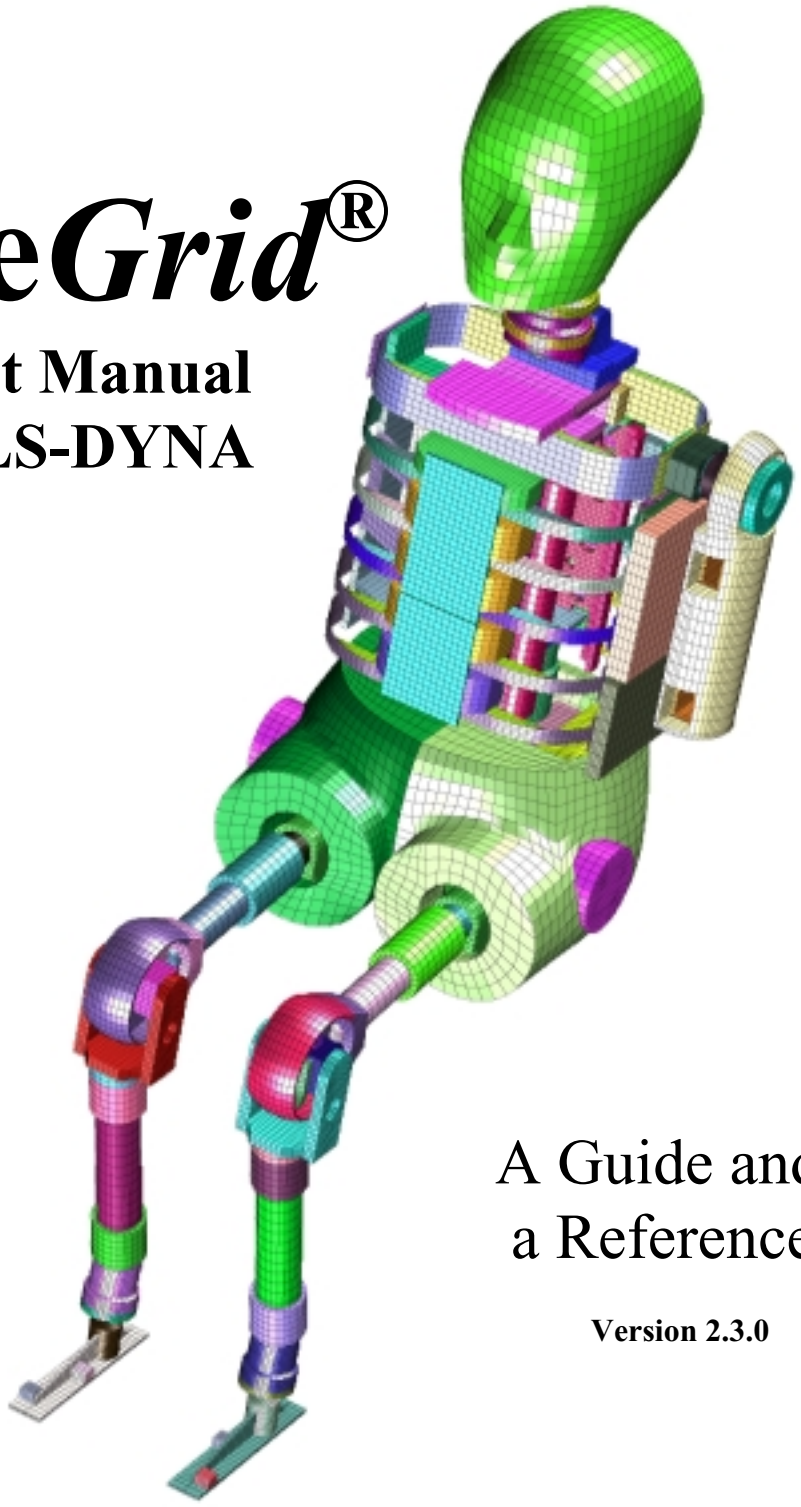
TrueGrid®

Output Manual For LS-DYNA



A Guide and a Reference

Version 2.3.0

Robert Rainsberger and Mike Burger

XYZ Scientific Applications, Inc.

Table of Contents

I. LS-DYNA Output Guide

Introduction

LS-DYNA is a nonlinear, explicit and implicit, two and three-dimensional finite element code for solid, structural, and fluid mechanics to simulate dynamic, static, heat transfer, electrostatic, and electromagnetic problems developed by Livermore Software Technology Company. The focus in this manual will be on those features in **True*Grid*®** that are specific to creating a LS-DYNA input file. The **True*Grid*®** User Manual covers the creation of a mesh and will not be covered in this manual. This manual is incomplete in another way because it cannot be used as a substitute for the LS-DYNA manual. For a full understanding of the use of these features, the user must have a working knowledge of LS-DYNA and be familiarity with a LS-DYNA User Manual.

A common source of confusion is the notion of a part and a material. A LS-DYNA part is equivalent to a **True*Grid*®** material. In contrast, a **True*Grid*®** material is defined using the **lsdymats** command. Then any group of elements (a region in the part phase of an element set in the merge phase) can be assigned this material. Also in contrast, a **True*Grid*®** part is formed using the **block** or **cylinder** command. A **True*Grid*®** part can contain many different materials or many **True*Grid*®** parts can all be assigned the same matterial.

Font Conventions

Different fonts are used through out this manual to indicate their meaning. A literal is highlighted in bold. A symbol to be substituted with a literal or a number is *italicized*. A computer example uses the `Courier font`. A button in from the Graphical User Interface is both ***italic and bold***.

Supported Features in LS-DYNA

There are many features in **True*Grid*®** to create a model for LS-DYNA. The table below shows the commands that are used for each feature in LS-DYNA. Sometimes there are several commands listed. For example, shells can be generated using both the **block** and **cylinder** commands. The **n** and **th** are used to set the properties of these shells. In another example, the **si** and **sii** commands are used to identify the faces of the mesh that form the sliding (or contact) surfaces. The associated **sid** command is used to assign properties to the sliding surface.

True*Grid*® is specifically a three-dimensional mesh generator and preprocessor. A two-dimensional mesh can be generated in **True*Grid*®** for LS-DYNA by forming a shell mesh where the z-coordinate is set to 0. The easiest way to be sure that the z-coordinates are kept at zero is to project the entire shell mesh onto the plane where z=0 (`sd # plan 0 0 0 0 0 1 sfi ;;-1; sd #`).

LS-DYNA feature	**True*Grid*®** commands
*ALE_SMOOTHING	**sc**
*BOUNDARY_CONVECTION_SEGMENT	**cv, cvi**
*BOUNDARY_FLUX_SEGMENT	**fl, fli, arri, dist**
*BOUNDARY_NON_REFLECTING	**nr, nri**
*BOUNDARY_PRESCRIBED_MOTION_NODE	**frb, fv, fvi, fvc, fvci, fvs, fvsi, acc, acci, accc, accci, accs, accsi, fvv, fvvi, fvvc, fvvci, fvvs, fvvsi, vacc, vacci, vaccc, vaccci, vaccs, vaccsi, fd, fdi, fdc, fdci, fds, fdsi.**
*BOUNDARY_PRESCRIBED_MOTION_RIGID	**lsdymats**
*BOUNDARY_RADIATION_SEGMENT	**rb, rbi**
*BOUNDARY_SLIDING_PLANE	**plane, sfb**
*BOUNDARY_SPC_NODE	**lb, lsys, sfb**
*BOUNDARY_SPC_SET	**plane**
*BOUNDARY_SYMMETRY_FAILURE	**plane, syf, syfi**
*BOUNDARY_TEMPERATURE_NODE	**ft, fti**
*CONSTRAINED_EXTRA_NODES_SET	**jt, jd**
*CONSTRAINED_JOINT_SPHERICAL	**jd, jt**
*CONSTRAINED_JOINT_REVOLUTE	**jd, jt**
*CONSTRAINED_JOINT_CYLINDRICAL	**jd, jt**
*CONSTRAINED_JOINT_PLANAR	**jd, jt**
*CONSTRAINED_JOINT_UNIVERSAL	**jd, jt**
*CONSTRAINED_JOINT_TRANSLATIONAL	**jd, jt**
*CONSTRAINED_NODAL_RIGID_BODY	**rigid, nset, nseti**
*CONSTRAINED_NODAL_RIGID_BODY_INERTIA	**rigid, nset, nseti**
*CONSTRAINED_NODE_SET	**jt, jd, mpc, nset, nseti**
*CONSTRAINED_RIGID_BODIES	**rigbm**
*CONSTRAINED_SHELL_TO_SOLID	**shtoso, shtosoi**
*CONSTRAINED_SPOTWELD	**spotweld, spw, spwd, jt, jd**
*CONSTRAINED_SPOTWELD_FILTERED_FORCE	**spotweld**
*CONSTRAINED_TIED_NODES_FAILURE	**fn, fni**
*CONTACT	**si, sii, sid, orpt**
*CONTACT_1D	**sid** (rebar option), **ibm, ibmi, jbm, jbmi, kbm, kbmi**
*CONTACT_AIRBAG_SINGLE_SURFACE	**sid, orpt, si, sii**
*CONTACT_AUTOMATIC_GENERAL	**sid, orpt, si, sii**
*CONTACT_AUTOMATIC_GENERAL_INTERIOR	**sid, orpt, si, sii**
*CONTACT_AUTOMATIC_NODES_TO_SURFACE	**sid, orpt, si, sii**
*CONTACT_AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE	**sid, orpt, si, sii**

*CONTACT_AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE_TIEBREAK **sid, orpt, si, sii**
*CONTACT_AUTOMATIC_SINGLE_SURFACE **sid, orpt, si, sii**
*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE **sid, orpt, si, sii**
*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_TIEBREAK **sid, orpt, si, sii**
*CONTACT_CONSTRAINT_NODES_TO_SURFACE **sid, orpt, si, sii**
*CONTACT_CONSTRAINT_SURFACE_TO_SURFACE **sid, orpt, si, sii**
*CONTACT_DRAWBEAD **sid, orpt, si, sii**
*CONTACT_ERODING_NODES_TO_SURFACE **sid, orpt, si, sii**
*CONTACT_ERODING_SINGLE_SURFACE **sid, orpt, si, sii**
*CONTACT_ERODING_SURFACE_TO_SURFACE **sid, orpt, si, sii**
*CONTACT_FORCE_TRANSDUCER_CONSTRAINT **sid, orpt, si, sii**
*CONTACT_FORCE_TRANSDUCER_PENALTY **sid, orpt, si, sii**
*CONTACT_FORMING_NODES_TO_SURFACE **sid, orpt, si, sii**
*CONTACT_FORMING_ONE_WAY_SURFACE_TO_SURFACE **sid, orpt, si, sii**
*CONTACT_FORMING_SURFACE_TO_SURFACE **sid, orpt, si, sii**
*CONTACT_NODES_TO_SURFACE **sid, orpt, si, sii**
*CONTACT_NODES_TO_SURFACE_INTERFERENCE **sid, orpt, si, sii**
*CONTACT_ONE_WAY_SURFACE_TO_SURFACE **sid, orpt, si, sii**
*CONTACT_ONE_WAY_SURFACE_TO_SURFACE_INTERFERENCE **sid, orpt, si, sii**
*CONTACT_RIGID_BODY_ONE_WAY_TO_RIGID_BODY **sid, orpt, si, sii**
*CONTACT_RIGID_BODY_TWO_WAY_TO_RIGID_BODY **sid, orpt, si, sii**
*CONTACT_RIGID_NODES_TO_RIGID_BODY **sid, orpt, si, sii**
*CONTACT_SINGLE_EDGE **sid, orpt, si, sii**
*CONTACT_SINGLE_SURFACE **sid, orpt, si, sii**
*CONTACT_SLIDING_ONLY **sid, orpt, si, sii**
*CONTACT_SLIDING_ONLY_PENALTY **sid, orpt, si, sii**
*CONTACT_SPOTWELD **sid, orpt, si, sii**
*CONTACT_SPOTWELD_WITH_TORSION **sid, orpt, si, sii**
*CONTACT_SURFACE_TO_SURFACE **sid, orpt, si, sii**
*CONTACT_SURFACE_TO_SURFACE_INTERFERENCE **sid, orpt, si, sii**
*CONTACT_TIEBREAK_NODES_ONLY **sid, orpt, si, sii**
*CONTACT_TIEBREAK_NODES_TO_SURFACE **sid, orpt, si, sii**
*CONTACT_TIEBREAK_SURFACE_TO_SURFACE **sid, orpt, si, sii**
*CONTACT_TIED_NODES_TO_SURFACE **sid, orpt, si, sii**
*CONTACT_TIED_SHELL_EDGE_TO_SURFACE **sid, orpt, si, sii**

*CONTACT_TIED_SURFACE_TO_SURFACE	**sid, orpt, si, sii**
*CONTACT_TIED_SURFACE_TO_SURFACE_FAILURE	**sid, orpt, si, sii**
*CONTROL_ACCURACY	**lsdyopts**, options **nosu, inn, pidosu**
*CONTROL_ADAPSTEP	**lsdyopts**, options **factin, dfactr**
*CONTROL_ADAPTIVE	**lsdyopts**, options **adpfreq, adptol, adpopt, maxlvl, tbirth, tdeath, lcadp, gsam, mnelsz, npss, ireflg, adpene, adpth, imem, orient, maxel**
*CONTROL_ALE	**lsdyopts**, options **dct, nadv, meth, afac, bfac, cfac, dfac, efac, tbeg, tend, aafac, vfact, vlimit, ebc**
*CONTROL_BULK_VISCOSITY	**lsdyopts**, options **q1, q2, ibq**
*CONTROL_CFD_AUTO	**lsdyopts**, options **itsflg, epsdt, dtsf, adtmax**
*CONTROL_CFD_GENERAL	**lsdyopts**, options **insol, dtinit, cfl, ickdt, iacurc**
*CONTROL_CFD_MOMENTUM	**lsdyopts**, options **mimass, iadvec, ifct, divu, thetak, thetaa, thetaf, msol, maxit, ichkit, iwrt, ihist, eps, ihg, ehg**
*CONTROL_CFD_PRESSURE	**lsdyopts**, options **ipsol, maxitr, ichcit, idiag, ihst, epsp, nvec, istab, pbeta, ssid, plev, plcid**
*CONTROL_CFD_TRANSPORT	**lsdyopts**, options **itemp, nspec, imss, ibaltd, iaflx, thetk, thtaa, thetf, itsol, mxiter, ickint, idiagn, ichist, epst, ihgt, ehgt**
*CONTROL_CFD_TURBULENCE	**lsdyopts**, options **itrb, smagc, sn1**
*CONTROL_COARSEN	**lsdyopts**, options **icoarse, fangl, sn1, sn2, sn3, sn4, sn5, sn6, sn7, sn8**
*CONTROL_CONTACT	**lsdyopts**, options **slsfac, rwpnal, islchk, shlthk, penopt, thkchg, orien, dkeep, usrstr, usrfrc, nsbcs, interm, xpene, tfst, itftss, itfpsn, dsfric, ddfric, dedc, dvfc, dth, dthsf, dpensf, ignore, frceng**
*CONTROL_COUPLING	**lsdyopts**, options **unleng, untime, unforc, timidl, flipx, flipy, flipz, sybcyl, mrpc, icsc, usaco, nsmcol**
*CONTROL_CPU	**lsdyopts**, options **cputim**

*CONTROL_DYNAMIC_RELAXATION	**lsdyopts**, options **nrcyck, drtol, drfctr, drterm, tssfdr, irelal, edttl, idrflg**
*CONTROL_ENERGY	**lsdyopts**, options **hgen, rwen, slnten, rylen**
*CONTROL_EXPLOSIVE_SHADOW	**lsdyopts**, options **expsh**
*CONTROL_HOURGLASS	**lsdyopts**, options **ihq, qh**
*CONTROL_HOURGLASS_936	**lsdyopts**, options **n36flg**
*CONTROL_IMPLICIT_AUTO	**lsdyopts**, options **iautf, iteropt, iterwin, dtmini, dtmaxi**
*CONTROL_IMPLICIT_DYNAMICS	**lsdyopts**, options **inal, newgam, newbet**
*CONTROL_IMPLICIT_EIGENVALUE	**lsdyopts**, options **neig, center, lflag, lftend, rflag, rhtend, eigmth, shfscl**
*CONTROL_IMPLICIT_GENERAL	**lsdyopts**, options **imflag, dt0, imform, nsbs, istress, cnstn, form, zerov**
*CONTROL_IMPLICIT_SOLUTION	**lsdyopts**, options **nsolvr, ilimit, maxref, dctoln, ectoln, lstoln, dnorm, diverg, istif, nlprt, arcctl, arcdir, arclen, arcmth, arcdmp, impln2, impln3**
*CONTROL_IMPLICIT_SOLVER	**lsdyopts**, options **lsolvr, lprint, negev, sorder, drcm, drcprm, autospc, autotol**
*CONTROL_IMPLICIT_STABILIZATION	**lsdyopts**, options **ias, ascale, strttim, endtime**
*CONTROL_NONLOCAL	**lsdyopts**, options **nmem**
*CONTROL_OUTPUT	**lsdyopts**, options **npopt, neecho, nrefup, iaccop, opifs, ipnint, ikedit, iflush, iprtf**
*CONTROL_PARALLEL	**lsdyopts**, options **ncpu, numrhs, iconst, ipllacc**
*CONTROL_REMESHING	**lsdyopts**, options **remin, remax**
*CONTROL_RIGID	**lsdyopts**, options **lmf, jntf, orthmd, partm, sparse**
*CONTROL_SHELL	**lsdyopts**, options **wrpang, itrist, irnxx, istupd, theory, bwc, miter, shproj, rotascl, intgrd, lamsht**
*CONTROL_SOLID	**lsdyopts**, options **esort**
*CONTROL_SOLUTION	**lsdyopts**, options **ianprc**

