the Lagrangian mapping, project the boundary nodes of the current mesh onto the boundary faces of the previous mesh.
- In a solid-to-solid mapping, map the strain array output by STRFLG=1.
- Map the nodal pressures for ELFORM=13 tetrahedra.
- If NELANGL=-2 in a 3D to 3D mapping, the data from the previous run are rotated around the axis of an angle ANGLE before mapping.
- *INITIAL_LAG_MAPPING_WRITE3DAXI: For a 3D axisymmetric Lagrangian model, add the option WRITE3DAXI to *INITIAL_LAG_MAPPING to output the mapping file as if it came from a 2D axisymmetric model.

Fix bug in the *INITIAL_STRESS_TSHELL module of the dynain output file. For Thick Shells with 6-10 integration points through the thickness, the z-coordinates of the integration points were incorrect.

Added new option ISTIFF to *INITIAL_STRESS_SECTION for enhanced stability.

Added new ISTIFF option for artificial stiffness in *INITIAL_STRESS_SECTION.

Fix *INITIAL_STRESS_TSHELL output to dynain for thick shell form 7. Output stresses for all 4 in-plane integration points instead of just 1.

Fix incorrect rotational axis when using *PART_INERTIA and *INITIAL_VELOCITY_GENERATION with IRIGID=1 and the rotational axis is defined using nodes by setting NX=-999 and defining the nodes with NY and NZ.

Make *MAT_GENERAL_NONLINEAR_1DOF_DISCRETE_BEAM/*MAT_121 work with *INITIAL_STRESS_BEAM.

Disable *INITIAL_AXIAL_FORCE_BEAM and *INITIAL_STRESS_SECTION when IDRFLG=-999 in *CONTROL_DYNAMIC_RELAXATION.

Fix seg fault or incorrect stresses when initializing stresses using *INITIAL_STRESS_SOLID with *MAT_107/*MAT_MODIFIED_JOHNSON_COOK.

Fixed a bug that made *INITIAL_VELOCITY_GENERATION_START_TIME not work for rigid parts.

STRFLG in *DATABASE_EXTENT_BINARY will be automatically set to unity when using *INITIAL_STRAIN_SOLID.

Fixed the writing of dynain files that use beam element formulations 7 or 8. They were writing the wrong number of integration points.

Fixed a problem that occurred when initialising stress in thick shells using a dynain file created by a previous run that used a different number of integration points per layer than the new run. In this case, when reading the dynain data to initialize the element, the stress tensor was being properly averaged or extrapolated as needed, but the material extra history variables were not, which for some materials caused wrong material behaviour.

Fixed the reading of initial strain data for thick shell formulations 2, 3, 5, 6, and 7. A strain transformation was missing.

Add new keyword *INITIAL_HISTORY_NODE(_SET) to initialize select history variables at node locations. These nodal values are internally interpolated to the element integration points using the finite element shape functions.

A main difference to using *INITIAL_STRESS_SHELL/SOLID, where you would need to initialise ALL history variables up the ones you really need/want to initialize is that *INITIAL_HISTORY_NODE(_SET) allows you to just pick the few variables you would like to initialise without touching the others.

*INITIAL_AXIAL_FORCE_BEAM now supported for beam formulation 1 (H-L beam) and any material. Previously, only beam formulation 9 with *MAT_SPOTWELD was allowed.

Selective mass scaling is not applicable to beam formulation 1 (Hughes-Liu beam).

*CONTROL_ADAPTIVE: Users are able to use *INITIAL_VELOCITY_GENERATION for the 3D adaptive parts as long as STYP=3 (node set) and the corresponding node set is defined using *SET_NODE_GENERAL with OPTION=PART. (This would apply only to the unusual situation in which initial velocities were invoked AFTER the mesh is adapted.)

1.4.22 Isogeometric Analysis (IGA)

*ELEMENT_SHELL_NURBS_PATCH:

- Include initial parameterization check for isogeometric shells.
- Include automatic removal of trimming loops fully contained within a single element.

Updated *DEFINE_SPOTWELD_RUPTURE to work with isogeometric shell elements.

Added IGA shell and IGA solid element erosion due to the material failure strain, *MAT_ADD_DAMAGE, and *MAT_ADD_EROSION.

Contact for IGA in implicit is now available using the automatically generated interpolation elements. The IGACTC option in *CONTROL_CONTACT, however, is not currently supported for implicit.

Added *INITIAL_VELOCITY_GENERATION support for IGA elements using parts and part sets.

Added support for checking element stiffness matrices for IGA through the LPRINT option in *CONTROL_IMPLICIT_SOLVER.

Both IGA solids and IGA shells now work with *MAT_RIGID. Previously, only IGA shells were supported for rigid material.

IGA now works for implicit springback.

Added the ISTUPD thickness change options (*CONTROL_SHELL) to IFORM=3 IGA shell elements.

Added drilling stiffness type 4 (see DRCM in *CONTROL_IMPLICIT; DRCPSID and DRCPRM in *CONTROL_SHELL) to the IGA shells.

Improved the time centering for IGA shells for improved convergence rates in implicit analysis.

*ELEMENT_SHELL_NURBS_PATCH: Enabled keyword *LOAD_NURBS_SHELL for trimmed elements.

*ELEMENT_SOLID_NURBS_PATCH:

- IINT defines the integration rule.

EQ.0: reduced Gauss integration rule (default)

EQ.1: full Gauss integration rule.

EQ.2: patch-wise reduced integration rule.

EQ.3: non-uniform Gauss integration rule uses full integration for exterior elements and reduced integration for interior elements.

- Add conventional mass-scaling for isogeometric solids

These material models are now supported for shells in isogeometric analysis:

- *MAT_224/*MAT_TABULATED_JOHNSON_COOK
- *MAT_054-055/*MAT_ENHANCED_COMPOSITE_DAMAGE

Laminated shell theory is now supported for shells in isogeometric analysis. This option can be activated for shell formulations that allow for transverse shear deformability by:

- Setting LAMSHT=1,3 or 5 in *CONTROL_SHELL, and
- Using *INTEGRATION_SHELL to specify an integration rule, which is referenced in *SECTION_SHELL.

*INITIAL_STRAIN_SHELL_NURBS_PATCH: Fixed output for thickness in case ISTUPD=0 in *CONTROL_SHELL.

*INITIAL_(STRESS/STRAIN)_SHELL_NURBS_PATCH: Fixed error in writing of *DEFINE_NURBS_CURVE to dynain file (*INTERFACE_SPRINGBACK_LSDYNA).

Fixed bugs that arose from improper merging of interpolation nodes at patch boundaries. This could have led to errors such as "Zero Normal Vector".

Remove the limitation to have NURBS patches and related trimming curves defined in a normalized parameter space as this is not common in CAD-files.

Fix bug when using IGA-shells (*ELEMENT_SHELL_NURBS_PATCH) together with user defined elements (ELFORM=100-105).

1.4.23 *LOAD

There were unexpected error terminations when *LOAD_THERMAL_LOAD_CURVE was used in an adaptive analysis. This has been fixed.

*LOAD_THERMAL_D3PLOT reads temperature data from a file in d3plot format. The file name is specified as an option on the command line. If this file does not exist in the working directory, LS-DYNA now gives a reasonable message and exits gracefully.

*LOAD_BODY_GENERALIZED: fix a bug that happens when CID >0 and the CID doesn't have its origin at (0,0,0).

Fix incorrect results when using *DEFINE_CURVE_FUNCTION with AX/AY/AZ form *LOAD_SEGMENT.

Fix output to nodfor when using *LOAD_SSA.

Fix ineffective *LOAD_THERMAL_CONSTANT when used with *MAT_ELASTIC_VISCOPLASTIC_THERMAL/*MAT_106.

Fix seg fault when using *LOAD_MOVING_PRESSURE due to uninitialized variable.

Fixed a bug preventing *LOAD_BLAST_ENHANCED from working for 2D analysis.

Added line-of-sight feature (LSFLG) to *LOAD_MOVING_PRESSURE: When LSFLG.EQ.1, the pressure will be applied to the first-hit segments from the nozzle.

When IDIR=3 in *LOAD_MOVING_PRESSURE, the pressure is in the direction from NODE1 to NODE2, but only the normal component of the pressure is applied on the segments.

Enabled *LOAD_THERMAL to work for beam nodes when nodal releases are defined at those nodes. The thermal loading was previously incorrect.

New keyword *LOAD_THERMAL_RSW: Prescribes nodal temperatures within a (possibly moving) ellipsoidal region of the structure in mechanical-only simulations. So there is no heat transfer available to the surroundings. Input is comparable to thermal boundary condition *BOUNDARY_TEMPERATURE_RSW.

Features include:

- Temperatures for the center and the boundary of the ellipsoid have to be input. In-between, there is a quadratic approximation.
- Outside of the ellipsoid, no temperature values are prescribed.
- Position and axis of symmetry are defined by two nodes
- Optional definition of a heat affected zone (HAZ) around the weld nugget. Temperatures at its boundary are to be given and a linear interpolation from the boundary of the nugget to the boundary of the HAZ is used. Outside the HAZ a default temperature is applied

Implement *LOAD_SPCFORC in MPP.

Reduce memory usage and computation time for *LOAD_SEISMIC_SSI_NODE when ground motions are specified at every node on the soil-structure interface.

New keyword *LOAD_EXPANSION_PRESSURE applies a uniform pressure (e.g., from explosive detonation) on the internal surfaces of a chamber while one wall of the chamber moves (as a piston) so that the loaded area increases.

1.4.24 *MAT and *EOS

Added *MAT_295/*MAT_ANISOTROPIC_HYPERELASTIC.

- This is a collection of (nearly-in)compressible, (an)isotropic, hyperelastic material models primarily aimed at describing the mechanical behaviour of biological soft tissues. Some of the material models may also be used to analyse a wider class of materials including fiber-reinforced elastomers and stretchable fabrics.
- Modules include isotropic, anisotropic, and active anisotropic.
- Allows for electrophysiology-fluid-structure interaction (EFSI).
- Available for solid element formulations 1, 2, and 10.

*MAT_079/*MAT_HYSTERETIC_SOIL: New option for cyclic degradation. See input fields SIGTH, SIGR and CHI on Card 5.

*MAT_119/*MAT_GENERAL_NONLINEAR_6DOF_DISCRETE_BEAM: New failure option FCRIT (Field 7, Card 6, see also Remark 1). The pre-existing failure algorithm uses two separate criteria: one that considers only negative displacements and rotations of all six degrees of freedom, and a second criterion considering only positive displacements and rotations. Thus,

positive x-displacement and positive y-displacement could combine to cause failure while positive x-displacement and negative y-displacement could not. The new option FCRT=1 offers a single criterion that considers both positive and negative displacements/rotations.

*MAT_169/*MAT_ARUP_ADHESIVE bugfix: The "interfacial failure" option (TMAXE, SMAXE) was not propagating from element to element as it should.

Fixed bugs in *MAT_172/*MAT_CONCRETE_EC shear capacity calculation (TYPESC > 0):

- This capability did not work with fully integrated shell elements such as ELFORM=16.
- Incorrect results or error terminations could occur if material types other than 172 were mixed with MAT_172 in PART_COMPOSITE and the MAT_172 definition had TYPESC > 0.

*MAT_197/*MAT_SEISMIC_ISOLATOR: New option for sliding isolator with rim failure (TYPE=5).

*MAT_203/*MAT_HYSTERETIC_REINFORCEMENT: Fixed bug in MAT_203 affecting beam elements with EPDAM1, EPDAM2 non-zero, and DRESID equal to zero. When the beam reached its erosion limit, error termination or unexpected erosion of neighbouring elements could occur.

*MAT_209/*MAT_HYSTERETIC_BEAM: New material model for modelling buildings in earthquakes. A significantly extended version of *MAT_191 (*MAT_SEISMIC_BEAM) with lumped plastic hinges at both ends of the beam and many additional capabilities.

Add heat generation factor QCURE to

*MAT_ADHESIVE_CURING_VISCOELASTIC/*MAT_277 which relates the heat generated in one-time step with the increment of the degree of cure in that step.

*MAT_296/*MAT_ANAND_VISCOPLASTICITY:

- Correct the deformation resistance S calculation.
- Fix floating point exception related bug when compiling with FPE option active.

Add keyword *MAT_ADD_CHEM_SHRINKAGE.

- The inputs and features are same as those of *MAT_ADD_THERMAL_EXPANSION except for the usage of curve LCID.
- The physical chemical shrinkage rate is calculated as $\beta = (1.0 - \alpha)^3 - 1.0$ where curve LCID defines alpha (ordinate) vs. temperature (abscissa).
- When defining curve LCID in *DEFINE_CURVE, the variable DATTYP should be set to -100.

Extension of THERML variable in *MAT_077_H to allow C11, C20, C02, and C30 to also be specified as a function of temperature.

Add calculation of internal energy to *EOS_019/*EOS_MURNAGHAN.

Make *EOS_019/*EOS_MURNAGHAN compatible with *INCLUDE_TRANSFORM.

Add support for optional EOS in

*MAT_COMPOSITE_FAILURE_SPH_MODEL/*MAT_059.

Variable PHEL in *MAT_JOHNSON_HOLMQUIST_CERAMICS/*MAT_110 can be specified equal to 0.0 by default or by user input. In this case PHEL is calculated internally. This value is now echoed as "phel" in d3hsp.

*MAT_ADD_PERMEABILITY: allow the permeability to be a function of volumetric strain, plastic strain rate, or pressure.

*MAT_ADD_PORE_AIR: fix an MPP bug triggered when input format is long=s, write standard keyword input and write long structured input deck.

*MAT_FABRIC: make strain restoration factor TSRFAC available for linear liner.

*MAT_GENERAL_NONLINEAR_6DOF_DISCRETE_BEAM/*MAT_119:

- Allow unloading option of "2" for crushable beam, IFLAG=2.
- Add the stiffness matrix to enable implicit for crushable beam.

*MAT_SEATBELT:

- Enhance 2D belt by offering choice of element formulations and material properties along transverse direction for 2D belt.
- Enable 2D belt material to have loading curve, LLCID, defined as a table.
- Issue error message when the edge nodal set of 2D belt part is not on the edge of the related 2D belt part

Added isotropic plasticity to *MAT_030/*MAT_SHAPE_MEMORY.

Added new material model *MAT_291/*MAT_SHAPE_MEMORY_ALLOY for shape memory alloys. This is a micromechanics-inspired constitutive model for shape-memory alloys that accounts for initiation and saturation of phase. This model is based on Kelly, Stebner, Bhattacharya (2016) and is available for solid elements only.

Added support for curves defining thermal properties vs. mechanical history variables in *MAT_THERMAL_ISOTROPIC_TD_LC.

Fixed the problem in *MAT_244/*MAT_UHS_STEEL where redundant table look-up operations in the plasticity algorithm increased the computational cost.

Added history variable (at the 8th position) in *MAT_133/*MAT_BARLET_YLD2000 to allow user input of an effective plastic strain shift which can be used to account for stored energy in sub-structures after heat treatment. This option is only applicable to the case of $HARD < 0$ (curve/table input of hardening curves).

Added 3D table option to hardening curve input of *MAT_233/*MAT_CAZACU_BARLAT so that the material model is applicable to simulations of hot/warm forming of materials such as magnesium.

Fix incorrect initial strains when using *MAT_MOONEY-RIVLIN_RUBBER/*MAT_027 with *INITIAL_FOAM_REFERENCE_GEOMETRY and running in single precision.

Fix problem of solution hanging when using *MAT_PIECEWISE_LINEAR_PLASTICITY_STOCHASTIC and *DEFINE_HAZ_PROPERTIES (MPP only).

Fix incorrect behaviour for *MAT_GENERAL_JOINT_DISCRETE_BEAM/*MAT_097 when BEGTIM in *CONTROL_START is set >0.0.