*CONTROL_SPH	**lsdyopts**, options **ncbs, boxid, sphdt, idim, sphmem, sphform, sphstart, sphmaxv, sphcont, sphderiv, sphini**
*CONTROL_STRUCTURED	**lsdyopts**, options **struct**
*CONTROL_STRUCTURED_TERM	**lsdyopts**, options **iterm**
*CONTROL_SUBCYCLE	**lsdyopts**, options **subcyl**
*CONTROL_TERMINATION	**lsdyopts**, options **endtim, endcyc, dtmin, endeng, endmas**
*CONTROL_THERMAL_NONLINEAR	**lsdyopts**, options **mxmrts, ctolt, divcp**
*CONTROL_THERMAL_SOLVER	**lsdyopts**, options **atype, ptype, thslvr, cgtol, gpt, eqheat, fwork, sbc**
*CONTROL_THERMAL_TIMESTEP	**lsdyopts**, options **ktst, tipt, itst, tmint, tmaxt, dtempt, tscpt**
*CONTROL_TIMESTEP	**lsdyopts**, options **dtinit, scft, isdo, tslimt, dt2ms, lctm, erode, ms1st, dt2msf, lsnwds**
*DAMPING_GLOBAL	**lsdyopts**, option **gflg**
*DAMPING_PART_MASS	**lsdyopts**, option **mflg**
*DAMPING_PART_STIFFNESS	**lsdyopts**, option **sflg**
*DAMPING_RELATIVE	**lsdyopts**, option **rflg**
*DATABASE_ABSTAT	**lsdyopts**, options **abstat, abstatbn**
*DATABASE_ADAMS	**lsdyopts**, options **iflagadm, m_units, l_units, t_units**
*DATABASE_AVSFLT	**lsdyopts**, options **avsflt, avsfltbn**
*DATABASE_BINARY_D3CRCK	**lsdyopts**, option **d3crck**
*DATABASE_BINARY_D3DRLF	**lsdyopts**, options **d3drlf, d3rdfl**
*DATABASE_BINARY_D3DUMP	**lsdyopts**, option **d3dump**
*DATABASE_BINARY_D3MEAN	**lsdyopts**, option **d3mean**
*DATABASE_BINARY_D3PART	**lsdyopts**, option **d3part**
*DATABASE_BINARY_D3PLOT	**lsdyopts**, option **d3plot**
*DATABASE_BINARY_D3THDT	**lsdyopts**, option **d3thdt**
*DATABASE_BINARY_INTFOR	**lsdyopts**, option **intfor**
*DATABASE_BINARY_RUNRSF	**lsdyopts**, option **runrsf**
*DATABASE_BINARY_XTFILE	**lsdyopts**, option **xtfile**
*DATABASE_BNDOUT	**lsdyopts**, options **bndout, bndoutbn**
*DATABASE_CROSS_SECTION_PLANE	**lsdyopts**, option **cplane**
*DATABASE_CROSS_SECTION_PLANE_ID	**lsdyopts**, option **idhead**
*DATABASE_DEFGEO	**lsdyopts**, options **defgeo, defgeobn**
*DATABASE_DEFORC	**lsdyopts**, options **deforc, deforcbn**
*DATABASE_ELOUT	**lsdyopts**, options **elout, eloutbn**
*DATABASE_EXTENT_AVS	**lsdyopts**, option **lsmpgs**

*DATABASE_EXTENT_BINARY	**lsdyopts**, options **neiph, neips, maxint, strflg, sigflg, epsflg, rltflg, engflg, cmpflg, ieverp, beamip, dcomp, shge, stssz, n3thdt, nintsld**
*DATABASE_EXTENT_MOVIE	**lsdyopts**, option **lsmovie**
*DATABASE_EXTENT_MPGS	**lsdyopts**, option **lsmpgs**
*DATABASE_EXTENT_SSSTAT	**lsdyopts**, option **ssstatex**
*DATABASE_FORMAT	**lsdyopts**, option **iform**
*DATABASE_FSI	**lsdyopts**, option **fsi**
*DATABASE_GCEOUT	**lsdyopts**, options **gceout, gceoutbn**
*DATABASE_GLSTAT	**lsdyopts**, options **glstat, glstatbn**
*DATABASE_H3OUT	**lsdyopts**, options **h3out, h3outbn**
*DATABASE_HISTORY_BEAM	**epb**
*DATABASE_HISTORY_NODE	**npb**
*DATABASE_HISTORY_SHELL	**epb**
*DATABASE_HISTORY_SOLID	**epb**
*DATABASE_HISTORY_TSHELL	**epb**
*DATABASE_JNTFORC	**lsdyopts**, options **jntforc, jntforcb**
*DATABASE_MATSUM	**lsdyopts**, options **matsum, matsumbn**
*DATABASE_MOVIE	**lsdyopts**, options **movie, moviebn**
*DATABASE_MPGS	**lsdyopts**, options **mpgs, mpgsbn**
*DATABASE_NCFORC	**lsdyopts**, options **ncforc, ncforcbn**
*DATABASE_NODFRC	**lsdyopts**, options **nodfor, nodforbn**
*DATABASE_NODAL_FORCE_GROUP	**lsdyopts**, options **nodfnsid, nodfcid, nsid, cid, xtfile**
*DATABASE_NODFOR	**lsdyopts**, options **nodfor, nodforbn**
*DATABASE_NODOUT	**lsdyopts**, options **nodout, nodforbn**
*DATABASE_RBDOUT	**lsdyopts**, options **rbout, rbdout, rbdoutbn**
*DATABASE_RCFORC	**lsdyopts**, options **rcforc, rcforcbn**
*DATABASE_RWFORC	**lsdyopts**, options **rwforc, rwforcbn**
*DATABASE_SBTOUT	**lsdyopts**, options **sbtout, sbtoutbn**
*DATABASE_SECFORC	**lsdyopts**, options **secforc, secforcb**
*DATABASE_SLEOUT	**lsdyopts**, options **sleout, sleoutbn**
*DATABASE_SPCFORC	**lsdyopts**, options **spcforc, spcforcb**
*DATABASE_SPHOUT	**lsdyopts**, options **sphout, sphoutbn**
*DATABASE_SPRING_FORWARD	**lsdyopts**, options **iflagspr, iflag**
*DATABASE_SSSTAT	**lsdyopts**, options **ssstat, ssstatbn**
*DATABASE_SUPERPLASTIC_FORMING	**lsdyopts**, option **superpl**
*DATABASE_SWFORC	**lsdyopts**, options **swforc, swforcbn**
*DATABASE_TPRINT	**lsdyopts**, options **tprint, tprintbn**

*DATABASE_TRACER	**trp**
*DATABASE_TRHIST	**lsdyopts**, options **trhist, trhistbn**
*DEFINE_BOX	**vd**
*DEFINE_BOX_ADAPTIVE	**vd**
*DEFINE_COORDINATE_SYSTEM	**lsys, rigid**
*DEFINE_COORDINATE_VECTOR	**sfb**
*DEFINE_CURVE	**lcd, flcd**
*DEFINE_SD_ORIENTATION	**spring, spdp**
*DEFINE_VECTOR	secondary result of many commands
*ELEMENT_BEAM	**bm, ibm, ibmi, jbm, jbmi, kbm, kbmi**
*ELEMENT_BEAM_THICKNESS	**bm, ibm, ibmi, jbm, jbmi, kbm, kbmi**
*ELEMENT_DISCRETE	**spdp, spring** (see **spd**)
*ELEMENT_MASS	**npm, pm**
*ELEMENT_SHELL_BETA	**block, cylinder, th, ssf, sffi**
*ELEMENT_SHELL_THICKNESS	**block, cylinder, th, ssf, sffi**
*ELEMENT_SOLID	**block, cylinder**
*ELEMENT_TSHELL	**block, cylinder, lsdymats, mate, mt, mti**
*END	automatic
*EOS_LINEAR_POLYNOMIAL	**lsdyeos**
*EOS_JWL	**lsdyeos**
*EOS_JWLB	**lsdyeos**
*EOS_SACK_TUESDAY	**lsdyeos**
*EOS_GRUNEISEN	**lsdyeos**
*EOS_RATIO_OF_POLYNOMIALS	**lsdyeos**
*EOS_LINEAR_POLYNOMIAL_WITH_ENERGY_LEAK	**lsdyeos**
*EOS_IGNITION_AND_GROWTH_OF_REACTION_IN_HE	**lsdyeos**
*EOS_TABULATED_COMPACTION	**lsdyeos**
*EOS_TABULATED	**lsdyeos**
*EOS_PROPELLANT_DEFLAGRATION	**lsdyeos**
*EOS_TENSOR_PORE_COLLAPSE	**lsdyeos**
*HOURGLASS	**lsdymats**
*INITIAL_DETONATION	**detp**
*INITIAL_MOMENTUM	**mdep**
*INITIAL_TEMPERATURE_NODE	**temp, tm, tmi**
*INITIAL_VELOCITY_NODE	**velocity, rotation, ve, vei**
*INTEGRATION_BEAM	**bind, bsd**
*INTEGRATION_SHELL	**sind, lsdymats**

*INTERFACE_COMPONENT_SEGMENT	**iss, issi**
*INTERFACE_SPRINGBACK	**lsdyopts**, option **spbk**
*LOAD_BODY_X	**lsdyopts**, options **xgrav, xgdr**
*LOAD_BODY_Y	**lsdyopts**, options **ygrav, ygdr**
*LOAD_BODY_Z	**lsdyopts**, options **zgrav, zgdr**
*LOAD_BODY_RX	**lsdyopts**, options **xvel, xavcrx**
*LOAD_BODY_RY	**lsdyopts**, options **yvel, xavcrx**
*LOAD_BODY_RZ	**lsdyopts**, options **zvel, xavcrx**
*LOAD_BRODE	**lsdyopts**, options **yldb, hiteb, xb0, yb0, zb0, tb0, lcb1, lcb2, clb, ctb, cpb**
*LOAD_DENSITY_DEPTH	**dymain**
*LOAD_HEAT_GENERATION_SOLID	**vhg, vhgi**
*LOAD_NODE_POINT	**fc, fci, fcs, fcsi, fcc, fcci, ll, ffc, ndl, mom, momi, fmom**
*LOAD_SEGMENT	**pr, pri, pramp, dom, arri, dist**
*LOAD_THERMAL_CONSTANT_NODE	**te, tei**, **temp**
*LOAD_THERMAL_TOPAZ	**lsdyopts**, options **mhk, tpz3d**
*LOAD_THERMAL_VARIABLE_NODE	**tepro**
*MAT_3-PARAMETER_BARLAT	**lsdymats**
*MAT_ACOUSTIC	**lsdymats**
*MAT_ADD_EROSION	**lsdymats**
*MAT_ANISOTROPIC_ELASTIC	**lsdymats**
*MAT_ANISOTROPIC_PLASTIC	**lsdymats**
*MAT_ANISOTROPIC_VISCOPLASTIC	**lsdymats**
*MAT_BAMMAN	**lsdymats**
*MAT_BAMMAN_DAMAGE	**lsdymats**
*MAT_BARLAT_ANISTROPIC_PLASTICITY	**lsdymats**
*MAT_BARLAT_YLD96	**lsdymats**
*MAT_BILKHU/DUBOIS_FOAM	**lsdymats**
*MAT_BLATZ-KO_FOAM	**lsdymats**
*MAT_BLATZ-KO_RUBBER	**lsdymats**
*MAT_BRITTLE_DAMAGE	**lsdymats**
*MAT_CABLE_DISCRETE_BEAM	**lsdymats**
*MAT_CELLULAR_RUBBER	**lsdymats**
*MAT_CLOSED_CELL_FOAM	**lsdymats**
*MAT_COMPOSITE_DAMAGE	**lsdymats**
*MAT_COMPOSITE_FAILURE_MODEL	**lsdymats**
*MAT_COMPOSITE_FAILURE_SHELL_MODEL	**lsdymats**
*MAT_COMPOSITE_FAILURE_SOLID_MODEL	**lsdymats**
*MAT_CONCRETE_DAMAGE	**lsdymats**
*MAT_CRUSHABLE_FOAM	**lsdymats**

*MAT_DAMPER_NONLINEAR_VISCOUS	**lsdymats**
*MAT_DAMPER_VISCOUS	**lsdymats**
*MAT_ELASTIC	**lsdymats**
*MAT_ELASTIC_FLUID	**lsdymats**
*MAT_ELASTIC_PLASTIC_HYDRO	**lsdymats**
*MAT_ELASTIC_PLASTIC_HYDRO_SPALL	**lsdymats**
*MAT_ELASTIC_PLASTIC_THERMAL	**lsdymats**
*MAT_ELASTIC_WITH_VISCOSITY	**lsdymats**
*MAT_ENHANCED_COMPOSITE_DAMAGE	**lsdymats**
*MAT_FABRIC	**lsdymats**
*MAT_FLD_TRANSVERSELY_ANISOTROPIC	**lsdymats**
*MAT_FORCE_LIMITED	**lsdymats**
*MAT_FRAZER_NASH_RUBBER_MODEL	**lsdymats**
*MAT_FU_CHANG_FOAM	**lsdymats**
*MAT_GENERAL_VISCOELASTIC	**lsdymats**
*MAT_GEOLOGIC_CAP_MODEL	**lsdymats**
*MAT_HIGH_EXPLOSIVE_BURN	**lsdymats**
*MAT_HONEYCOMB	**lsdymats**
*MAT_HYDRAULIC_GAS_DAMPER_DISCRETE_BEAM	**lsdymats**
*MAT_HYPERELASTIC_RUBBER	**lsdymats**
*MAT_HYSTERETIC_SOIL	**lsdymats**
*MAT_ISOTROPIC_ELASTIC_FAILURE	**lsdymats**
*MAT_ISOTROPIC_ELASTIC_PLASTIC	**lsdymats**
*MAT_JOHNSON_COOK	**lsdymats**
*MAT_KELVIN-MAXWELL_VISCOELASTIC	**lsdymats**
*MAT_LAMINATED_GLASS	**lsdymats**
*MAT_LINEAR_ELASTIC_DISCRETE_BEAM	**lsdymats**
*MAT_LOW_DENSITY_FOAM	**lsdymats**
*MAT_LOW_DENSITY_VISCOUS_FOAM	**lsdymats**
*MAT_MODIFIED_HONEYCOMB	**lsdymats**
*MAT_MODIFIED_ZERILLI_ARMSTRONG	**lsdymats**
*MAT_MOONEY-RIVLIN_RUBBER	**lsdymats**
*MAT_MTS	**lsdymats**
*MAT_NONLINEAR_ELASTIC_DISCRETE_BEAM	**lsdymats**
*MAT_NONLINEAR_ORTHOTROPIC	**lsdymats**
*MAT_NONLINEAR_PLASTIC_DISCRETE_BEAM	**lsdymats**
*MAT_NULL	**lsdymats**
*MAT_OGDEN_RUBBER	**lsdymats**
*MAT_ORIENTED_CRACK	**lsdymats**
*MAT_ORTHOTROPIC_ELASTIC	**lsdymats**
*MAT_ORTHOTROPIC_THERMAL	**lsdymats**

*MAT_ORTHOTROPIC_VISCOELASTIC	**lsdymats**
*MAT_PIECEWISE_LINEAR_PLASTICITY	**lsdymats**
*MAT_PLASTICITY_WITH_DAMAGE	**lsdymats**
*MAT_PLASTIC_GREEN-NAGHDI_RATE	**lsdymats**
*MAT_PLASTIC_KINEMATIC	**lsdymats**
*MAT_POWER_LAW_PLASTICITY	**lsdymats**
*MAT_PSEUDO_TENSOR	**lsdymats**
*MAT_RAMBERG_OSGOOD	**lsdymats**
*MAT_RATE_SENSITIVE_POWERLAW_PLASTICITY	**lsdymats**
*MAT_RESULTANT_PLASTICITY	**lsdymats**
*MAT_RIGID	**lsdymats**
*MAT_SHAPE_MEMORY	**lsdymats**
*MAT_SID_DAMPER_DISCRETE_BEAM	**lsdymats**
*MAT_SOIL_AND_FOAM	**lsdymats**
*MAT_SOIL_AND_FOAM_FAILURE	**lsdymats**
*MAT_SOIL_CONCRETE	**lsdymats**
*MAT_SPOTWELD	**lsdymats**
*MAT_SPRING_ELASTIC	**lsdymats**
*MAT_SPRING_ELASTOPLASTIC	**lsdymats**
*MAT_SPRING_GENERAL_NONLINEAR	**lsdymats**
*MAT_SPRING_INELASTIC	**lsdymats**
*MAT_SPRING_MAXWELL	**lsdymats**
*MAT_SPRING_MUSCLE	**lsdymats**
*MAT_SPRING_NONLINEAR_ELASTIC	**lsdymats**
*MAT_STEINBERG	**lsdymats**
*MAT_STEINBERG_LUND	**lsdymats**
*MAT_STRAIN_RATE_DEPENDENT_PLASTICITY	**lsdymats**
*MAT_TEMPERATURE_DEPENDENT_ORTHOTROPIC	**lsdymats**
*MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC	**lsdymats**
*MAT_USER_DEFINED_MATERIAL_MODELS	**lsdymats**
*MAT_VISCOELASTIC	**lsdymats**
*MAT_VISCOLASTIC_FABRIC	**lsdymats**
*MAT_VISCOUS_FOAM	**lsdymats**
*MAT_THERMAL_ISOTROPIC	**lsdythmt**
*MAT_THERMAL_ISOTROPIC_PHASE_CHANGE	**lsdythmt**
*MAT_THERMAL_ISOTROPIC_TD	**lsdythmt**
*MAT_THERMAL_ISOTROPIC_TD_LC	**lsdythmt**
*MAT_THERMAL_ORTHOTROPIC	**lsdythmt**
*MAT_THERMAL_ORTHOTROPIC_TD	**lsdythmt**

*NODE	**block, cylinder, bm, jt, npm, b, bi, plane**
*PART	**lsdymats**
*RIGIDWALL_PLANAR	**plane, sw, swi**
*SECTION_BEAM	**lsdymats**
*SECTION_DISCRETE	**spd**
*SECTION_SHELL	**lsdymats**
*SECTION_SOLID	**lsdymats**
*SECTION_SOLID_ALE	**lsdymats**
*SECTION_TSHELL	**lsdymats**
*SET_BEAM	**eset, eseti**
*SET_NODE_COLUMN	**sid, si, sii**
*SET_NODE_LIST	**nset, nseti**
*SET_PART_LIST	**lsdyopts** (gravity stress initialization), **sid** (contact between parts), **shtoso**, **shtosoi**, **dymain**
*SET_SEGMENT	**fset, fseti, plane, syf, syfi, iss, issi**
*SET_SHELL_LIST	**eset, esti**
*SET_SOLID	**eset, esti**
*SET_TSHELL	**eset, esti**
*TITLE	**title**

You may want to view some of the properties graphically using the **condition** (**co**) or **mlabs** command in the merge phase. Below is a table of those properties generated for LS-DYNA that can be annotated using these commands.

dx	nodes with a fixed translation in the x-direction
dy	nodes with a fixed translation in the y-direction
dz	nodes with a fixed translation in the z-direction
rx	nodes with a fixed rotation about their x-axis
ry	nodes with a fixed rotation about their y-axis
rz	nodes with a fixed rotation about their z-axis
mom	the nodes with prescribed torques
fc	point load vectors at nodes
fd	forced displacement vectors at nodes
pr	pressure surface amplitude vectors
sy	nodes assigned to a numbered symmetry plane
si	element faces or nodes on sliding interface or contact surface
rb	boundary radiation orientation vectors on faces
fl	boundary flux orientation vectors on faces
cv	boundary convection orientation vectors on faces
tm	initial temperature at nodes

ft boundary temperature at nodes
fv boundary velocity vectors at nodes
sw nodes assigned to a numbered stone wall
nr non-reflecting boundary faces
jt identify nodes within a numbered joint
iss interface save segments
vhg volumetric heat generation
or element local coordinate system axis
syf nodes assigned to a numbered symmetry plane with failure
acc the acceleration vectors
n shell element outward normal vectors
sp a numbered set of springs/dampers
pm all point masses
npb nodes in the nodal print block
epb elements in the element print blocks
thic shell thicknesses
mdep elements with momentum deposition
sfb local coordinate system
tepro temperature profile nodes by load curve
sc ALE smoothing constraint directions
spwf material 100 spot welds
frb fixed rotational boundary displacements
ffc follower forces
fmom follower moments
detp detonation points
trp tracer particles

The **labels** (**la**) command will label geometric objects in the physical display window. In particular, node, element, and element face sets can be annotated in the merge phase. You can also see the components of these sets in the merge phase by picking the ***Pick*** button and then the ***Sets*** button found in the Environment window.

The **tmm** command can be used to calculate the mass of each part. Be sure to merge the nodes using one of the merging commands such as **stp** and, finally, use the **lsdyna keyword** command to select LS-DYNA as the output option and the **write** command to actually create the input deck for LS-DYNA.

The file produced by **True*Grid*®** is an ASCII file that can be examined or modified using any text editor. Some experienced users always inspect the file and modify it instead to rerunning **True*Grid*®** when make easy changes to the model such as changing a material model parameter or the time step. For this reason, the output file has helpful comments. However, this file can be very large and it might be easier to modify the **True*Grid*®** session file and rerun **True*Grid*®**. One of the comments

that is automatically written has a time stamp for archiving purposes. The **title** is also helpful for archiving. You can insert your own comments with the use of the **comment** command.

The **readmesh** command has a **lsdyna** option so that you can import the LS-DYNA (keyword format only) elements into **True*Grid*®**.This is intended to be used to translate a LS-DYNA mesh (nodes and elements only) into another format. This feature does not replace the session file because the block structure of the mesh cannot be reconstructed. This also means there are no block boundary interfaces (**bb** command) to utilize. If there is a block structure underlying the mesh, you can form a block boundary interface using the **mbb** command, but this can be tedious.

Sliding (or Contact) Surfaces

To form a contact surface, use the **sid** command to define the surface type. Some types have only one side. Some are formed from faces of bricks or shells. Others are formed partially from nodes. Alternatively, you can select materials (equivalent to LS-DYNA parts) to identify the LS-DYNA parts that form either or both sides of an automatic contact. Be sure to choose a LS-DYNA type contact surface, since other types will not be recognized when **True*Grid*®** writes the output file.

While in the part phase use the **si** or **sii** commands to select faces of that part for inclusion in the surface definition. If the face is from a shell element, be sure to use the **orpt** orientation command prior to issuing the **si** or **sii** command so that the orientation of the face is towards the opposing face in the sliding surface. If you are using part replication (**lrep**, **grep**, or **pslv**), then you may want to use the **lsii** or the **gsii** to increment the sliding interface command for each replication. You must use the **sid** command for each sliding surface that is referenced when the **lsii** or **gsii** commands are used with replication.

You can use sets in the merge phase to add faces or nodes to a sliding surface. These sets can be formed with the combined use of the **fset** (for faces) and the **nset** (for nodes) commands in the part and merge phase. Only use node sets when defining a sliding surface where nodes are on the slave side and otherwise only use face sets. The node density between the master and slave sides of the interface should be roughly equal. When forming the mesh in the part phase, it may be necessary to build into the mesh a small gap between the master and slave sides of the contact surfaces, depending on the mesh density and the curvature to avoid initial penetration of the slave side into the master side.

When you merge the nodes (in the merge phase), the nodes from the slave side will not be allowed to merge with the nodes on the master side. Use the **mns** command in the merge phase to override this condition. When you fist merge the nodes, a table will be printed to the text window and the tsave file listing the number of faces and nodes associated with each sliding surface. Check this table carefully. You can also see the faces and nodes of either side of the sliding surfaces using the **co** command. When using this in combination with the hide graphics option, you can see the orientation of the faces. Use **labels** command to show how the nodes have merged graphically.

You can also assign materials (LS-DYNA parts) to both the slave and master sides instead of selecting faces (segments) or nodes. Care is needed to be sure that nodes between the master and slave materials (LS-DYNA parts) are not merged. Use the **bptol** command to avoid merging nodes between (**True*Grid*®**) parts using a tolerance of -1. You can also use the **sid** command with the **dummy** type along with the **si** or **sii** command to avoid merging nodes. Use the **vd** command to define boxes used to limit the region of a contact surface.

Initial and Boundary Conditions

There are several ways to constrain nodes. The **b** and **bi** commands in the part phase or the **b** command in the merge phase will constrain nodes in the global coordinate system. Use the **plane** command to specify symmetry plane constraints including symmetry planes with failure. Nodes in the model will be assigned to these symmetry planes based on the tolerance you specify in the **plane** command. The **lb** (and the associated **lsys**) command can be used to set the constraints in any coordinate system. The **sfb** command can also be used to do this. Be sure that something in the model has been constrained or the entire model might fly off.

To set non-reflective (or transmitting) boundary conditions, use the **nr** and **nri** commands in the part phase or the **nr** command in the merge phase. Special care is needed when developing a model using this type of boundary condition. See the LS-DYNA User Manual for details.

If you use the **velocity** or **rotation** command in the control phase, then all subsequent parts will be assigned this initial velocity. This can be over ridden using the **velocity** or **rotation** command within a part. Both of these conditions can be over ridden for specific regions of the mesh using the **ve** or **vei** commands in the part phase or the **ve** command in the merge phase. Velocities are not accumulative. Care is needed when assigning initial velocities so that when two nodes are merged, the velocities of those two nodes match. Only one of the velocities will be used and if they do not match, you may get an unexpect result. Usually, if the velocities of two merged nodes do not match, this indicates an error in the model. **True*Grid*®** does not protect you from or identify these incompatibilities.

Condition commands in the Part Phase

b	nodal constraints
bi	nodal constraints
sc	ALE Smoothing Constraint
nr	Non-Reflecting Boundary
nri	Non-Reflecting Boundary
frb	Fixed/Prescribed Rotational Boundary Condition
frbi	Fixed/Prescribed Rotational Boundary Condition
rb	Thermal Radiation Boundary Condition
rbi	Thermal Radiation Boundary Condition

vrb	Variable Thermal Radiation Boundary Condition
vrbi	Variable Thermal Radiation Boundary Condition
sfb	Surface Nodes Local Boundary Conditions
sfbi	Surface Nodes Local Boundary Conditions
plane	Symmetry, symmetry with failure, or stone wall plane
mdep	Initial Momentum Deposition

Condition commands in the Merge Phase

b	nodal constraints on a node set
nr	Non-Reflecting Boundary on a Face Set
frb	Fixed/Prescribed Rotational Boundary Condition
rb	Thermal Radiation Boundary Condition
plane	Symmetry, symmetry with failure, or stone wall plane
rigid	Define a Rigid Body from a Set of Node

The **detp** command is a global command used to select initial detonation points.

Loads

There are numerous ways to assign loads. The list of commands that can be used to assign loads in the part phase includes:

fc	Cartesian concentrated nodal loads
fci	Cartesian concentrated nodal loads
fcc	cylindrical concentrated nodal loads
fcci	cylindrical concentrated nodal loads
fcs	spherical concentrated nodal loads
fcsi	spherical concentrated nodal loads
mom	nodal moment about one of the nodal axis in the global coordinate system
momi	nodal moment about one of the nodal axis in the global coordinate system
ndl	pressure converted to distributed nodal loads
ndli	pressure converted to distributed nodal loads
pr	pressure loads on element faces
pri	pressure loads on element faces
pramp	pressure loads on element faces
fv	Cartesian prescribed nodal velocities
fvi	Cartesian prescribed nodal velocities
fvc	cylindrical prescribed nodal velocities
fvci	cylindrical prescribed nodal velocities
fvs	spherical prescribed nodal velocities
fvsi	spherical prescribed nodal velocities

fvv	Cartesian variable prescribed nodal velocities
fvvi	Cartesian variable prescribed nodal velocities
fvvc	cylindrical variable prescribed nodal velocities
fvvci	cylindrical variable prescribed nodal velocities
fvvs	spherical variable prescribed nodal velocities
fvvsi	spherical variable prescribed nodal velocities
acc	Cartesian prescribed nodal acceleration
acci	Cartesian prescribed nodal acceleration
accc	cylindrical prescribed nodal acceleration
accci	cylindrical prescribed nodal acceleration
accs	spherical prescribed nodal acceleration
accsi	spherical prescribed nodal acceleration
vacc	Cartesian variable prescribed nodal acceleration
vacci	Cartesian variable prescribed nodal acceleration
vaccc	cylindrical variable prescribed nodal acceleration
vaccci	cylindrical variable prescribed nodal acceleration
vaccs	spherical variable prescribed nodal acceleration
vaccsi	spherical variable prescribed nodal acceleration
fd	Cartesian displacement
fdi	Cartesian displacement
fdc	cylindrical displacement
fdci	cylindrical displacement
fds	spherical displacement
fdsi	spherical displacement
frb	prescribed rotation
frbi	prescribed rotation
cv	Boundary Convection
cvi	Boundary Convection
vcv	Variable Boundary Convection
vcvi	Variable Boundary Convection
fl	Boundary Flux
fli	Boundary Flux
vfl	Variable Boundary Flux
vfli	Variable Boundary Flux
ft	Boundary Temperature
fti	Boundary Temperature
vft	Variable Boundary w/ Functional Temperature
vfti	Variable Boundary w/ Functional Temperature
vhg	Volumetric Heat Generation
vhgi	Volumetric Heat Generation
vvhg	Variable Volumetric Heat Generation
ll	Linearly Interpolate Loads by Arc Length

The list of commands that can be used to assign loads in the merge phase includes:

fc	Cartesian concentrated nodal loads
mom	nodal moment about one of the nodal axis in the global coordinate system
ndl	pressure converted to distributed nodal loads
pr	pressure loads on element faces
pramp	pressure loads on element faces
fv	Cartesian prescribed nodal velocities
fvv	Cartesian variable prescribed nodal velocities
vacc	Cartesian variable prescribed nodal acceleration
fd	Cartesian displacement
frb	prescribed rotation
ffc	concentrated nodal load with a follower force
fmom	nodal moment with a follower force
cv	Boundary Convection on a Face Set
ft	Boundary Temperature by Load Curve
vvhg	Variable Volumetric Heat Generation
ffc	Follower Nodal Force by Load Curve
fmom	Follower Nodal Moment about an Axis

The **pramp** command is used with either **pr** or **pri**. It applies a pressure based on a function for all nodes that have a zero pressure. In most cases, the magnitude of the load is specified using a load curve. This varies the amplitude of the load with respect to time. The **arri** and associated **dist** commands can be used to form the arrival time of pressure in the **pr**, **pri**, **fl**, and **fli** commands.

Load Curves

Load curves are 2D polygonal curves that can be created using the **lcd** and **flcd** commands. Load curves are typically used to define the relative amplitude of a load with respect to time. They can be used to relate any two variables. Almost all prescribed loads require a load curve in time so that the amplitude of the load can vary. It is best to define a load curve before it is referenced in a load or material model to avoid a warning message. When the output file is written, if a load curve is referenced but not defined, you will also receive a warning message. Then a simple load curve will be used in the output file so that a valid LS-DYNA file is written. It is advised that you correct this by defining the appropriate load curve for the problem. Do not rely on the load curve that is automatically generated.

In some dialogue boxes you might be prompted for a load curve or a set id. This is because such commands can be used to define, for example, a dynamic load for LS-DYNA or a static load for another output option that has the option to turn loads on or off depending on the set id. Simply ignore the set id portion of the prompt and supply the load curve number.

Stone Walls

A stone wall is defined with two commands. Use the **plane** command to set the properties of the stone wall. Nodes to react to the stone wall will not be selected automatically based on the tolerance. Use the **sw** and **swi** commands in the part phase to assign faces of the model to react to the stone wall. You can use the **sw** command in the merge phase as well to assign faces from a face set to react to the stone wall.

Bricks

Brick elements refer to hexahedral, prism (wedge), and tetrahedral elements and are considered the same type in LS-DYNA. Only one element type can associated with a material definition. If you want two different element types with the same material properties, you must define two materials. Most, but not all, materials support the different brick element types. All the different element formulations for brick elements are assigned in the **lsdymats** command.. Be sure to use the **mate**, **mt**, or **mti** command to assign the proper material to each section of the mesh.

The element local coordinate system used in an orthotropic or anisotropic material is imposed by the order of the nodes that define the element. You can flip the nodal ordering to switch the orientation of this local coordinate system using the **or** command in the part phase.

Shells

Shell elements refer to both quadrilateral and triangular elements and sometimes referred to as structural elements. All the different element formulations for shell elements are assigned in the **lsdymats** command. Be sure to use the **mate**, **mt**, or **mti** command to assign the proper material to each section of the mesh. The default shell thicknesses are included as part of the cross sectional properties. These default thicknesses can be over ridden with the use of the **thic** command in the part phase. Both can be over ridden for a region of the part using the **th** and **thi** commands. If you have two surfaces that represent the inner and outer surfaces of a structure that is to be modeled using shell elements, than you can use the **ssf** and **ssfi** commands in the part phase to create shells with variable thickness.

The orientation of the positive normal direction to the shell is dictated by the nodal ordering of the nodes that define the shell. This positive direction is used, for example, to determine the direction of a positive pressure. This direction can be flipped using the **n** command in the part phase. The order of the nodes also dictate the local material coordinate system which can be important when using an orthotropic or anisotropic material. Use the **or** command to flip the coordinate system to the desired direction. When an angle is specified for the orientation of a composite material, it is with respect to this orientation.