Fix incorrect stresses for *MAT_ANISOTROPIC_VISCOPLASTIC/*MAT_103 and *MAT_ANISOTROPIC_PLASTIC/*MAT_103P when using BETA for *ELEMENT_SHELL_BETA or BETA in the *MAT input or material integration point angles input on *SECTION_SHELL.

Fix seg fault when using *MAT_PIECEWISE_LINEAR_PLASTICITY/*MAT_024 for Hughes-Liu beam with implicit analysis.

Output fitted data to curveplot for *MAT_HYPERELASTIC_RUBBER/MAT_77_H when N>1.

Fix effect of FBRT on tensile strength, XT, for *MAT_ENHANCED_COMPOSITE_DAMAGE/*MAT_055. FBRT should only affect the tensile strength upon compressive matrix mode failure.

If FAILTM < 0.0 in *MAT_ADD_EROSION, skip erosion failure evaluation during dynamic relaxation phase.

The calculation of Q12 for *MAT_SPECIAL_ORTHOTROPIC/MAT_130 and *MAT_RESULTANT_ANISOTROPIC/MAT_170 is incorrect. It should be $Q12=(v21p*E11p)/(1-v12p*v21p)$.

Fix ineffective curves LCSRA, LCSRB, LCSRC, LCSRAB, LCSRBC, LCSCA in *MAT_MODIFIED_HONEYCOMB/*MAT_126.

Error terminate with message, INI+484, if LOG_INTERPOLATION is invoked for *MAT_PIECEWISE_LINEAR_PLASTICITY/*MAT_024 and the strain rate for the table is defined by LCSS is negative.

Fix ineffective user-define failure subroutine (FAIL<0) for *MAT_MODIFIED_PIECEWISE_LINEAR_PLASTICITY/*MAT_123.

Fix output to elout for solids using *MAT_PAPER/*MAT_274 and CMPFLG>0 in *DATABASE_EXTENT_BINARY.

Allow *MAT_065/*MAT_MODIFIED_ZERILLI_ARMSTRONG to compute the flow stress using a more general form by using power exponent, m, of the effective plastic strain instead of the square root of the effective plastic strain in the case of an FCC metal.

Fix seg fault when using *MAT_ADD_EROSION for *PART_COMPOSITE when the first layer of the *PART_COMPOSITE is not using *MAT_ADD_EROSION.

Fix incorrect direction of thermal expansion when using *MAT_NONLINEAR_ORTHOTROPIC/
*MAT_040 with *MAT_ADD_THERMAL_EXPANSION.

Fix contact penetration after element erosion when using *MAT_110/
*MAT_JOHNSON_HOLMQUIST_CERAMICS with eroding contact.

Fix incorrect stress/strain output to d3plot when using *MAT_023/
*MAT_TEMPERATURE_DEPENDENT_ORTHOTROPIC and running 2D analysis with
MAXINT>3 in *DATABASE_EXTENT_BINARY.

Fix incorrect stress initialization when using *LOAD_DENSITY_DEPTH in combination with
*MAT_005/*MAT_SOIL_AND_FOAM where the material's volumetric strain vs. pressure is
input using curve LCID.

Fix strain rate effects on *MAT_157/*MAT_ANISOTROPIC_ELASTIC_PLASTIC for
implicit static analysis.

Fix seg fault when using *MAT_278/*MAT_CF_MICROMECHANICS.

Fixed a bug causing 2WAY=1 in *MAT_ENHANCED_COMPOSITE_DAMAGE/*MAT_054
not to be recognized.

Added rate sensitivity to *MAT_S04/*MAT_SPRING_NONLINEAR_ELASTIC: LCD can
point to a table (*DEFINE_TABLE) which gives loading rates (relative velocity values). There
is then defined a corresponding curve of force vs. displacement for each loading rate.

Added PRESTRAIN keyword option for
*MAT_MODIFIED_PIECEWISE_LINEAR_PLASTICITY/*MAT_123. When PRESTRAIN
is used, the variable IPS on Card 5 is used to control the prestrain option.
If IPS is set to 1, then the strain tensor defined by *INITIAL_STRAIN_SHELL will be used as
prestrain when checking major strain failure, EPSMAJ.

Added a new option called TTOPT to *MAT_SPOTWELD which is used by solid and solid
assembly spot welds. The option controls the behaviour of TRUE_T making it possible to
revert the TRUE_T behaviour back to how it worked in R8 and previously. If TTOPT=0 or if
the TTOPT is not input, then the behaviour is unchanged with this update. If TTOPT=1, then
R8 and earlier behaviour happens. TTOPT=2 is like TTOPT=1, but the average top and bottom
values of forces and moments are used for failure making failure invariant with respect to
element numbering.

Added support for user-defined spot weld failure in *MAT_100_DAMAGE-FAILURE with
individual solid elements. Previously, only beam elements and solid weld assemblies were
supported.

Enabled *MAT_ADD_THERMAL_EXPANSION to work in thick shell elements when the
solution is mechanical only. Some code was missing causing bad behaviour.

Enabled *MAT_128/*MAT_HEART_TISSUE for thick shell forms 3, 5, and 7. Initialization of the F tensor was missing causing a 1st cycle termination.

Enabled *MAT_145, *MAT_244, and *MAT_254 to work with thick shell formulations 3, 5, and 7.

Enabled multiple instances of nonlocal material averaging to be used in MPP. This change affects *MAT_NONLOCAL, and also *MAT_CODAM2, *MAT_GURSON_RCDC, or *MAT_PLASTICITY_WITH_DAMAGE with the RCDC option.

Enabled the variable EC of *MAT_058/*MAT_LAMINATED_COMPOSITE_FABRIC to be used with thick shell formulations 1, 2 and 6. If a positive value is input, then it will be used. If EC is left blank or set to zero or a negative number, then the minimum of EA and EB will continue to be used. Also, *MAT_058 will now echo to d3hsp EC=0 when used with thin shells because the EC value is ignored.

Improved hourglass form 6 when use with *MAT_089/*MAT_PLASTICITY_POLYMER. A poor choice of the material parameter used to scale the hourglass stiffness made it difficult to use.

Fixed spot weld assemblies that use user defined failure option 12 in *MAT_100. An error was causing OPT=12 to not work at all.

Fixed the use of *MAT_ADD_DAMAGE_DIEM with *MAT_024 and tetrahedral element formulation 13. The damage calculation used bad data leading to a wrong result.

Fixed the initialization of the material direction for orthotropic materials when used with AOPT=2 in 2D plane strain, plane stress, and axisymmetric elements.

Fixed implicit beam spot weld elements (*SECTION_BEAM ELFORM=9) when used with NF>0 on *MAT_100/*MAT_SPOTWELD. The averaging of the stress was causing the model to converge to the wrong solution with beam forces too low.

*MAT_296/*MAT_ANAND_VISCOPLASTICITY: Solved a problem that caused discrepancies between material parameters provided in the input keyword and the values reported in the d3hsp file. Analysis results were not impacted.

*MAT_097/*MAT_GENERAL_JOINT_DISCRETE_BEAM: Solved a problem that may have caused analysis termination due to fatal error when a simulation switched from implicit to explicit.

*MAT_261/*MAT_LAMINATED_FRACTURE_DAIMLER_PINHO: Solved a problem that may have prevented accumulation of back stress for in-plane shear plasticity model.

*MAT_LAMINATED_FRACTURE_DAIMLER_CAMANHO/*MAT_262:

- Fixed bug in in-plane shear behaviour for tshells if curves are used to define strain rate dependency of SIGY and ETAN.

- Corrected warning messages in case of tshell formulations 1/2 and adjusted the computation of element length for fracture toughness to match with thin shells.

- Added flag MSG in *MAT_262 to control the writing of warning messages.

- Added transverse shear damage similar to *MAT_054/*MAT_058.

- Added shell-only option to use nonlinear (elastic) stress-strain curves instead of constant moduli (EA, EB, GAB). This allows asymmetric tension-compression.

- Added flag DSF that controls the failure of an integration point based on in-plane shear failure. This was added to be able to reinstate the in-plane shear failure behaviour prior to revision 117876 (DSF=1).