You may need to specify the through thickness integration points when defining a composite material. This can be done by defining an integration rule with the **sind** command. Then identify this rule in the material definition.

Thick Shells

Thick shells are generated and look like hexahedral elements. Only a few materials support thick shells. Be sure to use the **mate**, **mt**, or **mti** command to assign the proper material to each section of the mesh.

The element local coordinate system used in an orthotropic or anisotropic material is imposed by the order of the nodes that define the element. You can flip the nodal ordering to switch the orientation of this local coordinate system using the **or** command in the part phase.

You may need to specify the through thickness integration points when defining a composite material. This can be done by defining an integration rule with the **sind** command. Then identify this rule in the material definition. All other cross section information is specified in the material definition.

Beams

Two nodes are required to form a beam element. In many cases, a third node is needed to define the local coordinate system used to form the cross sectional properties. These element are sometimes referred to as structural elements. Use the **ibm**, **ibmi**, **jbm**, **jbmi**, **kbm**, and **kbmi** commands to form beam elements with shell or brick structures while in the part phase. If the material of the shell or brick structure is set to zero using the **mt**, **mti**, or **mate** command, then the shells or bricks will be ignored, but the embedded beams will not be ignored. This is a convenient way to build an array of beams using block structured methods. You can also use the **bm** command in the merge phase to build a string of beams that can be made to follow a 3D curve. The **beam** command (this command has been denigrated) can also be used to form beam elements, but the command is not interactive.

Both the element type and the default cross section properties are defined in the material definition. You can also use the **bsd** command to define cross sectional properties to over ride the material default cross sectional properties. When you create a beam, refer to the **bsd** number to assign these cross sectional properties to the beam. Use the **bind** command to define beam integration rule, if needed. Then refer to this integration rule when defining the material.

Joints and Rigid Bodies

A rigid body is formed using shells and bricks that are assigned the rigid body material. Each rigid body can be attached to other parts of the model using joints. A joint is defined in two steps. The **jd** command is used to define the properties of a joint. Then the **jt** command is used to identify which

nodes are used to form the joint. Nodes within a joint are not merged. When forming a rigid body, define a material with the **lsdymats** command and assign this material to a region of the mesh. Use the **rigbm** to merge two rigid body materials to act as one rigid body.

Springs, Dampers and Point Masses

Springs and dampers are treated the same in **True*Grid*®**. They are only distinguishable by the material properties assigned them. Use the **spd** command to define the properties of the spring or damper. Then use the **spring** command to assign nodes to a numbered spring. Alternatively, the **spdp** command can be used in the part phase to create an array of springs between two parts, analogous to a contact surface.

Point masses can be generated in the part or merge phase. There are two types of point masses. The **pm** command will assign a mass to an existing node. The **npm** will create a new node and assign it a mass. The latter must then be connected either to a spring or beam.

Temperatures

Use the **tepro** command in the part or merge phase to apply a thermal variable node load. You can use the **temp** command to set a default constant temperature. Then use the **te** and **tei** commands in the part phase or the **te** command in the merge phase to change the constant temperature in different regions of the mesh.

Shared Constraints

Use the **mpc** command to couple a set of nodes. This requires that you create a node set first. The **nset** or **nseti** command can be used in the part phase and the **nset** command in the merge phase to create a node set. Also, click on the pick button in the environment window during the merge phase. Then you can use the mouse to modify or create a node set. The nodes sharing a set of constraints will not be merged together.

Use the **shtoso** and **shtosoi** commands to form the constraints supporting the shell-to-solid shared constraints in LS-DYNA.

Spotwelds

The ***Spotweld*** window, found in the boundary menu, is an easy and interactive way to create multiple spot welds using the mouse to click on the nodes. It is only available in the merge phase. It creates **spw** commands and writes them to the tsave (session) file. The **spw** creates a single spotweld constraint. The **spwd** command defines the properties of the spotweld in the **spw**

command. The **spwf** command is designed to create the unique spotweld feature using material type 100.

Tied Nodes with Failure

The **fn** and **fni** commands in the part phase will generate a shell or brick mesh where every element in the region has unique nodes. The coincident nodes are automatically tied together with a failure criteria. The nodes that are tied together are not merged together in the merge phase.

Sets

Node, face (segment), and element sets can be formed in several ways. The **nset** and **nseti** can be used to form node sets in the part phase. The **nset** command in the merge phase can be used to form a set of nodes by enumeration. Alternatively, nodes can be selected by proximity to geometry or from existing conditions. Boolean operations can be performed on these sets. A similar array of functions are also available for face sets (**fset** and **fseti**) and for element sets (**eset** and **eseti**). These sets will always be written to the LS-DYNA output file unless they are deleted using the **delset** command. Since conditions, contact surfaces, and loads can be applied to a set, it may be redundant to have both the property applied to a set and to have the set written to the LS-DYNA output file. This is when it is appropriate to use the **delset** command, after the property has been applied to the set.

You can name sets in **True*Grid*®** and when they are written to the LS-DYNA output file, they are assigned a number. If you name a set with a number, it will appear in the LS-DYNA output file by that number. This feature is critical when referring to the set in a **verbatim** command.

Smooth Particle Hydro (SPH)

The **sparticle** command is used before the **cylinder** or **block** command to transform any brick elements into Smooth Particle Hydrodynamic elements. To continue using linear bricks, issue the **linear** command. When defining the material for these elements, choose the SPH element type. Be sure to specify the material density when defining this material. It is needed to complete the SPH elements. Use the **plane** command to specify the SPH symmetry plane. The **vd** command has an option to form a SPH box. The **lsdyopts** has a number of SPH controlling parameters. Use the **etd** command to control the element types being drawn in the picture. The **sph** option can activate or deactivate graphics for SPH elements. The **co** or **condition** command with the **sph** option labels the SPH element with their numbers.

Data for Post Processing

There are a number of options of the **lsdyopts** command to control the data saved in the database by LS-DYNA for post processing. Use the **trp** command to define tracer points. Use the **iss** or **issi** command to identify cross section interfaces that LS-DYNA will save in a file.

You may wish to analyze in greater detail the evolution of certain nodes or elements. Use the **npb** and **epb** commands (referred to as time history blocks), respectively, to identify areas of the mesh requiring a more detailed accumulation of data by LS-DYNA.

Unsupported Features in LS-DYNA

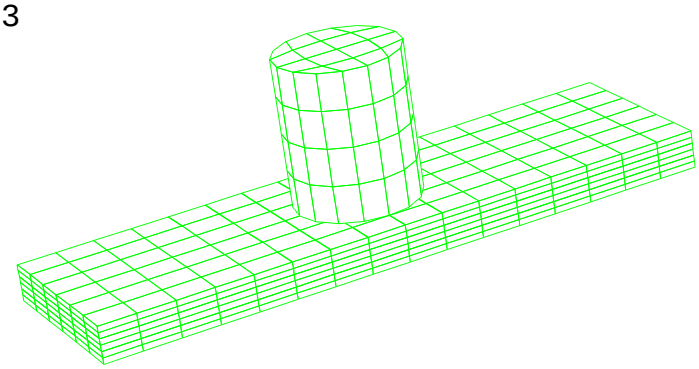
The **verbatim** command can be used for a substitute if **True*Grid*®** does not support a feature in LS-DYNA.

II. LS-DYNA Output Examples

The following examples were provided by Mike Burger.

Example 1 impacting rod

```
lsdyna keyword
lsdyopts endtim .01 d3plot dtcycl .001 ; ;
lsdymats 1 3 rho .001 e 1.e7 pr .3 sigy 40000 etan 100 ;
lsdymats 2 3 rho .001 e 1.e7 pr .3 sigy 40000 etan 100 ;
sid 1 lsdsi 24 slvmat 1;mstmat 2;atbo 2 5000 5000 ;
sid 2 lsdsi 24 slvmat 2;mstmat 3;atbo 2 5000 5000 ;
sid 3 lsdsi 24 slvmat 3;mstmat 4;atbo 2 5000 5000 ;
partmode i
c
lcd 1 0 0 .01 500;
sid 1 lsdsi 24 slvmat 1;mstmat 2;atbo 2 10000 5000 ;
sd 1 cy 5 1.5 0 0 0 1 1
c ****************** part 1 beam
block 16; 6; 2 1 2 1 2; 0 10 0 3 0 .2 .2  .4 .4 .6
dei 1 2; 1 2; 2 3 0 4 5;
mate 1 mti ;;3 4; 2 mti ;;5 6; 3
bi -1 0 -2;;;  dx 1 dz 1;
endpart
c ********************* part 2
block 4; 4; 4; 4 6 .5 2.5 .6 3
sfi -1 -2; -1 -2; 1 2;sd 1
fci 1 2;1 2;-2;1 1 0 0 1 ;
mate 4
endpart
merge write
```

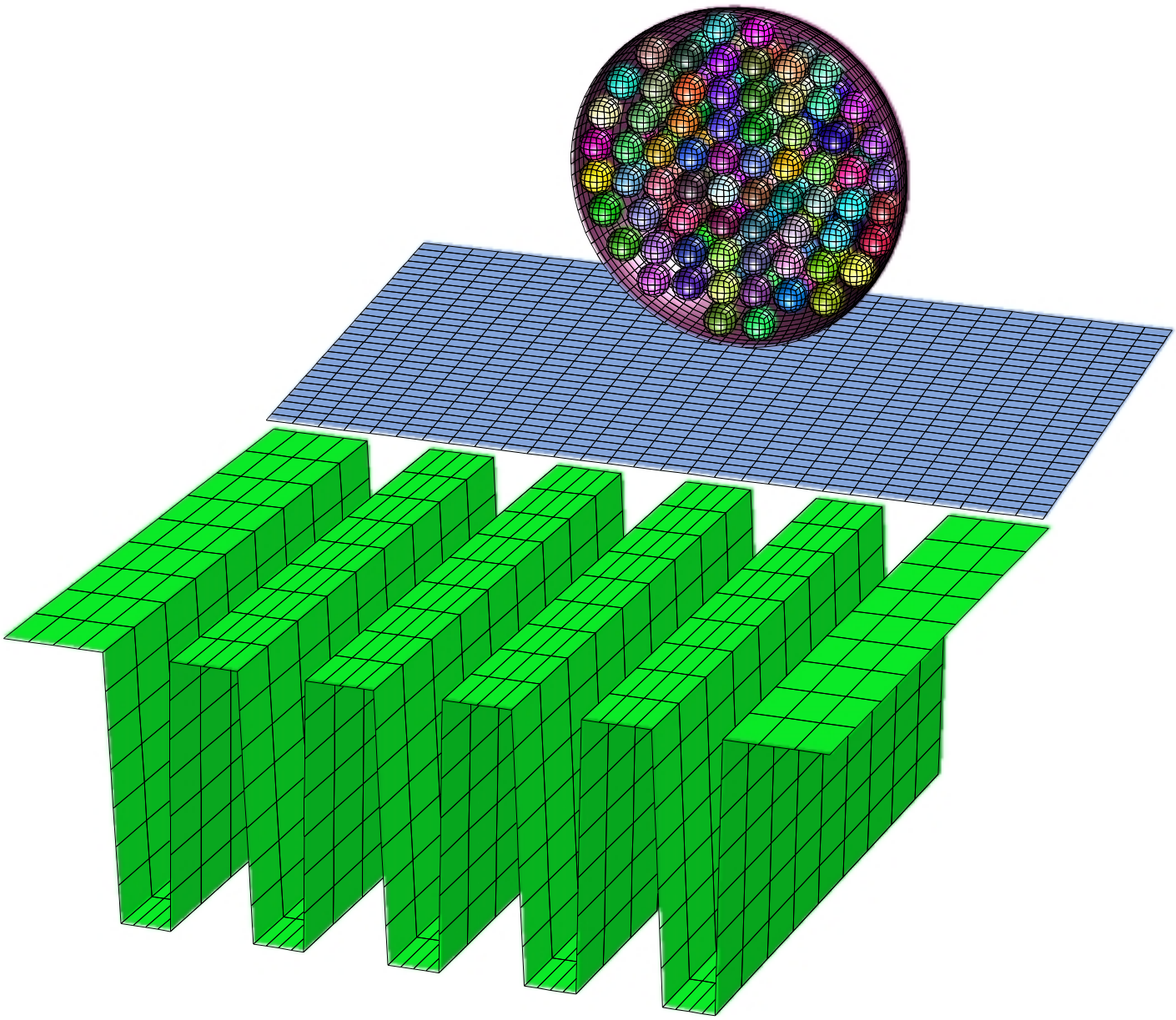


Example 2 cascading balls

```
lsdyna keyword
lcd 1 0 1 0.025 1 0.030 1 1 1;
lsdyopts endtim .2 d3plot dtcycl .0006 ;
         ygrav 10800 1 zgrav 3800 1 scft .75 nsbcs 5;;
para matn 1;
while(%matn.lt.153)
 lsdymats %matn 1 struct rho .012 e 4.e5 pr .3;
 para matn [%matn+1];
endwhile
lsdymats 200 20 shell elfor bt shth .2 rho .01 e 1.e6
                pr .3 cmo con 7 7 ;
```

```
lsdymats 201 20 shell elfor bt shth .01 rho .01 e 1.e6
                pr .3 cmo con 7 7 ;
verbatim
*CONTACT_automatic_SINGLE_SURFACE_TITLE
1,TrueGrid Sliding Interface #          1
0,0,0,0,,,0,0
0.,0.,0.,0.,10.,0,0.,0.
0.,0.,0.,0.,0.,0.,0.,0.
2
endverbatim
sd  1 sp  0 0 0 1
partmode i
c ****************** part 1 ***************
block 4;4;4; -.2 .2 -.2 .2 -.2 .2
sfi -1 -2; -1 -2; -1 -2;sd 1
relaxi 1 2;;; 20 0 1
lmi 1 gmi 17 mate 1
lct 16 mx -2; repe 4;
       my 2 ; my 2 mx -2.5; my 2 mx -5.0; my 2 mx -7.5;
       my 4 ; my 4 mx -2.1;my 4 mx -4.2;my 4 mx -6.3;
       my 6 ; my 6 mx -2.2; my 6 mx -4.4;
       my 8 mx -.2 ;
gct 4  my 1 mx -1;rxz my -1 mx -1;my 1 ryz mx 1;rxz my -1 ryz mx 1;
lrep 0:16; grep 1:4;
endpart
c ****************** part 2 ***************
block 4;4;4; -.2 .2 -.2 .2 -.2 .2
sfi -1 -2; -1 -2; -1 -2;sd 1
relaxi 1 2;;; 20 0 1
lmi 1 gmi 12 mate 69
lct 11 mx -2; repe 3;my 2 ; my 2 mx -2.5; my 2 mx -5.0;
       my 4 ; my 4 mx -2.1;my 4 mx -4.2;my 6 ; my 6 mx -2.2;
gct 4  my 1 mx -1 mz 2 ; rxz my -1 mx -1 mz 2;
       my 1 ryz mx 1 mz 2; rxz my -1 ryz mx 1 mz 2;
lrep 0:11; grep 1:4;
endpart
c ****************** part 3 ***************
block 4;4;4; -.2 .2 -.2 .2 -.2 .2
sfi -1 -2; -1 -2; -1 -2;sd 1
relaxi 1 2;;; 20 0 1
lmi 1 gmi 6 mate 117
lct 5 mx -2; repe 2; my 2 ; my 2 mx -2.5; my 4 ;
gct 4  my 1 mx -1 mz 4 ; rxz my -1 mx -1 mz 4;
       my 1 ryz mx 1 mz 4; rxz my -1 ryz mx 1 mz 4;
lrep 0:5; grep 1:4;
endpart
c ****************** part 4 ***************
```

```
block 4;4;4; -.2 .2 -.2 .2 -.2 .2
sfi -1 -2; -1 -2; -1 -2;sd 1
relaxi 1 2;;; 20 0 1
lmi 1 gmi 3 mate 141
lct 2 mx -2 ; my 2;
gct 4  my 1 mx -1 mz 6 ; rxz my -1 mx -1 mz 6;
       my 1 ryz mx 1 mz 6; rxz my -1 ryz mx 1 mz 6;
lrep 0:2; grep 1:4;
endpart
c ****************** part 5 ***************
partmode s
cylinder -1; 1 90; 1 30; 10 0 360 0 9
sd 2 sp 0 0 0 10.2
mbi -1; 1 2; -2; x -9.50866
sfi -1;;;sd 2
mbi -1; 1 2; -2; x -0.429586
mbi -1; 1 2; -2; x -0.522362e-01
mbi -1; 1 2; -2; x -0.728810e-02
mate 200
lct 1 mz -1; lrep 1;
endpart
c ****************** part 6 ***************
block 1 30; -1; 1 30; -24 24 -11.  -15 9
tri 1 2; -1;; v 0.841084e-07 12.0562 0.975823
              tf rt -0.841084e-07 -12.0562 -0.975823
                 rt 1.00000 -12.0562 -0.975823
                 rt -0.775911e-07 -11.0674 -1.12492;
mate 201
endpart
c ****************** part 7 ***************
block 1 -4 -8 -12 -16 -20 -24 -28 -32 -36 -40 44; -1 -8;
1 10; -24 -18 -14 -10 -6 -2 2 6 10 14 18 24 -30 -14 -50 -14
dei 2 3 0 4 5 0 6 7 0 8 9 0 10 11; -2;;
dei 1 2 0 3 4 0 5 6 0 7 8 0 9 10 0 11 12; -1;;
mbi -2; -1; 1 2; x .5 mbi -3; -1; 1 2; x -.5
mbi -4; -1; 1 2; x .5 mbi -5; -1; 1 2; x -.5
mbi -6; -1; 1 2; x .5 mbi -7; -1; 1 2; x -.5
mbi -8; -1; 1 2; x .5 mbi -9; -1; 1 2; x -.5
mbi -10; -1; 1 2; x .5 mbi -11; -1; 1 2; x -.5
mate 201
endpart
merge
bptol 1 2 0 bptol 2 3 0 bptol 3 4 0 bptol 1 1 0
bptol 2 2 0 bptol 3 3 0 bptol 4 4 0
stp .001 write
```



III. LS-DYNA Reference

The commands found here are provided to the user so that a complete input file can be generated by **True*Grid*®**. This manual does not try to explain the meaning of the associated parameters. For this, the user is referred to the LS-DYNA User Manual. Some of these commands have additional options not shown here because they do not apply to the formation of a LS-DYNA output. For a full description of these commands, see the **True*Grid*®** User's Manual. Most of the commands used to form a complete input input file for LS-DYNA are only documented in the **True*Grid*®** User's Manual because they do not require special explanations for their use with LS-DYNA.