- Fixed incorrect behaviour of *MAT_262 when first layer in *PART_COMPOSITE used a non-262 material model (shells only).

- Fixed bug when using table (2D) for defining strain rate dependent fracture toughness's values with respect to characteristic element length.

*MAT_LAMINATED_COMPOSITE_FABRIC/*MAT_058:

- Fixed a error in calculation of the initial time step when curves define the stiffness values for EA, EB, GAB (shells only).

- Add the SOLID option to support *MAT_058 for solid elements.

- Fixed bug when using curve LCEFS to define ERODS failure strain as a function of strain rate (shells only).

- Add option to define direction-dependent failure strains. For this, define a curve LCDFAIL that has 5 (for shells) or 9 (for solids) data points and has LCINT set to the number of data points. The ordinate values define the failure strains in the different directions as follows: 1: ef_11t ; 2: ef_11c ; 3: ef_22t ; 4: ef_22c ; 5: ef_12 ; (and additionally for solids) 6: ef_33t ; 7: ef_33c ; 8: ef_23 ; 9: ef_31

The curve's abscissa values must be ascending to meet input requirements but are otherwise ignored.

*MAT_ANISOTROPIC_ELASTIC_PLASTIC/*MAT_157: Output the history variable that tracks failure (according to the Tsai-Wu or Tsai-Hill criterion) as history variable #10.

*MAT_ENHANCED_COMPOSITE_DAMAGE/*MAT_054: Correct the computation of transverse shear strains for solids and tshells when EPSR and EPSF are defined.

*MAT_VISCOPLASTIC_MIXED_HARDENING/*MAT_225: Fix issues when a table to account for strain-rate dependency is used together with kinematic hardening (BETA<1).

Add support of IORTHO=1 for cohesive user-define materials.

*MAT_106/*MAT_ELASTIC_VISCOPLASTIC_THERMAL: Added possibility to specify up to 8 user-defined history variables for output. The values of these history variables are computed via *DEFINE_FUNCTION. These values can, for example, be used to evaluate the hardness of the material.

*MAT_254/*MAT_GENERALIZED_PHASE_CHANGE:

-
- New transformation laws (PTLAW=5,6,7,8,9) which are tailored for titanium (Ti-6Al-4V).
 - 5: Recovery of alpha martensite based on JMAK, using Koistinen-Marburger equation to estimate equilibrium concentration. Transformation of recovered phase partially to to-phase/
 - 6: Associated with 5; transformation of remaining recovered phase to this to-phase.
 - 7: Parabolic growth rate calculating the stepwise dissolution of two or more phases into one other phase. This law calculates the sum of remaining phase concentrations. The from-phase is the first to be dissolved.
 - 8: Associated with 7; additional phases to be dissolved according to 7. Phases are dissolved one after the other.
 - 9: Extensions of Koistinen-Marburger; threshold of fr-phase and incomplete transformation within a certain range of temperature rates.
 - 12: JMAK transformation law (same as 2), but active within the temperature range from start and end temperature, i.e. it is active in heating and cooling.

- Read PTTAB6 for JMAK transformation law to account for accelerated phase transformation due to plastic strain in the material.

- Account for TSF (*CONTROL_THERMAL_SOLVER) in phase kinetics.

- History data only written for phases that are actually defined.

- Added variable POSTV which allows user to specify additional history variables (accumulated thermal strain, accumulated strain tensor, plastic strain tensor, equivalent strain) for output. When the temperature is in the annealing range, these values are reset.

- New input variable for initial plastic strain to initialize a constant nonzero equivalent plastic strain value in the whole part.

- Added enhanced annealing option that controls the reset of plastic strains above a certain critical temperature. The anneal process can be described by a time and temperature dependent approach (JMAK, tabular input, load curve inputs).

- Add a cut-off temperature for the thermal expansion.

*MAT_270/*MAT_CWM:

-
- New option ANOPT (annealing option) that allows cutting the thermal expansion above a temperature limit (e.g. annealing temperature).
 - Added variable POSTV which allows user to specify additional history variables (accumulated thermal strain, accumulated strain tensor, plastic strain tensor, equivalent strain) for output. When the temperature is in the annealing range, these values are reset.

*MAT_277/*MAT_ADHESIVE_CURING_VISCOELASTIC:

- Implemented Arrhenius shift function as alternative to the WLF shift function.

- Heat generation due to curing can be accounted for.

*MAT_278/*MAT_CF_MICROMECHANICS:

- Added curing induced heating.

- Fixed the solid formulation to give comparable results as the shell formulations.

- For the solid formulation, added input for the elastic properties of the fiber contribution in the direction that is orthogonal to the fiber plane.

Fix for combination of *MAT_ADD_PORE_AIR and *MAT_ADD_EROSION with solid element formulations 1, 10, 43, 60, 61, and 99. History variables could have been mixed up.

Increase accuracy for rate curves SIGVM<0 and MXEPS<0 of *MAT_ADD_EROSION. The original input curve is used instead of the rediscrretised one.

Add new variable P4 for *MAT_ADD_DAMAGE_DIEM to define if transverse shear stresses are included in the stress invariant computations (e.g. triaxiality) or not. Useful for shells with corresponding "plane stress" material models such as *MAT_024_2D, *MAT_036,

Add new option MIDFAIL = 4 to GISSMO (*MAT_ADD_DAMAGE_GISSMO) and "eGISSMO" (*MAT_ADD_GENERALIZED_DAMAGE).

Add new option LP2BI to *MAT_ADD_DAMAGE_GISSMO (shells only): failure becomes a function of triaxiality and a new bending indicator (adopted from *MAT_258).

Fixes for GISSMO damage when used together with an equation-of-state (*EOS).

Take IRATE flag from *CONTROL_IMPLICIT_DYNAMICS into account for GISSMO.

Add new keyword *MAT_ADD_SOC_EXPANSION for geometric expansion due to State Of Charge from EM solver (currently only available for isotropic, hypoelastic materials with solid element formulations -2, -1, 1, 2, and 10).

Add option to *MAT_024 with VP=3, where yield stress can now be a function of plastic strain, strain rate, and up to five history variables that can be set using *INITIAL_HISTORY_NODE. That means LCSS refers to *DEFINE_TABLE_XD up to a level of 7.

Add *MAT_024's option VP=3 (filtered total strain rate) for solid elements.

Enhancement when using log interpolation of strain rate data with VP=1 in shell elements in *MAT_024, *MAT_123, or *MAT_251: eliminate numerical noise for very low strain rates by internally adding lower limits for strain rate and yield stress.

Add effective strain as history variable #28 for *MAT_034/*MAT_FABRIC with FORM = 14, 24, or -14.

Fix problem with internal computation of E0 for *MAT_036/*MAT_3
PARAMETER_BARLAT if hardening rule HR=2 or HR=5 is used AND E0 is zero in the input
AND E is less than zero (curve for Young's modulus).

Add new options to *MAT_068/*MAT_NONLINEAR_PLASTIC_DISCRETE_BEAM:
translational and rotational stiffnesses TK{R,S,T} and RK{R,S,T} can now be assigned
negative values referring to curve IDs for load/moment vs. displacement/twist to get nonlinear
elastic behaviour.

Fix for *MAT_105/*MAT_DAMAGE_2 with large curve IDs for LCSS (only solids).

Make *MAT_133/*MAT_BARLAT_YLD2000 available for solid elements (currently only for
explicit analysis and without C-R kinematic hardening).

Fix for *MAT_169/*MAT_ARUP_ADHESIVE with variables referring to
*DEFINE_FUNCTIONS: function ID offsets from *INCLUDE_TRANSFORM (IDOFF) did
not work.

*MAT_187/*MAT_SAMP-1: instability criterion now available as history variable #28.

Modifications for *MAT_187L/*MAT_SAMP_LIGHT:
- log interpolation for table (optional) and no extrapolation
- improved tolerance for plane stress iteration procedure
- add viscoelasticity (defined by LCEMOD and BETA)
- add flag to control curve treatment, either using
discretized (0, default) or original (1).

Fix thermal expansion in *MAT_188/*MAT_THERMO_ELASTO_VISCOPLASTIC_CREEP
for solid elements: double precision version could show no expansion at all.

Enable *MAT_199/*MAT_BARLAT_YLD2004 to be used with shell element formulations
25,26,27 and tshell element formulations 3,5,7. Set default values (1.0) for the transformation
matrix coefficients.

Add option to *MAT_224/*MAT_TABULATED_JOHNSON_COOK: variable BETA<0 can
now refer to a *DEFINE_TABLE_4D for triaxiality (TABLE_4D), temperature (TABLE_3D),
strain rate (TABLE) and plastic strain (CURVE) dependence.

Allow *MAT_224 to be used with Lode parameter dependent tables even for shell elements,
i.e., in cases where LCF refers to *DEFINE_TABLE and/or LCI refers to
*DEFINE_TABLE_3D.

Allow *MAT_224 implicit to be used with EFG and MEFEM (ELFORMs 41, 42, 43, 45; see
*SECTION_SOLID).

Change behaviour of NUMINT>1 for *MAT_224 if used with multi-integration point solid
elements. Integration points that reach damage=1 still update stresses and therefore carry load.

Only if NUMINT integration points reach damage=1, the element gets eroded. The old way (single IPs got zero stresses before the element was deleted) just led to severe deformations and instabilities.

Fix floating invalid problem in *MAT_224 (solids and LCG>0): plastic strain rate could have become negative in rare cases.

Make *MAT_251/*MAT_TAILORED_PROPERTIES available for solid elements.

Enable numerical tangent for *MAT_252/*MAT_TOUGHENED_ADHESIVE_POLYMER allowing implicit analysis with moderate nonlinearities.

Fixed problem using *MAT_258/*MAT_NON_QUADRATIC_FAILURE and *DAMPING_PART_STIFFNESS together with RYLEN=2 in *CONTROL_ENERGY.

Modification for *MAT_280/*MAT_GLASS: Parameter NIPF now also applies to the EPSCR failure criteria, i.e., elements get deleted when NIPF integration points failed due to reaching EPSCR.

*MAT_ADD_INELASTICITY introduced for low order solids and shells. This keyword can be used to add inelastic behaviours, including plasticity, viscoelasticity, and creep, to any material model whose stress is calculated incrementally.

*MAT_ADD_DAMAGE_DIEM now supports Hughes-Liu beams, thick shell type 2, and axisymmetric elements.

All parameters on first two cards on *MAT_ADD_EROSION are supported for Instability on *DEFINE_MATERIAL_HISTORIES.

*MAT_EXTENDED_3-PARAMETER_BARLAT/*MAT_036E now supports tables to define stress vs. plastic strain and strain rate. Viscoplastic consistency is mandatory.

Enhanced message SOL+737. "Strain range of stress-strain input for *MAT_031 has been exceeded."

In checking for fixed nodes for PML boundary (*MAT_PML_...), correctly merge translational constraints from *NODE and *BOUNDARY_SPC, in case both are used at the same node.

Calculate internal energy correctly for PML elements. *MAT_PML_NULL may have issues.

Implement *DAMPING_FREQUENCY_RANGE_DEFORM for solid PML elements.

Introduce new label 'shl2d' for user materials in plane stress elements, distinguishing them from shells.

Add keyword "*MAT_ADD_CHEM_SHRINKAGE_TITLE", with which the chemical shrinkage effect can be simulated.

`*MAT_242/*MAT_KINEMATIC_HARDENING_BARLAT2000:`

- Define Young's modulus with a curve. If EA is less than zero, the absolute value of EA is the curve ID defining Young's modulus as a function of plastic strain.

- Add option for modified Yoshida formulation with two variables SC1, SC2 (same as `*MAT_125`).

1.4.25 MPP

When MPP predecomposition is done for more processors than the calculation, and the model contains airbags, the airbag ownership was not being properly handled leading to problems including possibly segfault. This has been fixed.

Improvements and fixes for MPP handling of decomposition regions and processor subsets.

Fix MPP contact initialization issue that was segfaulting in some models if parts were being excluded from the d3plot output.

Add new pfile option "transform_keyword" to the "decomp" section. Including this new flag will cause the global decomposition transformations to apply to all the decomposition regions created by these keywords:

- `*PARTS_DISTRIBUTE`
- `*PARTSET_DISTRIBUTE`
- `*ARRANGE_PARTS`
- `*CONTACT_DISTRIBUTE`

Before r86885, and after that was reverted in r102224, these regions were decomposed without any coordinate transformation at all.

Between those versions, the global transformation applied to all these regions. I think this behaviour makes sense, but someone complained that r86885 changed behaviour, so it was removed in r102224. Then someone ELSE complained that r10224 changed the behaviour that THEY wanted, so this option is now added.

Fix missing initialization that broke d3plot files when MPP predecomposition was used with models having `*DEFINE_CURVE_FUNCTIONS` that used the PIDCTL function.

Corrected MPP processing of `*PART_MODES` option of ANSID .ne. 0. The node list for this option needed to be sorted and logic adjusted so that all processes, even those with no such nodes, are involved in the communication.

1.4.26 Output

`*DATABASE_TRACER_GENERAL`: In this keyword, tracers are nodes that output histories for any solids or beams or shells or thick shells with the tracers in their volumes.

`*DATABASE_TRACER_GENERAL` and `*CONTROL_REFINE_SHELL`: Implement the tracers in the shell refinement.

*DATABASE_TRACER_GENERAL and *CONTROL_REFINE_SOLID: Implement the tracers in the solid refinement.

*DATABASE_PROFILE: Implement TYPE=6 to output the distribution (along a given direction) of thick shell data.

*DATABASE_PROFILE: Output strain profiles if STRFLG=1 in
*DATABASE_EXTENT_BINARY.

MPP fix for frictional work output to intfor (*DATABASE_BINARY_INTFOR) when soft=2 contact is being used

"Interface Pressure" output to intfor is now returned to always being positive. There is no clear way to distinguish between compression and tension at this point in the code, so restoring the old behaviour seems best.

Fix issue with reported time step controlling element in glstat.

Reaction forces for superelements (*ELEMENT_DIRECT_MATRIX_INPUT) using explicit time integration are now available in spcforc.

SPCs in *CONSTRAINED_NODAL_RIGID_BODY_SPC are now internally converted to equivalent *BOUNDARY_PRESCRIBED_MOTION_RIGID if SPC2BND is set to 1 on *CONTROL_OUTPUT. This allows output of reactions to bndout.

New stress extrapolation from gauss points to nodes for shells (similar to the SOLSIG option for solids). Enable by setting the SHLSIG variable to 1 in *CONTROL_OUTPUT. Applies to shell element formulations 16, 20, and 21.

*DATABASE_RECOVER_NODE:

- Fix an MPP bug for the nodal stress calculation of 10-node tetrahedron element.
- Add 3 more nodal stress components to be recovered by using the velocity vector.
- Add elemental extrapolation method for stress recovery.

Error terminate with KEY+501 message if different BEAMIP settings are used in duplicate *DATABASE_EXTENT_BINARY.

*ele

Eliminate order dependency of IDOF in *SECTION_SHELL in which the IDOF in the first shell part became the default for the other shell with the same formulation instead of being independent.

Fix incorrect rigid body velocities output to matsum for axisymmetric analysis with eroding elements.

Fix incorrect rigidwall forces & stonewall energy reported for
*RIGIDWALL_PLANAR_ORTHO.

Shell reference surface offset NLOC in *SECTION_SHELL will now properly affect moment calculation for output to secforc (*DATABASE_SECFORC).

Eliminated output of nodes having zero mass to massout (*DATABASE_MASSOUT).

Fixed a bug that caused inconsistencies in strains output to d3plot and d3part.

Fixed a bug that caused discrepancies in elout and disbout for discrete beams using *MAT_121/*MAT_GENERAL_NONLINEAR_1DOF_DISCRETE_BEAM.

Fixed a bug that caused zero forces in nodfor for discrete beams (beam ELFORM 6).

Fixed a bug that caused incorrect stresses in d3plot and elout for shell ELFORM 23.