Conventions

When an arbitrarily long list of arguments are required, a semi-colon terminates the list. Sometimes the abbreviation #_*things* is used to mean "number of things" and *thing*_# for a "numbered thing". The square brackets "[...]" are used to indicate that something inside the square brackets can be repeated. Each command is described by an entry like the following:

command summary description

command *arguments* brief description of functionality
with brief descriptions of what the *arguments* should be.
indentation is used to indicate a list of options to the *arguments*

Remarks

When present, the Remarks section describes the command in even greater detail. It may describe the context in which the command is normally used, and other commands used in association with this command. It may describe side effects. It may describe other, similar commands. In many cases, it includes a description of where to find the command in the menus.

Example

When present, this shows the exact use of the command. If you use the dialogues, this command will be generated by simple selection options with the mouse and entering data where indicated. The command, as shown here, will appear in the session file for later reuse and possible modification. You can also enter the command into the text window or insert it into a command file to be run in batch mode.

bind Hughes-Liu beam user-defined integration points

bind *rule_# s_1 t_1 w_1 s_2 t_2 w_2 s_3 t_3 w_3 ... ;*

where *rule_#* can be any positive integer used to refer to this rule followed by a list of local coordinates of integration points s_i and t_i and corresponding weights w_i .

Remarks

The coordinates s_i and t_i are parametric coordinates of integration points from interval <-*1,1*> (Denoted by crosses). The weights w_i are determined from the term:

$$w_i = A_i / A$$

where A_i is the area corresponding to the *i-th* integration node. A is the total area of the cross section determined by:

$$A = \sum A_i.$$

The t_t and s_t dimensions are used for scaling from parametric to real coordinates. The t_t and s_t dimensions are specified using the **bsd**, **bm**, **ibm**, **ibmi**, **jbm**, **jbmi**, **kbm**, **kbmi**, or the **lsdymats** commands.

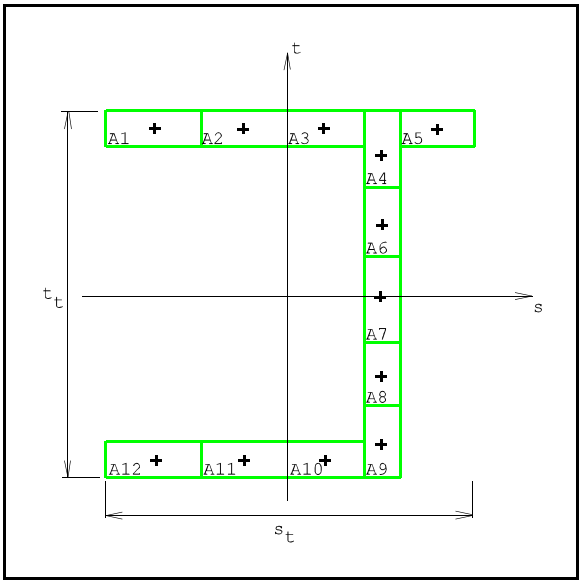


Figure 3 Cross Section with Integration Points

bm create a string of beam elements

bm *options ;*

where *option* can be

Selection of the first node

n1 *node_#*	to make an existing node the first node of the beams.
pm1 *point_mass_#*	to make a point mass node the first node of the beams.
rt1 *x y z const ;*	to create the first node of the beams in Cartesian coordinates.
cy1 *ρ θz const ;*	to create the first node of the beams in cylindrical coordinates.
sp1 *ρ θ ϕ const ;*	to create the first node of the beams in spherical coordinates.

Selection of the second node

n2 *node_#*	to make an existing node the last node of the beams.
pm2 *point_mass_#*	to make a point mass the last node of the beams.
rt2 *x y z const ;*	to create the last node of the beams in Cartesian coordinates.
cy2 *ρ θz const ;*	to create the last node of the beams in cylindrical coordinates.

sp2 ρ θ ϕ *const ;* to create the last node of the beams in spherical coordinates.

Selection of the orientation

n3 *node_#* to make an existing node the last node of the beams.
pm3 *point_mass_#* to make a point mass the last node of the beams.
rt3 *x y z const ;* to create the last node of the beams in Cartesian coordinates.
cy3 ρ θ*z const ;* to create the last node of the beams in cylindrical coordinates.
sp3 ρ θ ϕ *const ;* to create the last node of the beams in spherical coordinates.
orient *x y z* to specify a coordinate triple to orient the beams.
sd *surface_#* to orient beam axes in the orientation of the normal of the surface
v *x y z* to orient beam axes in the direction of the vector

Misc. options

mate *material_#* to specify the material number.
cs *cross_section_#* to specify the cross section number (see **bsd**).
nbms *number_of_beams* to specify the number of beams in the string (default is 1).
indc *const ;* to specify the constraints on the intermediate nodes.
cur *3d_curve_#* to interpolate the string of beams along a 3D curve.

where *const* can be any of

dx to constrain the x-displacement
dy to constrain the y-displacement
dz to constrain the z-displacement
rx to constrain the x-axis rotation
ry to constrain the y-axis rotation
rz to constrain the z-axis rotation

Selection of the nodal spacing

res *ratio* for relative spacing of nodes (default is equal spacing).
drs *1st_ratio 2nd_ratio* for double relative spacing of nodes.
nds *distribution_#* for nodal distribution by a function.
as 0 *1st_thickness* first element thickness
as 1 *last_thickness* last element thickness
das *1st_thickness last_thickness* first and last element thickness

Selection for Hughes-Liu Beam Thicknesses

sthi *sthi* for thickness in the y-direction.
sthi1 *sthi1* for thickness in the y-direction at the first end point.
sthi2 *sthi2* for thickness in the y-direction at the last end point.
tthi *tthi* for thickness in the z-direction.
tthi1 *tthi1* for thickness in the z-direction at the first end point.

tthi2 *tthi2* for thickness in the z-direction at the last end point.

Selection for Belytschko-Schwer beams or Trusses

csarea *csarea* for the cross section area

Selection for Belytschko-Schwer beams

sharea *sharea* shear area
inertia *Iss Itt Irr* inertia moments
vold *volume* volume of Discrete Beam
lump *inertia* lumped inertia

Selections for Discrete 3D beams

cablcid *system_#* local coordinate system id number defined by the **lsys**
cabarea *area* cable area
caboff *offset* cable offset

Selection of the nodal offsets

noint for no interior node offset interpolation
roff1 *roff1* for x-component of offset vector for first end point.
soff1 *soff1* for y-component of offset vector for first end point.
toff1 *toff1* for z-component of offset vector for first end point.
roff2 *roff2* for x-component of offset vector for last end point.
soff2 *soff2* for y-component of offset vector for last end point.
toff2 *toff2* for z-component of offset vector for last end point.

Selection of the pin flags

ldr1 *ldr1* to release the x-translation constraint at first end point.
lds1 *lds1* to release the y-translation constraint at first end point.
ldt1 *ldt1* to release the z-translation constraint at first end point.
lrr1 *lrr1* to release the rotation constraint about the x-axis at first end point.
lrs1 *lrs1* to release the rotation constraint about the y-axis at first end point.
lrt1 *lrt1* to release the rotation constraint about the z-axis at first end point.
ldr2 *ldr2* to release the x-translation constraint at first end point.
lds2 *lds2* to release the y-translation constraint at first end point.
ldt2 *ldt2* to release the z-translation constraint at first end point.
lrr2 *lrr2* to release the rotation constraint about the x-axis at first end point.
lrs2 *lrs2* to release the rotation constraint about the y-axis at first end point.

lrt2 *lrt2*	to release the rotation constraint about the z-axis at first end point.
ldr3 *ldr3*	to release the x-translation constraint at first end point.
lds3 *lds3*	to release the y-translation constraint at first end point.
ldt3 *ldt3*	to release the z-translation constraint at first end point.
lrr3 *lrr3*	to release the rotation constraint about the x-axis at first end point.
lrs3 *lrs3*	to release the rotation constraint about the y-axis at first end point.
lrt3 *lrt3*	to release the rotation constraint about the z-axis at first end point.

Remarks

This command is functional in the Merge Phase, and it is designed to create a string of beams or a single beam. We recommend that you use the dialogue box for **bm**.

You can use an existing node of the mesh for a beam, specify coordinates to create a new node for a beam, or you can use a point mass as a node for a beam. Coordinates can be specified in Cartesian, cylindrical, or spherical coordinates.

Beam orientation can be defined using a third node, using a point mass, or by creating another node in Cartesian, cylindrical, or spherical coordinates. Be sure to use the **lsdymats** command to select the beam element type and associated properties. Use the **bsd** to define a beam cross-section. Some of the beam element properties specified in the **lsdymats** command or the **bsd** command are the default values and are over ridden by values specified in the **bm** command. Some of the properties in this command are specific to the type of beam selected in the **lsdymats** command.

Nodes are automatically created if the number of beams specified is greater than 1.

You can define beam elements that follow a 3D curve, and specify the number of such elements, along with a spacing rule for the intermediate nodes.

Optional thickness parameters may be specified for the first and last beams when creating multiple beams. Intermediate beams will have thicknesses that are interpolated from the end beams. You may specify offsets for the first and last nodes, and optionally interpolate these offsets to intermediate nodes.

Constraints which couple the beams to the existing mesh can be eliminated. This may be done separately for the first, last, and intermediate nodes.

bsd global beam cross section definition

bsd *option_list ;*

where *option_list* depends on the element type selected in the **lsdymats** command and consists of some of the following:

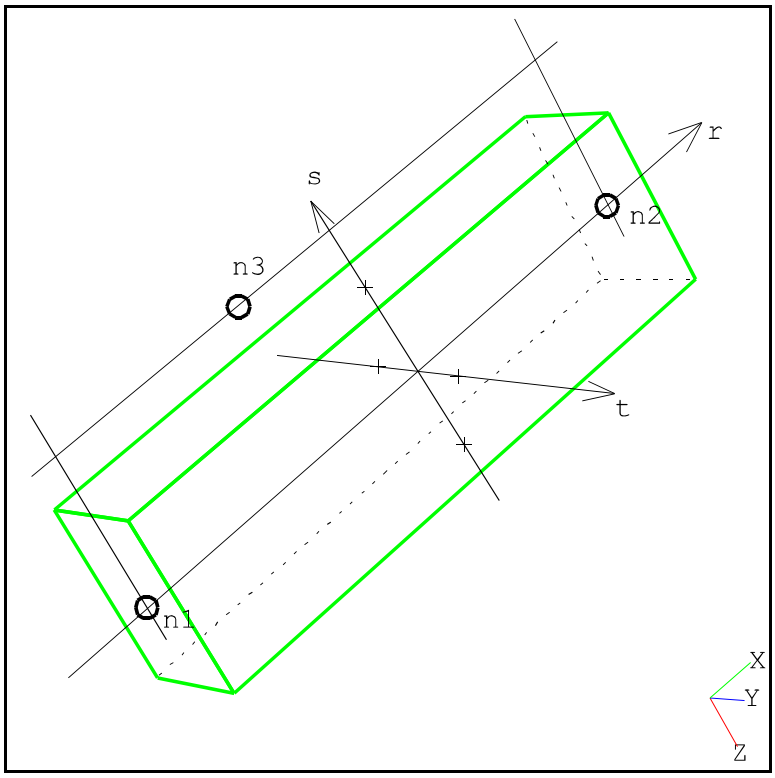


Figure 4 Beam Local Coordinate System for LS-DYNA

Hughes-Liu Standard Sections

lsd 1 *flange_width flange_thickness depth web_thickness* W-section
lsd 2 *flange_width flange_thickness depth web_thickness* C-section
lsd 3 *flange_width flange_thickness depth web_thickness* angle
lsd 4 *flange_width flange_thickness depth web_thickness* T-section
lsd 5 *flange_width flange_thickness depth web_thickness* rectangular
lsd 6 *flange_width flange_thickness depth web_thickness* Z-section
lsd 7 *flange_width depth web_thickness* trapezoidal
sref *location*

where *location* can be:

1 meaning the side where s is 1

0 meaning centered
-1 meaning the side where s is -1

tref *location*

where *location* can be:

1 meaning the side where t is 1
0 meaning centered
-1 meaning the side where t is -1

Hughes-Liu Constant Thickness or Diameter

sthi *thickness* s-thickness at both ends
tthi *thickness* t-thickness at both ends
sref *location*

where *location* can be:

1 meaning the side where s is 1
0 meaning centered
-1 meaning the side where s is -1

tref *location*

where *location* can be:

1 meaning the side where t is 1
0 meaning centered
-1 meaning the side where t is -1

Hughes-Liu Variable Thicknesses

sthi1 *thickness* s-thickness at beginning
sthi2 *thickness* s-thickness at ending
tthi1 *thickness* t-thickness at beginning
tthi2 *thickness* t-thickness at ending
sref *location*

where *location* can be:

1 meaning the side where s is 1
0 meaning centered
-1 meaning the side where s is -1

tref *location*

where *location* can be:

1 meaning the side where t is 1
0 meaning centered
-1 meaning the side where t is -1

Belytschko-Schwer beam

carea *area* cross section area
iss *iss* area moment of inertia about s-axis
itt *itt* area moment of inertia about t-axis

irr *irr* area moment of inertia about r-axis
sarea *area* shear area of cross section

Truss

carea *area* cross section area

Belytschko-Schwer Full Integration Beam Standart Sections

lsd 1 *flange_width flange_thickness depth web_thickness* W-section
lsd 2 *flange_width flange_thickness depth web_thickness* C-section
lsd 3 *flange_width flange_thickness depth web_thickness* angle
lsd 4 *flange_width flange_thickness depth web_thickness* T-section
lsd 5 *flange_width flange_thickness depth web_thickness* rectangular
lsd 6 *flange_width flange_thickness depth web_thickness* Z-section
lsd 7 *flange_width depth web_thickness* trapezoidal

Belytschko-Schwer Full Integration Beam Constant Thickness or Diameters

sthi *thickness* s-thickness or outer diameter at both ends
tthi *thickness* t-thickness or inner diameter at both ends

Belytschko-Schwer Full Integration Beam Variable Thicknesses or Diameters

sthi1 *thickness* s-thickness or outer diameter at beginning
sthi2 *thickness* s-thickness or outer diameter at ending
tthi**1** *thickness* t-thickness or inner diameter at beginning
tthi2 *thickness* t-thickness or inner diameter at ending

Belytschko-Schwer Tubular Beam Constant Diameter

sthi *outer_diameter* outer diameter at both ends
tthi *inner_diameter* inner diameter at both ends

Belytschko-Schwer Tubular Beam Variable Diameter

sthi1 *first_outer_diameter* (outer diameter at beginning)
sthi2 *last_outer_diameter* (outer diameter at ending)
tthi1 *first_inner_diameter* (inner diameter at beginning)
tthi2 *last_inner_diameter* (innter diameter at ending)

Discrete 3D Beam

vold *volume*
cabarea *cable_area*
lump *inertia* lumped geometric inertia
cablcid *local_coordinate_system_#* defined by the **lsys** command
caboff *cable_offset*

Spot Weld Beam Standart Sections

lsd 1	*flange_width flange_thickness depth web_thickness*	W-section
lsd 2	*flange_width flange_thickness depth web_thickness*	C-section
lsd 3	*flange_width flange_thickness depth web_thickness*	angle
lsd 4	*flange_width flange_thickness depth web_thickness*	T-section
lsd 5	*flange_width flange_thickness depth web_thickness*	rectangular
lsd 6	*flange_width flange_thickness depth web_thickness*	Z-section
lsd 7	*flange_width depth web_thickness*	trapezoidal

Spot Weld Beam Constant Thickness

sthi *thickness*	s-thickness at both ends
tthi *thickness*	t-thickness at both ends

Spot Weld Beam Variable Thicknesses

sthi1 *thickness*	s-thickness at beginning
sthi2 *thickness*	s-thickness at ending
tthi1 *thickness*	t-thickness at beginning
tthi2 *thickness*	t-thickness at ending

ibm generate beams in the i-direction (part phase)

ibm *region #_in_j #_in_k material orientation cross_section option* ;

where

#_in_j is the number of columns of beam elements in the j-direction

#_in_k is the number of columns of beam elements in the k-direction

material is the material number

orientation is the option of orientation of the cross section axis

j	second axis orientation in the j-direction
k	second axis orientation in the k-direction
sd *surface_#*	second axis orientation in the normal to the surface
v *xn yn zn*	second axis orientation by the vector
none	

cross_section is the cross-section definition number assigned with **bsd** *option* can be

reverse	the order of the nodes is the reverse of the default
si *sid_#*	Sliding Interface Number

Selections for Discrete 3D Beam

vold *volume*	volume of Discrete Beam
lump *inertia*	lumped inertia
cablcid *system_#*	local coordinate system id number defined by the **lsys**
cabarea *area*	cable area

caboff *offset* cable offset

Selection for Belytschko-Schwer beams or Trusses

csarea *area* cross section area

Selection for Belytschko-Schwer beams

sharea *area* shear area of cross section

inertia *iss itt irr* cross section moments of inertia

Selection for Hughes-Liu Beam and Spot Weld Beam Thicknesses

sthi *sthi* thickness in the y-direction.

sthi1 *sthi1* thickness in the y-direction at the first end point.

sthi2 *sthi2* thickness in the y-direction at the last end point.

tthi *tthi* thickness in the z-direction.

tthi1 *tthi1* thickness in the z-direction at the first end point.

tthi2 *tthi2* thickness in the z-direction at the last end point.