Fixed a bug that caused discrepancy in rigid body velocity output to glstat and matsum.

Fixed a bug that caused segfault when using LCUR in *DATABASE_DISBOUT.

Added missing report of triggered failure criterion when solid elements fail due to *MAT_ADD_EROSION.

Fixed a bug that caused incorrect filtered results in nforc when using *DATABASE_NCFORC_FILTER in MPP.

Enabled the change in beam length to be viewed when fringe plotting discrete beams.

Enabled variable ENGFLG on *DATABASE_EXTENT_BINARY to apply to thick shells as well as thin shells. This allows internal energy density of tshells to be written d3plot.

Fixed a thick shell output problem affecting d3plot that occurred when MAXINT on *DATABASE_EXTENT_BINARY was larger than the number of layers in the element. For the nonexistent layers, we were reading data from the next element in memory. Now data for the last existing point is repeated.

Fixed two issues when reporting eroded hourglass energy to glstat and matsum. For solid elements, the eroded energy was not removed from the regular hourglass energy, so it was counted twice causing a poor energy balance in the glstat data. Also, thick shells were not reporting eroded hourglass energy.

Fixed bug in which DECOMP=2 or 4 in *DATABASE_EXTENT_BINARY corrupts d3drlf.

Fixed bug whereby a simple restart corrupted the d3part database.

Echo *DATABASE_EXTENT_BINARY_COMP variables to d3hsp.

1.4.27 Restarts

*DAMPING_FREQUENCY_RANGE_DEFORM now works smoothly with full deck restarts (*STRESS_INITIALIZATION). Applies to solids, beams, shells, thick shells. Previously, a stress-jump would occur at the beginning of the restart run.

Fix conditional so that node rotational masses are properly synchronized when initializing *CONTACT_TIED_..._BEAM_OFFSET during MPP full deck restart.

Fix MPP full deck restart issue that caused hanging if the new input file contained encrypted input.

Correct the handling of restarting implicit after full restart.

Fix bug affecting restarts in implicit.

Support the following for full deck restart and consequently,

*CONTROL_MPP_DECOMPOSITION_REDECOMPOSITION.

- SPH active region with local coordinate system (*DEFINE_SPH_ACTIVE_REGION, ICID).
- SPH injection (*DEFINE_SPH_INJECTION)
- Explicit joints (*CONSTRAINED_JOINT).
- LMF joint formulation (*CONTROL_RIGID, LMF=1).

Fix seg fault when using *DELETE_CONTACT for restart when running with SMP.

Fix error termination for full deck restart using *DEFINE_ELEMENT_DEATH.

Added new option COMP to *DELETE_PART for simple restart. *DELETE_PART_COMP
This option eliminates data pertaining to deleted parts from d3plot thereby reducing the size of the d3plot database.

1.4.28 *SENSOR

Limit the usage of *SENSOR to transient analysis only, not dynamic relaxation.

*SENSOR_DEFINE_CALC-MATH: fix a bug triggered when CALC=SUM and any of involved sensors, SENSi, are negative for subtraction.

*SENSOR_DEFINE_ELEMENT: fix a bug triggered when trying to trace the stress/strain of a failed element.

*SENSOR_DEFINE_FORCE: fix an MPP bug for TYPE=CONTACT2D.

*SENSOR_DEFINE_MISC: fix a bug in which incorrect kinetic energy of a rigid body was sensed.

*SENSOR_SWITCH_CALC-LOGIC: allow as many switches as needed for logic calculation.

*SENSOR_CONTROL:

- Fix a bug for TYPE=BAGVENTPOP in which venting is not turned off for *AIRBAG_HYBRID.

- Fix a bug for TYPE=PRESC-MOT triggered when the referenced node or rigid part has a big ID.

- Fix a bug for TYPE=BELTPRET that fails to turn on the pretensioner.
- Fix a bug for TYPE=BELTRETRA that fails to lock the retractor when NEWLEG of *CONTROL_OUTPUT>0 and no SBTOUT output is requested.
- Add CNRB option to control *CONSTRAINED_NODAL_RIGID_BODY.
- Add TYPE=LOADTHM to control *LOAD_THERMAL_VARIABLE and *LOAD_THERMAL_VARIABLE_NODE
- Add optional time delay.

Fix order dependency problem when using *DEFORMABLE_TO_RIGID activated by *SENSOR_CONTROL TYPE='DEF2RIG'.

1.4.29 SPG (Smooth Particle Galerkin)

Volume of 6-noded solid elements using element formulation 42 (SPG) was calculated incorrectly. That bug is now fixed.

Fix SPG bug in MPP whereby the time step size varies with different number of MPP processes.

*CONSTRAINED_IMMERSSED_IN_SPG: Fix fatal memory issue if the number of immersed nodes is much larger than the number of nodes of the SPG part.

Enabled additional material models for SPG application, some of which include orthotropic behaviour:

*MAT_110, *MAT_122, *MAT_123, *MAT_143, *MAT_199, *MAT_260A, *MAT_269.

*SECTION_SOLID_SPG now automatically sets element formulation to 47.

If FAIL is set greater than 0 in *MAT_003 or *MAT_024, it is used in the SPG bond failure.

Added SPG bond failure criteria corresponding to *MAT_ADD_DAMAGE_GISSMO and *MAT_ADD_EROSION (1st principal stress, max shear strain, 3rd principal strain).

1.4.30 SPH (Smooth Particle Hydrodynamics)

Incompressible SPH (FORM=13) is available in beta form.

Add two material models for implicit SPH formulation (FORM=13):

- *MAT_300/*MAT_SPH_02/*MAT_SPH_IMPLICIT_FLUID: Input includes surface tension coefficient and surface area minimization coefficient.
- *MAT_301/*MAT_SPH_03/*MAT_SPH_IMPLICIT_STRUCTURE: Used for boundary particles. Input includes surface adhesion coefficient.

Add *DATABASE_SPHMASSFLOW and *DEFINE_SPH_MASSFLOW_PLANE to measure SPH mass flow rate across a defined plane. This feature is to SPH as *DATABASE_DEMASSFLOW is to DEM.

Add option to move *DEFINE_SPH_ACTIVE_REGION around by following a moving coordinate system.

Add option for buffer zone in *DEFINE_SPH_ACTIVE_REGION.

Add an error message when a *DEFINE_BOX_SPH is referenced in *CONTROL_SPH boxid. This should point to a regular *DEFINE_BOX.

Add option to reactivate particles and add an optional buffer zone in *DEFINE_BOX_SPH.

*DEFINE_BOX_SPH gets a special treatment with FORM=13. Particles outside of the activation box get frozen in space, and deactivated particles can become active when entering the box. This allows calculating only particles surrounding the region of interest.

Reduce d3plot size for incompressible SPH by trimming down irrelevant variables.

*DEFINE_ADAPTIVE_SOLID_TO_SPH:

- Zero out increment of internal energy for eroded particles.
- Bug fix in SMP: Results were inconsistent in SMP when ncpu was greater than one.
- Bug fix in stress calculation of embedded SPH particles when ICPL = 1. Internal energy of these particles was also inconsistent.
- When elements erode, distribute their internal energy to their child particles.
- Reset IOPT=0 if ICPL=0.
- Fix the MPP 3D strain calculation when adaptive with pure thermal analysis is used. Remove contribution of adaptive particles to strain rate calculation.
- Properly update density of embedded particles based on element current volume. When particles became active, they used to still carry the rest density.
- Enforce embedded SPH particles to output zero internal energy, for energy conservation.

*DEFINE_SPH_INJECTION:

- Speed-up initial smoothing length calculation when *DEFINE_SPH_INJECTION is present. It used to be very slow for models containing a large number of injected particles.
- Add option to define a variable speed of injection. If the injection speed scale factor is defined as negative, it refers to a *DEFINE_CURVE defining the velocity magnitude vs. time.
- Use yet-to-be injected particles as ghost particles on the other side of the injection plane.
- If the coordinate system is updated in time (IFLAG=1 in *DEFINE_COORDINATE_NODES), we also update the injection direction.

- If the injection plane follows a node (NID in *DEFINE_SPH_INJECTION), and IFLAG=1 in *DEFINE_COORDINATE_NODES, we also continuously rotate the whole stack of particles to be injected with the plane.

Added SPH support for *MAT_193/*MAT_DRUCKER_PRAGER.

Bug fix in psetid option of *DATABASE_BINARY_D3PLOT if psetid contains SPH parts.

Fix ASCII output of SPH neighbour count, which was always zero.

Bug fix in SPH neighbour search when variable smoothing length is employed. The neighbour list was created based on minimum smoothing length instead of maximum. As a result, some neighbours were potentially omitted.

Add option in *CONTROL_MPP_DECOMPOSITION_REDECOMPOSITION to remove dead SPH particles from the model at each redecomposition step.

Improve MPP load balancing when a *CONTROL_MPP_DECOMPOSITION_REDECOMPOSITION is present and a moving box defines the activation region of SPH.

Create MPP variants of 2D plane-strain and 2D axisymmetric SPH formulations with thermal coupling.

Add support for *LOAD_THERMAL_LOAD_CURVE in SPH.

Fix bug for SPH thermal formulation in MPP. Artificial viscosity calculation was incorrect.

Fix bug in MPP with ERODING contact and SPH. In some MPP decompositions, some processors were ignoring contact between SPH and Lagrangian parts.

Restored TEROD and CEROD parameters for *MAT_NULL with SPH particles (these two parameters were not applied from R9).

Material models added for SPH:

- *MAT_COMPOSITE_DAMAGE/MAT_022 for SPH plane strain.
- *MAT_MODIFIED_CRUSHABLE_FOAM/*MAT_163 for SPH axisymmetric, plane strain, and 3D.
- *MAT_ENHANCED_COMPOSITE/*MAT_054, 055 for SPH axisymmetric and 3D

*CONTROL_SPH:

- Improvements in adaptive smoothed particle hydrodynamics formulations (FORMs 9,10) with an anisotropic kernel. Speed up SMP operations; optimize bucket cycle; update principle direction and scale factor.

- Renamed MEMORY variable to NMNEIGH and added error/warning messages to help clarify meaning for users.

New kernel functions (SPHKERN in *SECTION_SPH) for 3D SPH.

- SPHKERN=1 (quintic spline kernel function) now supports FORMs 5,6. (Previously only supported for FORMs 0,1,9,10.
- Added SPHKERN=2 for FORMs 0,1,5,6. This quadratic spline kernel function mainly aims for high velocity impact to relieve the problem of compressive instability.
- Added SPHKERN=3 for FORMs 0,1,5,6. This quartic smoothing kernel function (G.R.Liu 2003) is similar to the standard cubic spline kernel function but should be more stable and accurate.

*DEFINE_SPH_INJECTION: (both SMP and MPP):

-
- Add injection plane moving with a node NID.
 - Add injection velocity based on a local coordinate system CID.

Fix inexplicable penetration when using

*CONTACT_AUTOMATIC_NODES_TO_SURFACE with SPH elements. This affects SMP only.

1.4.31 Thermal Solver

*CONTROL_EXPLICIT_THERMAL_SOLVER and *ELEMENT_TSHELL: Implement heat conduction for thick shells in the explicit thermal solver.

*CONTROL_EXPLICIT_THERMAL_SOLVER can use *INITIAL_TEMPERATURE to initialize the element temperatures (it averages the initial nodal temperatures at element centres).

*CONTROL_EXPLICIT_THERMAL_INITIAL with ID<0 allows initialization of temperature by element instead of element set (ID>0).

New SOLVER options in *CONTROL_THERMAL_SOLVER:

- 18: preconditioner based on a local incomplete factorization;
- 19: preconditioner based on a block low-rank factorization of the global problem.

*CONTROL_THERMAL_SOLVER:

-
- Fixed the echo of thermal solver options to d3hsp.
 - Enhanced the setting of default for GMRES parameters when nonsymmetric iterative was internally chosen over the input value for thermal linear equation solver.
 - Migrated the linear algebra data structures in thermal mechanics to Fortran 95 dynamic storage as part of the storage modernization effort. Reduces requirement of user on setting the memory command line parameter.
 - Declare all old SMP linear equation solvers for thermal as obsolete. Automatically switch to the new solvers with an appropriate warning. This is a step towards modernization of linear algebra for thermal.
 - Added nonsymmetric problem support to the modern linear algebra path in thermal. This is mostly the THERMAL_BULK feature.

- Turned off support for direct linear equation solvers for thermal in single precision. In single precision change direct solver options to appropriate iterative solvers. Symmetric (11) goes to modified Incomplete Choleski PCG (16) which is the strongest symmetric iterative solver. Nonsymmetric (20) goes to GMRES (17).

- Correct the case where cgtol is specified on *CONTROL_THERMAL_SOLVER but reltol is not.

- Corrected testing of linear equation solver option for thermal. The addition of GMRES option for conjugate heat transfer problems had to be properly dealt with.

Turn off all implicit mechanics if a thermal-only solution.

*CONTROL_THERMAL_SOLVER: Fix a conflict associated with using the nonsymmetric direct solver (SOLVER=30) for the thermal side of a coupled thermal/mechanical analysis when the mechanical analysis is implicit.

*BOUNDARY_RADIATION_<option>_VF_READ requires a view factor file. An input trap was added if this view factor file is missing.

There was a limit on *BOUNDARY_TEMPERATURE_NODE and *INITIAL_TEMPERATURE_NODE keyword appearances. This limit was removed, and the speed of the keyword reader was improved.

Added new keyword *CONTROL_THERMAL_FORMING for easy setup of hot/warm forming simulations. This keyword sets default values for thermal control keywords so that there is no need to include *CONTROL_SOLUTION, *CONTROL_THERMAL_SOLVER and *CONTROL_THERMAL_TIMESTEP in the input deck unless fine tuning of simulation parameters are needed. This keyword can also be used to set up default thermal parameters for forming contact pairs.

Added new keyword *BOUNDARY_TEMPERATURE_TRAJECTORY. Purpose: Apply a temperature boundary condition on nodes enclosed in either a cylindrical or rectangular prism volume moving along a trajectory. The centre of the volume moves along the trajectory defined by a nodal path at a prescribed velocity. The geometry of the enclosing volume can be time-variant. This keyword applies only to solid elements.

New keyword *BOUNDARY_FLUX_TRAJECTORY defines a moving flux surface boundary condition along a nodal path. Based on the shape of the heat source and its aiming direction, the projection onto the surface is calculated and used for application of the flux. Additional features include:

- Propagation to newly exposed surfaces after element erosion.
- Option to balance the change in projected area by modifying the surface density.
- Total amount of heat input as resulting from the numerical integration of the surface densities can be enforced to coincide with the input.
- Various heat distributions are available:
 - + Constant distribution within a double elliptic region.
 - + Gaussian distribution within a double elliptic region.
 - + User-defined function (*DEFINE_FUNCTION) based on local coordinates, time, and temperature.

***LOAD_HEAT_GENERATION:**

Added variable REFNODE to define a reference node for the function definition. The current nodal coordinates can be referred to in the function definition as 'xref','yref','zref'. It allows defining a heat source motion associated with the motion of the reference node.

Thermal contact for 'edge' contact of composite tshell elements now considers the composite stacking sequence by way of an internally constructed mesh of stacked tshell elements.

1.4.32 XFEM (eXtended Finite Element Method)

***BOUNDARY_PRECRACK:** Adjusted the location of pre-crack to avoid passing through nodal points.

Recoded neighbours list to handle triangular mesh better.

Added support of GISSMO damage model for XFEM. Set FAILCR=0 in ***SECTION_SHELL_XFEM** to activate GISSMO model.

2D XFEM shell (ELFORM=52 in ***SECTION_SHELL_XFEM**) is a fully integrated form and so NIP=4 is now hardwired.

Second order stress update can now be specified for 2D XFEM shell form 52 (OSU=1 in ***CONTROL_ACCURACY**).