Selection of the nodal offsets

roff1 x x-component of offset vector for first end point.

soff1 y y-component of offset vector for first end point.

toff1 z z-component of offset vector for first end point.

roff2 x x-component of offset vector for last end point.

soff2 y y-component of offset vector for last end point.

toff2 z z-component of offset vector for last end point.

Selection of the pin flags

ldr1 release the x-translation constraint at first end point.

lds1 release the y-translation constraint at first end point.

ldt1 release the z-translation constraint at first end point.

lrr1 release the rotation constraint about the x-axis at first end point.

lrs1 release the rotation constraint about the y-axis at first end point.

lrt1 release the rotation constraint about the z-axis at first end point.

ldr2 release the x-translation constraint at last end point.

lds2 release the y-translation constraint at last end point.

ldt2 release the z-translation constraint at last end point.

lrr2 release the rotation constraint about the x-axis at last end point.

lrs2 release the rotation constraint about the y-axis at last end point.

lrt2 release the rotation constraint about the z-axis at last end point.

ldr3 release the x-translation constraint at intermediate point.

lds3 release the y-translation constraint at intermediate point.

ldt3 release the z-translation constraint at intermediate point.

lrr3 release the rotation constraint about the x-axis at intermediate points.

lrs3 release the rotation constraint about the y-axis at intermediate points.
lrt3 release the rotation constraint about the z-axis at intermediate points.

Remarks

This command is available only in the **block** or **cylinder** Part Phase. The **ibmi**, **jbm**, **jbmi**, **kbm**, and **kbmi** commands can also be used to generate embedded beam elements in the part phase. Their full description can be found in the **True*Grid*®** User's Manual. The **ibm** command is presented here as representative of this class of commands. These commands generate an array of beam elements conforming to the geometry and nodes of a solid or shell regions. This feature is useful in generating structural elements embedded within the solid or shell region.

Be sure to use the **lsdymats** command and select the beam element type and associated properties. Also use the **bsd** command to define the cross section properties of these beams. Some of the beam element properties specified in the **lsdymats** command and the **bsd** command are the default values and are over ridden by values specified in the **bm** command. Some of the properties in this command are specific to the type of beam selected in the **lsdymats** command.

The orientation of the beam can be selected in many ways or none at all in some cases (see the LS-DYNA manual).

The *v* option specifies a vector for the orientation. That vector is defined by the coordinate system. If the part is a cylinder, the vector is in the form of a radial, angular, and z-offset. Depending on the coordinates of the beam, the cylindrical vector will define a different orientation for each beam since the vector offset is made in cylindrical coordinates and then transformed to Cartesian coordinates.

lsdyeos LS-DYNA3D equation of state

lsdyeos *material_# eos_type parameter_list* ;
where the *eos_type* can be one of the types listed below and
where the *parameter_list* depending on *eos_type* and
where the *material_#* associates this equation of state with a numbered material.

For linear polynomial model type 1
c0 *constant*
c1 *coefficient*
c2 *coefficient*
c3 *coefficient*
c4 *coefficient*
c5 *coefficient*
c6 *coefficient*

e0 *energy*
v0 *volume*

For JWL model type 2
a *constant*
b *constant*
r1 *constant*
r2 *constant*
omega *constant*
e0 *energy*
v0 *volume*

For SACK model type 3
a1 *constant*
a2 *constant*
a3 *constant*
b1 *constant*
b2 *constant*
e0 *constant*
v0 *constant*

For GRUNEISEN model type 4
vci *intercept*
s1 *coefficient*
s2 *coefficient*
s3 *coefficient*
gamma *coefficient*
sa *coefficient*
e0 *energy*
v0 *volume*

For ratio of polynomials model type 5
a10 *constant*
a11 *constant*
a12 *constant*
a13 *constant*
a20 *constant*
a21 *constant*
a22 *constant*
a23 *constant*
a30 *constant*
a31 *constant*

a32 *constant*
a33 *constant*
a40 *constant*
a41 *constant*
a42 *constant*
a43 *constant*
a50 *constant*
a51 *constant*
a52 *constant*
a53 *constant*
a60 *constant*
a61 *constant*
a62 *constant*
a63 *constant*
a70 *constant*
a71 *constant*
a72 *constant*
a73 *constant*
alpha *constant*
beta *constant*
a14 *constant*
a24 *constant*
e0 *energy*
v0 *volume*

For linear polynomial with energy deposition model type 6

c0 *constant*
c1 *coefficient*
c2 *coefficient*
c3 *coefficient*
c4 *coefficient*
c5 *coefficient*
c6 *coefficient*
e0 *energy*
v0 *volume*
lc *load_curve*

For ignition and growth of reaction in he model type 7

ap *constant*
bp *constant*
r1p *constant*
r2p *constant*

g *coefficient*
wpcp *constant*
ae *constant*
be *constant*
wece *constant*
r1e *constant*
r2e *constant*
fcrit *fraction*
i *coefficient*
h *coefficient*
z *exponent*
x *exponent*
y *exponent*
cp *heat_capacity*
ce *heat_capacity*
m *exponent*
e0 *energy*
t0 *temperature*

For tabulated with compaction model type 8
eps *list_strains* ;
pc *list_constants* ;
t *list_temperatures* ;
ku *list_modulus* ;
gamma *gamma*
e0 *energy*
v0 *volume*

For tabulated model type 9
eps *list_strains* ;
pc *list_constants* ;
t *list_temperatures* ;
gamma *gamma*
e0 *energy*
v0 *volume*

For propellant model type 10

a *coef*	product JWL coefficient
b *coef*	product JWL coefficient
xp1 *coef*	product JWL coefficient
xp2 *coef*	product JWL coefficient
frer *volume*	unreacted c0-volume

g *product*	product wcv
r1 *coef*	unreacted JWL coefficient
r2 *coef*	unreacted JWL coefficient
r3 *coef*	unreacted wcv
r5 *coef*	unreacted JWL coefficient
r6 *coef*	unreacted JWL coefficient
fmxig *fraction*	initial fraction reacted f0
freq *pressure*	initial pressure p0
grow1 *coef*	first burn rate coefficient
em *exponent*	pressure exponent 1st term
ar1 *exponent*	exponent on F (1st term)
es1 *exponent*	exponent on 1-F (1st term)
cvp *capacity*	heat capacity products
cvr *capacity*	heat capacity unreacted
ccrit *volume*	product co-volume
enq *heat*	heat of reaction
tmp0 *temp*	initial temperature 298 k
grow2 *coef*	second burn rate coefficient
ar2 *exponent*	exponent on F (2nd term)
es2 *exponent*	exponent on 1-F (2nd term)
en *exponent*	pressure exponent 2nd term
fmxgr *max*	maximum F for 1st term
fmngr *min*	minimum F for 2nd term

For pore collapse model type 11

mu1 *compression*
mu2 *intersection*
e0 *energy*
mu0 *compression*
virgin *load_curve_pairs* ;
crushed *load_curve_pairs* ;
lcvir *load_curve_#*
lccru *load_curve_#*

For JWLB model type 13

a1 *coefficient*
a2 *coefficient*
a3 *coefficient*
a4 *coefficient*
a5 *coefficient*
r1 *coefficient*
r2 *coefficient*

r3 *coefficient*
r4 *coefficient*
r5 *coefficient*
al1 *coefficient*
al2 *coefficient*
al3 *coefficient*
al4 *coefficient*
al5 *coefficient*
bl1 *coefficient*
bl2 *coefficient*
bl3 *coefficient*
bl4 *coefficient*
bl5 *coefficient*
rl1 *coefficient*
rl2 *coefficient*
rl3 *coefficient*
rl4 *coefficient*
rl5 *coefficient*
c0 *coefficient*
omega *coefficient*
e0 *coefficient*
v0 *coefficient*

lsdymats LS-DYNA materials

lsdymats *material_# material_type parameter_list* ;
where *material_#* is the positive integer used to identify the material model and
where *material_type* can be any of the numbers associated with the models described below.

Check the LS-DYNA manual or use the dialogue boxes in **True*Grid*®** to avoid mismatches in options.

The following options can form part of the *parameter_list* and are available for all materials:

rho *density*
hgqt *type* Hourglass Stabilization Method
where *type* can be
1 standard
2 Flanagan-Belytschko integration
3 Flanagan-Belytschko integration with exact volume
4 stiffness form of type 2 (Flanagan-Belytschko)
5 stiffness form of type 3 (Flanagan-Belytschko)

hgq *coefficient* hourglass coefficient
bqt *type* bulk viscosity type
bqq *coefficient* quadratic viscosity coefficient
bql *coefficient* linear viscosity coefficient
migl *option*
where the gravity loading *option* can be
0 for all initialized
1 for only current material is initialized
head *comment* (1st 200 materials only)
aet *type* for ambient element type
where the ambient element *type* can be
1 for temperature
2 for pressure & temperature
3 for pressure outflow
4 for pressure inflow

The following option in the *parameter_list* is available for materials that do not allow failure and erosion:

excl *exclusion_number*
pfail *pressure_at_failure*
sigp1 *principal_stress_at_failure*
sigvm *equivalent_stress_at_failure*
epsp1 *principal_strain_at_failure*
epssh *shear_strain_at_failure*
sigth *threshold_stress*
impulse *stress_impulse_for_failure*

The following are element options that can be included in the *parameter_list*. Not all types are allowed for every material. Check the LS-DYNA Users Manual for the pairing of element types and material models or use the dialogue box for this command to select these properties. It is required that you choose the element type (brick, shell, thick shell, or beam) within these material models. The default is brick. The material can only be applied to elements of that type using the **mate**, **mt**, **mti**, or **mtv** commands.

Brick element types only.

brick for brick elements with the following options that can be added to the *parameter_list*:
elfob *option*
where the *option* can be
csb for constant stress brick
i8b for 8 point integration brick

i14b for 14 point integration quadratic 8-node brick
aleb for 1 point ALE brick
e1b for 1 point Eulerian brick
ea1b for 1 point Eulerian ambient brick
apb for acoustic pressure brick
crfm for 1 point crushable foam brick
alemm for 1 point ale multi-material brick
1wvoid for 1 point w/ single material and void brick
cflf for 1 point corotational crushable foam brick
apb8 for 8 point acoustic brick
ess8 for 8 point enhanced strain solid brick
ce4p for 4 point cohesive element
ce4o for 4 point cohesive element with offset for use with shells
ens for 1 point Eulerian Navier-Stokes
ens8 for 8 point Eulerian Navier-Stokes
mff for mesh-free solid formulation
sle for simplified linear brick

afac for simple average smoothing factor
bfac for volume weighting smoothing factor
cfac for isoparametric smoothing factor
dfac for equipotential smoothing factor
sts for start time for smoothing
ets for end time for smoothing
aaf for ALE advection factor

Shell element type only.

shell for shell elements with the following options that can be added to the *parameter_list*:
elfor *option*

where the *option* can be

hl for Hughes-Lui shell
bt for Belytschko-Tsay shell
bciz triangular shells
c0 triangular shells
membrane for Belytschko-Tsay membrane
srhl for s/r Hughes-Lui
srcr for s/r co-rotational Hughes-Lui
ew for Belytschko-Leviathan/Englemann-Whirley
blev for Belytschko-Leviathan/Englemann-Whirley
fibt for fully integrated Belytschko-Tsay membrane
bwc for Belytschko-Wong-Chaing
chl for co-rotational Hughes-Lui

fast for fully integrated shell (very fast)
psts for plane stress (xy-plane)
pstn for plane strain (xy-plane)
assa for axisymmetric solid - area weighted
assv for axisymmetric solid - volume weighted
fidk for fully integrated dk quad/tri
fc0 for fully integrated c0 shell
fc05 for fully integrated c0 shell(5 dof)
lsp3 for linear shear panel element
btth for Belytschko-Tsay with stretch
fcth for fully integrated shell with stretch
ens for 1 point Eulerian Navier-Stokes
ens8 for 8 point Eulerian Navier-Stokes
mfsf for mesh-free strain formulation
mfas for mesh-free axisymmetric solid form
sle for simplified linear shell

shear *factor*
tsti *#_points*
propt *option*

where *option* can be

1 for average resultants & fiber lengths
2 for resultants at plan points & fiber lengths
3 for resultants, stresses at all points & fiber lengths

userrl *integration_rule_#* (from **sind** command)
quad *integration_rule_#*

where the *integration_rule_#* can be

n positive for the number of points using the trapezoidal rule
0 for Gauss
-*n* negative number for the user specified rule (see **sind**)

shth *thickness*
shth1 *thickness*
shth2 *thickness*
shth3 *thickness*
shth4 *thickness*
shloc *location*

where *location* can be

1 for top surface
0 for middle surface
-1 for bottom surface

afac for simple average smoothing factor
bfac for volume weighting smoothing factor
cfac for isoparametric smoothing factor

dfac for equipotential smoothing factor
efac for equilibrium smoothing facto
sts for start time for smoothing
ets for end time for smoothing
aaf for ale advection factor
marea for non-structural mass per unit area
idof *option* for shell element types btth and fcth only
where *option* can be
1 for continuous thickness field across element edges
2 for discontinuous thickness field across element edges
seltyp *type* for 2d solid element type (for element types pstn, assa and assv only)
where *type* can be
1 for Lagrangian
2 for Eulerian
3 for ALE

Beam element types only.

beam for beam elements with the following options that can be added to the *parameter_list*:
elfom *option*
where the *option* can be
hl for Hughes-Lui beams
bs for Belytschko-Schwer beams
truss for truss
bsi for Belytschko-Schwer full integration beam
bst for Belytschko-Schwer tubular beam
dis1 Discrete 3D Beam
spw Spot Weld Beam
shear *factor*
userrl *integration_rule_#* (**bind** required)
stcs standart cross sctions (**bsd** is required)
quad *option*
where the *option* can be
1 for a truss
2 for 2x2 Gauss quadrature
3 for 3x3 Gauss quadrature
4 for 3x3 Lobatto integration
5 for 4x4 Gauss quadrature
bmcross *shape*
where the *shape* can be
0 for rectangular
1 for tubular

2 for standard or user define

sthi *thickness*
tthi *thickness*
sthi1 *thickness*
sthi2 *thickness*
tthi1 *thickness*
tthi2 *thickness*
sloc *location*

where *location* can be

1 meaning the side where s is 1
0 meaning centered
-1 meaning the side where s is -1

tloc *location*

where *location* can be

1 meaning the side where t is 1
0 meaning centered
-1 meaning the side where t is -1

vold *volume* for volume of discrete beam
lump *inertia* for lumped inertia
cablcid *local_coordinate_#* for a cable coordinate system Id
cabarea *area* for a cable area
caboff *offset* for a cable offset
cabrx for a cable rotational constraint r-direction
cabry for a cable rotational constraint s-direction
cabrz for a cable rotational constraint t-direction

Thick shell element types only.

tshell for thick shell elements with the following options that can be added to the *parameter_list*:

elfot *option*

where the *option* can be

spp for single point in plane quadrature thick shell
r22 for reduced 2x2 thick shell
s22 for assumed strain 2x2 in plan

shear *factor*
tsti *#_points*
propt *option*

where *option* can be

1 for average resultants & fiber lengths
2 for resultants at plan points & fiber lengths
3 for resultants, stresses at all points & fiber lengths

quad *integration_rule_#*

where the *integration_rule_#* can be

n	positive for the number of points using the trapezoidal rule
0	for Gauss quadrature
-*n*	negative of the user specified rule number (see **sind**)

Smooth Particle Hydro (SPH) element types only.

sph for sph elements with the following options that can be added to the *parameter_list*:

cslh *constant*	smoothing length constant
hmin *min*	minimum smoothing length factor
hmax *max*	maximum smoothing length factor
sphini *length*	initial smoothing length
sphdeath *time*	stopping time for sph
sphstart *time*	starting time for sph
hxcslh *length*	smoothing length constant in x
hycslh *length*	smoothing length constant in y
hzcslh *length*	smoothing length constant in z
hxini *length*	initial smoothing length in x
hyini *length*	initial smoothing length in y
hzini *length*	initial smoothing length in z
sphuser	user option

The remainder of options in the *parameter_list* are specific to the selected *material_type*:

For material type **1 struct** (Elastic)

e *young's_modulus*
da *axial_damping_factor*
db *bending_damping_factor*
pr *poisson's_ratio*

For material type **1 fluid** (Elastic Fluid)

k *bulk_modulus*
vc *tensor_viscosity_coefficient*
cp *cavitation_pressure*
pr *poisson's_ratio*

For material type **2 ortho** (Orthotropic Elastic)

ea *coefficient_ea*
eb *coefficient_eb*
ec *coefficient_ec*
prba *coefficient_vba*

prca *coefficient_vca*
prcb *coefficient_vcb*
gab *coefficient_gab*
gbc *coefficient_gbc*
gca *coefficient_gca*
aopt *option*

where the option can be one of

0	for by nodes
1	for by point and element center
2	for by normal vectors
3	for by cross product with shell normal (shell elements only)
4	for by normal vectors in cylindrical coordinates

xp *x-coordinate*	aopt 1
yp *y-coordinate*	aopt 1
zp *z-coordinate*	aopt 1
ax *x-component*	aopt 2
ay *y-component*	aopt 2
az *z-component*	aopt 2
dx *x-component*	aopt 2
dy *y-component*	aopt 2
dz *z-component*	aopt 2
vx *x-component*	aopt 3 & 4
vy *y-component*	aopt 3 & 4
vz *z-component*	aopt 3 & 4
p1 *x-component*	aopt 4
p2 *y-component*	aopt 4
p3 *z-component*	aopt 4

beta *angle*

For material type **2 aniso** (Anisotropic Elastic)

c11 *constitutive_matrix_coefficient*
c12 *constitutive_matrix_coefficient*
c22 *constitutive_matrix_coefficient*
c13 *constitutive_matrix_coefficient*
c23 *constitutive_matrix_coefficient*
c33 *constitutive_matrix_coefficient*
c14 *constitutive_matrix_coefficient*
c24 *constitutive_matrix_coefficient*
c34 *constitutive_matrix_coefficient*
c44 *constitutive_matrix_coefficient*
c15 *constitutive_matrix_coefficient*
c25 *constitutive_matrix_coefficient*

c35 *constitutive_matrix_coefficient*
c45 *constitutive_matrix_coefficient*
c55 *constitutive_matrix_coefficient*
c16 *constitutive_matrix_coefficient*
c26 *constitutive_matrix_coefficient*
c36 *constitutive_matrix_coefficient*
c46 *constitutive_matrix_coefficient*
c56 *constitutive_matrix_coefficient*
c66 *constitutive_matrix_coefficient*
aopt *option*

where the option can be one of

0	for by nodes
1	for by point and element center
2	for by normal vectors
3	for by cross product with shell normal (shell elements only)
4	for by normal vectors in cylindrical coordinates

xp *x-coordinate*	aopt 1
yp *y-coordinate*	aopt 1
zp *z-coordinate*	aopt 1
ax *x-component*	aopt 2
ay *y-component*	aopt 2
az *z-component*	aopt 2
dx *x-component*	aopt 2
dy *y-component*	aopt 2
dz *z-component*	aopt 2
vx *x-component*	aopt 3 & 4
vy *y-component*	aopt 3 & 4
vz *z-component*	aopt 3 & 4
p1 *x-component*	aopt 4
p2 *y-component*	aopt 4
p3 *z-component*	aopt 4

For material type **3** (Kinematic/Isotropic Elastic-Plastic)

e *young's_modulus*
pr *poisson's_ratio*
src *strain_rate_parameter*
srp *strain_rate_parameter*
sigy *yield_stress*
etan *tangent_modulus*
beta *hardening_parameter*
fs *failure_strain*
vp

For material type **4** (Thermo-Elastic-Plastic**)**

temp *temperature_list* **;**
e *young's_modulus_list* **;**
pr *poisson's_ratio_list* ;
alpha *secant_coefficient_list* ;
sigy *yield_stress_list* ;
etan *plastic_hardening_modulus_list* ;

For material type **5** (Soil and Crushable Foam)

g *shear_modulus*
ku *bulk_unloading_modulus*
a0 *yield_function_constant*
a1 *yield_function_constant*
a2 *yield_function_constant*
pc *pressure_cutoff*
vcr *volumetric_crushing_option*

where the volumetric crushing option can be

0	for on
1	for loading and unloading paths are the same

vs *volumetric_strain_list* ;
ps *pressure_list* ;

For material type **6** (Linear Viscoelastic)

k *bulk_modulus*
g0 *short_time_shear_modulus*
gi *long_time_shear_modulus*
beta decay_constant

For material type **7** (Blatz-Ko Rubber)

g *shear_modulus*

For material type **8** (High Explosive Burn)

d *detonation_velocity*
pcj *chapman-jouget_pressure*
beta *option*

where the option can be

0	for beta + programmed burn
1	for beta burn only
2	for programmed burn only

k *bulk_modulus*
g *shear_modulus*
sigy *yield_stress*

For material type **9** (Null (Hydrodynamic w/o Deviatoric Stress))

pc *pressure_cutoff*
mu *viscosity_coefficient*
rvt *relative_volume*
rvc *relative_volume*
e *young's_modulus*
pr *poisson's_ratio*

For material type **10** (Isotropic Elastoplastic Hydrodynamic)

g *shear_modulus*
sigy *yield_stress*
etan *plastic_hardening_modulus*
pc *pressure_cutoff*
eps *effective_plastic_strain_list* ;
es *yield_stress_list* ;

For material type **11** (Steinberg-Guinan Thermal Elastoplastic Hydrodynamic)

g0 *shear_modulus_constant*
sig0 *yield_stress_constant*
beta *strain_hardening_law_constant*
n *strain_hardening_exponent*
gama *initial_plastic_strain*
sigm *yield_stress_work_hardening_limit*
b *shear_modulus_pressure_constant*
bp *yield_stress_pressure_constant*
h *energy_coefficient*
f *energy_exponent_coefficient*
t0 *melting_temperature_constant*
gam0 *thermodynamic_gamma*
sa *thermodynamic_constant*
ispall *spall_model_option*

where the spall model option can be

pl	for p >= p_min
smax	for sigma_max >= sigma_p element spalls and tension
hydro	for p < p_min element spalls and tension

pcut *pressure_cutoff*
a *atomic_weight*
r *r_prime*
ivar *cold_compression_option*

where the cold compression option can be

0	for cold compression polynomial coefficient in eta
1	for cold compression polynomial coefficient in mu

ec0 *polynomial_coefficient_ec0*
ec1 *polynomial_coefficient_ec1*
ec2 *polynomial_coefficient_ec2*
ec3 *polynomial_coefficient_ec3*
ec4 *polynomial_coefficient_ec4*
ec5 *polynomial_coefficient_ec5*
ec6 *polynomial_coefficient_ec6*
ec7 *polynomial_coefficient_ec7*
ec8 *polynomial_coefficient_ec8*
ec9 *polynomial_coefficient_ec9*
luk *energy*
lc1 *prefact*
lc2 *coeffic*
lyp *stress*
lya *stress*
lymax *max*

For material type **12** (Isotropic Elastoplastic)
g *shear_modulus*
sigy *yield_stress*
eh *tangent_modulus*
k *bulk_modulus*

For material type **13** (Isotropic Elastoplastic with Failure)
g *shear_modulus*
sigy *yield_stress*
eh *tangent_modulus*
fs *effective_plastic_strain_at_failure*
fp *failure_pressure*
k *bulk_modulus*
rem *element_removal_option*
where the element removal option can be
0 for failed element eroded after failure
1 for element kept, no removal except by delta-t
dt *delta-t*

For material type **14** (Soil and Crushable Foam with Failure)
g *shear_modulus*
ku *bulk_unloading_modulus*
a0 *yield_function_constant*
a1 *yield_function_constant*
a2 *yield_function_constant*

pc *pressure_cutoff*
vcr *volumetric_crushing_option*
where the *volumetric crushing option* can be
0 for on
1 for loading and unloading paths are the same
vs *volumetric_strain_list* ;
ps *pressure_list* ;

For material type **15** (Johnson/Cook Plasticity)
g *shear_modulus*
a *yield_stress*
b *strain_hardening_coefficient*
n *strain_hardening_exponent*
sc *strain_rate_dependent_coefficient*
m *temperature_dependence_exponent*
tm *melt_temperature*
tr *room_temperature*
x0 *effective_plastic_strain_rate*
sh *specific_heat*
ispall *spall_model*
where the *spall model* option can be
pl for p >= p_min
smax for sigma_max >= sigma_p element spalls and tension
hydro for p < p_min element spalls and tension
pcut *pressure_cutoff*
iter *plastic_strain_iteration*
where the *plastic strain iteration* can be
fast for fast approximate solution for plastic strain
accurate for accurate iterative solution for plastic strain
d1 *failure_parameter*
d2 *failure_parameter*
d3 *failure_parameter*
d4 *failure_parameter*
d5 *failure_parameter*
e *young's_modulus*
pr *poisson's_ratio*
dtcrit *time_step_size*
vp

For material type **16** (Pseudo Tensor Concrete/Geological model)
g *shear_modulus*
pr *poissons_ratio*

sigy *tensile_cutoff*
a0 *cohesion*
a1 *1st_pressure_hardening*
a2 *2nd_pressure_hardening*
a0f *failed_mat_cohesion*
a1f *failed_mat_pressure_hardening*
b1 *damage_scaling_factor*
r *percent_reinforcement*
emr *reinforcement_elastic_modulus*
prr *reinforcement_poissons_ratio*
sigma0 *initial_yield_stress*
tm *tangent_modulus*
lc *principle_material_load* curve_#
lcr *reinforcement_load_curve_#*
opt *option*
where the option can be
1 for effective plastic strain curve w/ tensile cutoff
2 for effective plastic strain curve w/ max. principal stress failure
3 for pressure curve w/ tensile cutoff
eps *effective_plastic_strain_list* ;
es *yield_stress_list* ;

For material type **17** (Elastoplastic with Oriented Crack)
e *young's_modulus*
pr *poisson's_ratio*
sigy *yield_stress*
eh *tangent_modulus*
fs *fracture_strength*
pc *pressure_cutoff*

For material type **18** (Power Law Isotropic Plasticity)
e *young's_modulus*
pr *poisson's_ratio*
k *yield_stress_coefficient*
n *strain_hardening_exponent*
src *strain_rate_parameter*
srp *strain_rate_parameter*
sigy *parameter*
vp

For material type **19** (Strain Rate Dependent Plasticity)
e *young's_modulus*

pr *poisson's_ratio*
lcs0 *yield_stress_curve_#*
etan *tangent_modulus*
lce *young's_modulus_curve_#*
lcet *tangent_modulus_curve_#*
lcfs *effective_stress_failure_curve_#*
tss *auto_deletion_time_step*
vp

For material type **20** (Rigid)

e *young's_modulus*
nmad *option*
where the madymo3d coupling *option* can be
upno for normal ls-dyna3d rigid body updates
upel for couple to madymo ellipsoid
uppl for couple to madymo plane
coup *option*
where the madymo3d/cal3d coupling *option* can be
vdacoup for attach vda surface and generate mesh for ls-taurus.
meshcoup for undeformed geometry corresponds to local system
undcoup for undeformed geometry corresponds to global system
plncoup for generate a mesh for the ellipsoids and planes
seatoup for generate madymo seatbelts
conoup for generate a mesh for a contact entity
mopt *madymo/cal3d_system_#*
pr *poisson's_ratio*
alias *vda_file_name*
cmo center of mass constraint option
con constraints in the global coordinate system
spc constraints in the local coordinate system
lco *option*
where the *option* can be
id *local_sys_#* (see lsys)
df *x y z vx vy vz* local x axis and in-plane vector
xc *x* x-coordinate of center of mass for inertia
yc *y* y-coordinate of center of mass for inertia
zc *z* z-coordinate of center of mass for inertia
tm *mass* translational mass for inertia
nodeid *node_#* cg nodal point for inertia
ixx *xx* xx component of inertia tensor
ixy *xy* xy component of inertia tensor
ixz *xz* xz component of inertia tensor

iyy *yy* yy component of inertia tensor
iyz *yz* yz component of inertia tensor
izz *zz* zz component of inertia tensor
vtx *x* initial x-translational velocity for inertia
vty *y* initial y-translational velocity for inertia
vtz *z* initial z-translational velocity for inertia
vrx *x* initial x-rotational velocity for inertia
vry *y* initial y-rotational velocity for inertia
vrz *z* initial z-rotational velocity for inertia
ircs *flag* coordinate system flag for inertia

where *flag* can be
0 global inertia tensor
1 local inertia tensor

xl *x* x-coordinate of local axis of local inertia tensor
yl *y* y-coordinate of local axis of local inertia tensor
zl *z* z-coordinate of local axis of local inertia tensor
xlip *x* x-component of vector in local xy plane of local inertia tensor
ylip *y* y-component of vector in local xy plane of local inertia tensor
zlip *z* z-component of vector in local xy plane of local inertia tensor
cid *system_#* local coordinate system id of local inertia tensor
bpm *options* ;

where an *option* can be
dof *flag*

where *flag* can be
1 x-translational degree-of-freedom
2 y-translational degree-of-freedom
3 z-translational degree-of-freedom
4 translational motion in the given vector direction (use **v** below)
5 x-rotational degree-of-freedom
6 y-rotational degree-of-freedom
7 z-rotational degree-of-freedom
8 rotational motion about the given vector (use **v** below)
9 degree-of-freedom rotation about x-axis (use **offset** below)
-9 degree-of-freedom radial about x-axis (use **offset** below)
10 degree-of-freedom rotation about y-axis (use **offset** below)
-10 degree-of-freedom radial about y-axis (use **offset** below)
11 degree-of-freedom rotation about z-axis (use **offset** below)
-11 degree-of-freedom radial about z-axis (use **offset** below)

vad *flag*

where *flag* can be
0 velocity
2 displacement

3 velocity versus displacement
4 relative displacement

lcid *load_curve_#*
sf *scale_factor*
v x_0 y_0 z_0
birth *time*
death *time*
offset $offset_1$ $offset_2$
mrb *rigid_material_#*
nodes $node_1$ $node_2$

rbv *load_curve_# amplitude x y z* rigid body velocity (obsolete)
rbd *load_curve_# amplitude x y z* rigid body displacement (obsolete)

For material type **21** (Thermal Orthotropic w/ 12 Constants)

ea *a-elastic_modulus*
eb *b-elastic_modulus*
ec *c-elastic_modulus*
prba *ba-poisson's_ratio*
prca *ca-poisson's_ratio*
prcb *cb-poisson's_ratio*
alpa *a-thermal_expansion*
alpb *b-thermal_expansion*
alpc *c-thermal_expansion*
gab *ab-shear_modulus*
gbc *bc-shear_modulus*
gca *ca-shear_modulus*
aopt *option*

where the *option* can be one of

0 for by nodes
1 for by point and element center
2 for by normal vectors
3 for by cross product with shell normal (shell elements only)
4 for by normal vectors in cylindrical coordinates

xp *x-coordinate* aopt 1
yp *y-coordinate* aopt 1
zp *z-coordinate* aopt 1
ax *x-component* aopt 2
ay *y-component* aopt 2
az *z-component* aopt 2
dx *x-component* aopt 2
dy *y-component* aopt 2
dz *z-component* aopt 2

vx *x-component* aopt 3 & 4
vy *y-component* aopt 3 & 4
vz *z-component* aopt 3 & 4
p1 *x-component* aopt 4
p2 *y-component* aopt 4
p3 *z-component* aopt 4