1.4.33 Miscellaneous

***DAMPING_FREQUENCY_RANGE** and ***DAMPING_FREQUENCY_RANGE_DEFORM** had an error trap related to the time step being too large to compute damping at frequency FHIGH. The trap was unnecessary for ***DAMPING_FREQUENCY_RANGE_DEFORM** and has been removed. The trap was previously necessary for ***DAMPING_FREQUENCY_RANGE** to avoid instability, but now the code has been modified to remain unconditionally stable and the error trap has been removed. The code modification will result in some change of results for ***DAMPING_FREQUENCY_RANGE** (but not for the **_DEFORM** option). The difference will be most noticeable when the product timestep*FHIGH is large, for example in Implicit calculations.

Added new command line option to run only a subset of the cases defined in the input deck via ***CASE**. As before, if all cases in a model should be run, the single word "CASE" should appear as a command line argument. But if only some cases should be run, those case ids can be listed on the command line like: CASE=17,22,147 There should be NO spaces before or after the =, and the list following the = should be a comma delimited list of case ids, with no spaces. LS-DYNA now returns an exit code of 1 to the system in case of non-normal termination. Previously, we always returned 0 (no error) no matter what, which could be misleading if a user's job script made use of the exit code.

Suppress d3hsp output of any ***PARAMETER** that was encrypted in the input.

Fix old bug in *INTERFACE_LINKING that would cause incorrect behaviour for displacements scaled via a *DEFINE_FUNCTION if (and only if) the function was defined in terms of 3 variables (spatial displacement dependence only).

The third variable in *PART_ADAPTIVE_FAILURE now does this:

1 = disable adaptivity of THIS part after it is split into two pieces

2 = disable ALL adaptivity after this part is split into two pieces

The default value is 0, which does nothing.

Add keyword *DEFINE_DRIFT_REMOVE to restore periodicity to specified curve(s) that physically should exhibit periodicity. For example, accelerometer data collected as a car drives one circuit around a track should exhibit periodicity such that when integrated to get displacement, that displacement is zero. However, noise in the data leads to nonphysical drift. This keyword activates an algorithm that approximately restores periodicity.

*INTERFACE_LINKING_NODES: Interface nodes that were shared between processes were not properly dealt with for implicit.

*DEFINE_CURVE_FUNCTION: fix a bug for SENSOR or SENSORD that may result in erroneous information in CURVOUT.

*DEFINE_CURVE_FUNCTION: Fix a bug for function PIDCTL that occurred in a thermal-only analysis.

*PART_DUPLICATE:

- Fix a bug in which 10-noded tet elements were not duplicated.

- Add option of BOXID to assure that the transformed configuration falls inside the box

*SET_PART_TREE: Define a branch in a tree structure. A branch is a part set that can be defined using parts and/or sub-branches. With this keyword, the whole model can be modelled as a hierarchical tree structure.

*SET_SHELL: allow *SET_SHELL_LIST_GENERATE to generate a set of 2D belts.

New keyword *DEFINE_QUASAR_COUPLING to enable coupling of LS-DYNA and Cadlm's QUASAR ROM (Reduced Order Model). Supports multiple ROMs attached to LS-DYNA FEM model.

Madymo coupling:

- Support TASS Madymo coupling using Intel MPI.

- Support Madymo 7.7 and above.

Fixed outdated argument lists for user subroutines. Before this fix user-defined features using *MODULE may not work correctly.

*DEFINE_PRESSURE_TUBE now supports decomposition of automatically generated solid/shell tubes in MPP.

Added support for user element history variables in *USER_NONLOCAL_SEARCH.

Implement *DEFINE_DRIFT_REMOVE for MPP.

Fix empty *SET_SEGMENT error, KEY+140, when using *SET_SEGMENT_GENERAL with OPTION set to SET_SOLID/SET_SLDI0/SET_SLDF#/SET_TSHELL/SET_TSHIO.

Fix seg fault when using *INTERFACE_LINKING_NODE_SET.

New functions AX2, AY2, and AZ2 for *DEFINE_CURVE_FUNCTION. The difference compared to AX, AY, and AZ is that if n2=0, then return component relative to axes fixed in n1.

New function DIST() in *DEFINE_CURVE_FUNCTION, which is used to calculate the distance travelled by a node

Add feature to *DEFINE_MATERIAL_HISTORIES, LABEL='History' whereby if the first attribute A1 is negative, its absolute value points to a curve. The first ordinate value of that curve defines an operation to be performed on a list of history variables. The subsequent ordinate values give the history variable numbers in the list. See the User's Manual for details and an example.

Add keyword *DEFINE_ELEMENT_EROSION_(SHELL/TSHELL) to allow more flexible control of when to actually delete an element.

Add two new OPTIONS for *DEFINE_TRANSFORMATION.

- TRANSL2ND: translation given by two nodes and a distance
- ROTATE3NA: rotation given by three nodes and an angle (two nodes provide rotation axis, going through the third node)

Add new option DTYPE=1 to *PERTURBATION_NODE to allow uniform distribution between $SCL \times [-AMPL, AMPL]$ for TYPE=8.

Add runtime output to "Open include file: ..." instance.

Removed the echo of encrypted data to d3hsp for the following keywords: *PARAMETER, *DEFINE_FRICTION, *ELEMENT_DISCRETE, *CONTACT, *DEFINE_CURVE.

Added three sense switches swb, swc and swd, all of which will create dynain data.

*INTERFACE_SPRINGBACK_LSDYNA must be defined in the input deck.

swb: a dynain is written and the analysis continues.

swc: a restart and dynain are written and the analysis continues.

swd: a restart and dynain are written and the analysis terminates.

The first dynain file created by one of these sense switches is named dynain.1, and the filename is incremented by 1 (dynain.2, dynain.3, etc.) each time one of the sense switches is submitted.

Replace keyword *SET_SPRING with *SET_DISCRETE.

Changed the default of plabel to plabel=no for faster input processing. If the input contains non-numeric (character) ID(s), an error message will be issued, in which case the user would need to include "plabel=yes" on the execution line so that the non-numeric ID(s) could be properly read.

Fix bug in which message announcing loading of an include files appeared twice.

Fixed bug in madymo coupling in which the restart dump was called twice.

Improve check for empty or duplicate *SETs and issue appropriate warnings/errors.

Can now use an environment variable to set the default memory:

setenv LSTC_DEFMEM n where n is the memory size

Can now suppress almost all output by including "benchmark=y" on either the execution line or on the *KEYWORD line in the input deck. It will also remove part, node and element output in d3hsp.

Fixed bug in reading long format if *KEYWORD long=yes is used in include file.

A new flag TOL in *INCLUDE_UNITCELL is introduced for users to define the tolerance for identifying the pairs of nodes in the periodic positions.

New keywords: *COSIM_FMI_CONTROL and *COSIM_FMI_INTERFACE:

Adds capability to remotely co-simulate with other software supporting FMI standard.

Deliver model order reduction (MOR) *CONTROLLER_PLANT for the linear structural and piezo-electric system. The state space matrices are exported based on the modal truncation or Krylov subspace.

*PART_DUPLICATE: Add a new optional variable ZMIN. Transformed part(s) will have a minimum z-coordinate equal to ZMIN.

*INCLUDE_STAMPED_PART_SOLID_TO_SOLID:

This keyword extends the idea of *INCLUDE_STAMPED_PART to solid elements.

Information about a solid part is mapped from one analysis (e.g., stamping) to another analysis (e.g., crash).

- Maps the stress tensor, effective plastic strain, and strain tensor.

- Calculates the thickness from the first analysis and modifies the coordinates of the nodes in the second model to conform to that thickness.

- When PID is a negative number, it means that the top surface of the solid needs to be reversed to match the forming result. Otherwise, the mapping will be wrong.

- When the element orientation is defined in *ELEMENT_SOLID_ORTHO, the angles are mapped to the new mesh.

*PART_MOVE:

- Maximum number of parts moved in one input increased from 400 to 1000.

- When CID is a negative value, it defines a vector id or VID pointing to the direction of the part move and the variable ZMOV is the move distance.