For material type **22** (Fiber Composite w/ Damage)

ea *ea-elastic_modulus*
eb *eb-elastic_modulus*
ec *ec-elastic_modulus*
prba *vba-poissons_ratio*
prca *vca-poissons_ratio*
prcb *vcb-poissons_ratio*
gab *gab-shear_modulus*
gbc *gbc-shear_modulus*
gca *gca-shear_modulus*
k *k-bulk_modulus*
aopt *option*

where the *option* can be one of

0	for by nodes
1	for by point and element center
2	for by normal vectors
3	for by cross product with shell normal (shell elements only)

axes *flag* material axes change flag (brick elements only)

where *flag* can be

1	default
2	switch material axes a and b
3	switch material axes a and c

xp *x-coordinate* aopt 1
yp *y-coordinate* aopt 1
zp *z-coordinate* aopt 1
ax *x-component* aopt 2
ay *y-component* aopt 2
az *z-component* aopt 2
vx *x-component* aopt 3
vy *y-component* aopt 3
vz *z-component* aopt 3
dx *x-component* aopt 2
dy *y-component* aopt 2
dz *z-component* aopt 2
sc *strength* shear strength on a-b plane

xt *strength* longitudinal tensile strength along a-axis, xt
yt *strength* longitudinal tensile strength along b-axis, yt
yc *strength* transverse compressive strength, yc
alpha *stress* nonlinear shear stress parameter
sn *strength*
syz *strength*
szx *strength*
mangle *angle*
beta *list_of_material_angles ;* aopt 3

For material type **23** (Temperature Dependent Thermal Orthotropic w/ 12 Curves)

aopt *option*

where the option can be one of

0 for by nodes
1 for by point and element center
2 for by normal vectors
3 for by cross product with shell normal (shell elements only)

xp *x-coordinate* aopt 1
yp *y-coordinate* aopt 1
zp *z-coordinate* aopt 1
ax *x-component* aopt 2
ay *y-component* aopt 2
az *z-component* aopt 2
dx *x-component* aopt 2
dy *y-component* aopt 2
dz *z-component* aopt 2
vx *x-component* aopt 3
vy *y-component* aopt 3
vz *z-component* aopt 3
mangle *angle*
beta *list_of_material_angles ;* aopt 3
ea *ea-orthotropic_list* ;
eb *eb-orthotropic_list* ;
ec *ec-orthotropic_list* ;
vba *vba-orthotropic_list* ;
vca *vca-orthotropic_list* ;
vcb *vcb-orthotropic_list* ;
aa *alpha_a-orthotropic_list ;*
ab *alpha_b-orthotropic_list* **;**
ac *alpha_c-orthotropic_list* **;**
gab *gab-orthotropic_list* **;**
gbc *gbc-orthotropic_list* **;**

gca *gca-orthotropic_list* **;**
t *temperatures_list* **;**

For material type **24** (Rate-Dependent Tabular Isotropic Plasticity)

e *young's_modulus*
pr *poisson's_ratio*
sigy *yield_stress*
et *tangent_modulus*
efp *effective_strain_failure*
dtcrit *time_step_size*
srp *p-strain_rate*
src *c-strain_rate*
lcss *effective_stress_curve_#*
lc *yield_stress_curve_#*
vp *rate_effect_formulation*
eps *effective_strain_list* ;
es *yield_stress_list* ;
vp *type*

For material type **25** (Inviscid, Two Invariant Geologic Cap)

k *initial_bulk_modulus*
g *initial_shear_modulus*
alpha *failure_envelope*
theta *linear_failure_envelope*
gamma *exponential_failure_envelope*
beta *exponent_failure_envelope*
r *axis_ratio*
d *hardening_law_exponent*
w *hardening_law_coefficient*
x0 *hardening_law_parameter*
cbar *kinematic_hardening_coefficient*
n *kinematic_hardening_parameter*
nplot *option*

where the plot data base option can be

1	for hardening variable, k
2	for cap - j1 axis intercept, x(k)
3	for volumetric plastic strain
4	for first stress invariant, j1
5	for second stress invariant, square root of j2d
8	for response mode number
9	for number of iterations

ltype *option*

where the formulation option can be

1 for soil or concrete (cap surface may contract)

2 for rock (cap surface does not contract)

ivec *option*

where the vectorization option can be

0 for vectorized (fixed number of iterations)

1 for fully iterative

t *tension_cutoff*

For material type **26** (Orthotropic Crushable Honeycomb)

e *young's_modulus*
pr *poisson's_ratio*
sigy *yield_stress*
crv *relative_volume*
mu *material_viscosity*
bulk *option*

where the bulk viscosity option

0 for bulk viscosity is not used (recommended)

1 for bulk viscosity is active & mu=0, like previous versions of ls-dyna

sigaa *sigma_aa_curve_#*
sigbb *sigma_bb_curve_#*
sigcc *sigma_cc_curve_#*
ssrv *shear_stress_curve_#*
sigab *sig_ab_curve_#*
sigbc *sig_bc_curve_#*
sigca *sig_ca_curve_#*
sre *strain-rate effects_curve_#*
ea *ea-elastic_modulus*
eb *eb-elastic_modulus*
ec *ec-elastic_modulus*
gab *gab-shear_modulus*
gbc *gbc-shear_modulus*
gca *gca-shear_modulus*
aopt *option*

where the option can be one of

0 for by nodes

1 for by point and element center

2 for by normal vectors

3 by cross product with shell normal(Shell elements only)

xp *x-coordinate* aopt 1
yp *y-coordinate* aopt 1
zp *z-coordinate* aopt 1

ax *x-component* aopt 2
ay *y-component* aopt 2
az *z-component* aopt 2
dx *x-component* aopt 2
dy *y-component* aopt 2
dz *z-component* aopt 2
tsf *tensile_strain*
ssf *shear_strain*

For material type **27** (Compressible Mooney-Rivlin Hyperelastic Rubber)
pr *poisson's_ratio*
m1 *specify_constants*
a *first_invariant*
b *second_invariant*
m2 *least_square_fit*
sgl *specimen_gauge_length*
sw *specimen_width*
st *specimen_thickness*
lcid *ce_vs_change_curve_#*

For material type **28** (Resultant plasticity)
e *young's_modulus*
pr *poisson's_ratio*
sigy *yield_stress*
etan *hardening_modulus*

For material type **29** (Force Limited Resultant Formulation for Beams)
e *young's_modulus*
pr *poisson's_ratio*
df *damping_factor*
dept *for axial collapse force is dependent on the bending moment*
indept *for axial collapse force is not dependent on the bending moment*
noax *for no axial collapse*
bten *option*
where the beam tension option can be
0 for beam does not yield in tension
1 for beam can yield in tension
lpr *torsional_moment_curve_#*
sfr *lpr_scale_factor*
ymr *torsional_yield_moment*
asoft *softening_factor*
lps1 *1st_plastic_s-moment_curve_#*

sfs1 *lps1_scale_factor*
lps2 *2nd_plastic_s-moment_curve_#*
sfs2 *lps2_scale_factor*
yms1 *1st_s-axis_yield_moment*
yms2 *2nd_s-axis_yield_moment*
lpt1 *1st_plastic_t-moment_curve_#*
sft1 *lpt1_scale_factor*
lpt2 *2nd_plastic_t-moment_curve_#*
sft2 *lpt2_scale_factor*
ymt1 *1st_t-axis_moment*
ymt2 *2nd_t-axis_moment*

For material type **30** (Closed-Form Update Plasticity Shell)
e *young's_modulus*
pr *poisson's_ratio*
sigy *yield_stress*
etan *tangent_modulus*
sigass *starting_value*
sigasf *final_value*
sigsas *starting_value*
sigsaf *final_value*
epsl *recovered_strain*
alpha *measure_param*
ymrt *matensite_mod*

For material type **31** (Slightly Compressible Rubber Model)
pr *poisson's_ratio*
c100 *constant*
c200 *constant*
c300 *constant*
c400 *constant*
c110 *constant*
c210 *constant*
c010 *constant*
c020 *constant*
exct *exit_or_continue*
mxst *maximum_strain_limit*
mnst *minimum_strain_limit*
spgg *specimen_guage_length*
spwd *specimen_width*
sptk *specimen_thickness*
ldgl *load_curve*

For material type **32** (Laminated Glass Model)

e *young's_modulus*	glass
pr *poisson's_ratio*	glass
sigy *yield_stress*	glass
etan *hardening_modulus*	glass
psf *strain*	glass plastic strain at failure
pe *young's_modlus*	polymer
ppr *poisson's_ratio*	polymer
psigy *yield_stress*	polymer
petan *hardening_modulus*	polymer
imt *list* ;	list of 0s & 1s for integration points

For material type **33** (Barlat's Anisotropic Plasticity Model)

e *young's_modulus*
pr *poisson's_ratio*
ck *constant*
eps0 *constant*
cn *constant*

cm *potential*	flow potential in Barlat's model
ca *coef*	anistropy coefficient in Barlat's model
cb *coef*	anistropy coefficient in Barlat's model
cc *coef*	anistropy coefficient in Barlat's model
cf *coef*	anistropy coefficient in Barlat's model
cg *coef*	anistropy coefficient in Barlat's model
ch *coef*	anistropy coefficient in Barlat's model

aopt *option*

where the option can be one of

0	for by nodes
1	for by point and element center
2	for by normal vectors
3	by cross product with shell normal(Shell elements only)

xp *x-coordinate*	aopt 1
yp *y-coordinate*	aopt 1
zp *z-coordinate*	aopt 1
ax *x-component*	aopt 2
ay *y-component*	aopt 2
az *z-component*	aopt 2
dx *x-component*	aopt 2
dy *y-component*	aopt 2
dz *z-component*	aopt 2
vx *x-component*	aopt 3
vy *y-component*	aopt 3

vz *z-component* aopt 3
mangle *angle*
beta *list_angle* ;
yld96
esr0 *factor*
hard *option*
c1 *value*
c2 *value*
c3 *value*
c4 *value*
alphax *value*
alphay *value*
alphaz0 *value*
alphaz1 *value*

For material type **34** (Fabric Model)

ea *direction* longitudinal direction
eb *direction* transverse direction
ec *direction* normal direction
vba *poisson's_ratio*
vca *poisson's_ratio*
vcb *poisson's_ratio*
csflg *flag* compressive stress flag
where the flag can be
0 for don't eliminate compressive stresses
1 for eliminate compressive stresses
gab *coefficient*
gbc *coefficient*
gca *coefficient*
csflg *flag*
e *young's_modulus* Young's modulus for elastic liner
pr *poisson's_ratio* Poisson's ratio for elastic liner
rlf *ratio* ratio of liner thickness to total fabric thickness
rla *coef* Rayleigh damping coefficient
aopt *option*
where the option can be one of
0 for by nodes
1 for by point and element center
2 for by normal vectors
3 by cross product with shell normal (Shell elements only)
xp *x-coordinate* aopt 1
yp *y-coordinate* aopt 1

zp *z-coordinate* aopt 1
ax *x-component* aopt 2
ay *y-component* aopt 2
az *z-component* aopt 2
vx *x-component* aopt 3
vy *y-component* aopt 3
vz *z-component* aopt 3
dx *x-component* aopt 2
dy *y-component* aopt 2
dz *z-component* aopt 2
mangle angle
beta *list_of_material_angles* aopt 3
flc *fabric_leakage_coefficient*
fac *fabric_area_coefficient*
ela *effective_leakage_area*
lnrc *liner_compression_flag*
form *membrane_formulation_flag*
fvopt *option*
lca *lc*
lcb *lc*
lcab *lc*
lcua *lc*
lcub *lc*
lcuab *lc*

For material type **35** (Kinematic/Isotropic Elastic-Plastic Green-Naghdi Rate Model)

e *young's_modulus*
src *c* strain rate parameter, c
srp *p* strain rate parameter, p
pr *poisson's_ratio*
sigy *yield_stress*
etan *hardening_modulus*
eb *hardening_parameter*

For material type **36** (Barlat's 3-Parameter Plasticity Model)

e *young's_modulus*
pr *poisson's_ratio*
hr *rule* hardening rule
where the rule can be
1 for linear model
2 for exponential model
cm *exponent* exponent in Barlat's yield surface

r00 *constant*
r45 *constant*
r90 constant
lcid *load_curve*
e0 *parameter*
spi *parameter*
beta l*ist_angle* ;
aopt *option*
where the option can be one of

0	for by nodes
1	for by point and element center
2	for by normal vectors
3	by cross product with shell normal(Shell elements only)

xp *x-coordinate*	aopt 1
yp *y-coordinate*	aopt 1
zp *z-coordinate*	aopt 1
ax *x-component*	aopt 2
ay *y-component*	aopt 2
az *z-component*	aopt 2
dx *x-component*	aopt 2
dy *y-component*	aopt 2
dz *z-component*	aopt 2
vx *x-component*	aopt 3
vy *y-component*	aopt 3
vz *z-component*	aopt 3

For material type **37** (Transversely Anisotropic Elastic-Plastic)
e *young's_modulus*
pr *poisson's_ratio*
sigy *yield_stress*
etan *hardening_modulus*

er *r*	anisotropic hardening parameter
ldss *load_curve*	effective stress vs. effective plastic strain

For material type **38** (Blatz-Ko Compressible Foam)
sm *shear_modulus*

For material type **39** (Transversely Anisotropic Elastic-Plastic with FLD)
e *young's_modulus*
ldfl *load_curve* defining the flow limit diagram
pr *poisson's_ratio*
sigy *yield_stress*

etan *hardening_modulus*
er *r* anisotropic hardening parameter
ldss *load_curve* effective stress vs. effective plastic strain

For material type **40** (Nonlinear Elastic Orthotropic Material)
ex0 *direction* modulus-longitudinal direction
ey0 *direction* modulus-transverse direction
ez0 *direction* modulus-normal direction
vba *poisson's_ratio*
vca *poisson's_ratio*
vcb *poisson's_ratio*
dtm *dt* temperature increment for stress initialization
trmp *t-ramp* time to ramp up to the final temperature
alpha *alpha* thermal expansion coefficient
gab *shear_modulus*
gca *shear_modulus*
gbc *shear_modulus*
aopt *option*
where the option can be one of
0 for by nodes
1 for by point and element center
2 for by normal vectors
3 by cross product with shell normal(Shell elements only)
xp *x-coordinate* aopt 1
yp *y-coordinate* aopt 1
zp *z-coordinate* aopt 1
ax *x-component* aopt 2
ay *y-component* aopt 2
az *z-component* aopt 2
dx *x-component* aopt 2
dy *y-component* aopt 2
dz *z-component* aopt 2
vx *x-component* aopt 3
vy *y-component* aopt 3
vz *z-component* aopt 3
beta *list_of_material_angles* aopt 3
ldnsa *load_curve* nominal stress vs. a-axis strain
ldnsb *load_curve* nominal stress vs. b-axis strain
efail *epsilon-fail* failure strain
dtfail *dt-fail* time step for automatic element erosion
cdamp *c-damp* damping coefficient

For material type **41** to **50** (User-Defined Material Models)

card1 *parameters* ;
card2 *parameters* ;
card3 *parameters* ;
card4 *parameters* ;
card5 *parameters* ;
card6 *parameters* ;
card7 *parameters* ;
card8 *parameters* ;
card9 *parameters* ;
card10 *parameters* ;
aopt *option*

where the option can be one of

0	for by nodes
1	for by point and element center
2	for by normal vectors
3	by cross product with shell normal(Shell elements only)

xp *x-coordinate*	aopt 1
yp *y-coordinate*	aopt 1
zp *z-coordinate*	aopt 1
ax *x-component*	aopt 2
ay *y-component*	aopt 2
az *z-component*	aopt 2
dx *x-component*	aopt 2
dy *y-component*	aopt 2
dz *z-component*	aopt 2
vx *x-component*	aopt 3
vy *y-component*	aopt 3
vz *z-component*	aopt 3
mopt *flag*	material axes change flag for bricks

where flag can be

1	default
2	switch material axes a and b
3	switch material axes a and c

nhv *n*	number of history variables
iortho *flag*	orthotropic flag
ibulk *address*	address of bulk modulus array
ig *address*	address of shear modulus array
ivect *flag*	vectorization flag
ifail *flag*	failure flag
ithermal *flag*	temperature flag

ihyper *flag*

where *flag* can be
-1 for deformation gradient between reference global and material frames (orthotropic materials only)
0 for deformation gradient off
1 for deformation gradient on

mangle *angle*
beta *list_angle* ;

For material type **51** (Temperature and Rate Dependent Plasticity)

e *young's_modulus*
pr *poisson's_ratio*
t *initial_temperature*
hc *coef* heat generation coefficient
c1 *constant*
c2 *constant*
c3 *constant*
c4 *constant*
c5 *constant*
c6 *constant*
c7 *constant*
c8 *constant*
c9 *constant*
c10 *constant*
c11 *constant*
c12 *constant*
c13 *constant*
c14 *constant*
c15 *constant*
c16 *constant*
c17 *constant*
c18 *constant*
al1 *alpha1* initial value of state variable 1
al2 *alpha2* initial value of state variable 2
al4 *alpha4* initial value of state variable 3
al5 *alpha5* initial value of state variable 4
al6 *alpha6* initial value of state variable 5
kap *kappa* initial value of state variable 6

For material type **52** (SANDIA's Damage Model)

e *young's_modulus*
pr *poisson's_ratio*

t *initial_temperature*
hc *coef* heat generation coefficient
c1 *constant*
c2 *constant*
c3 *constant*
c4 *constant*
c5 *constant*
c6 *constant*
c7 *constant*
c8 *constant*
c9 *constant*
c10 *constant*
c11 *constant*
c12 *constant*
c13 *constant*
c14 *constant*
c15 *constant*
c16 *constant*
c17 *constant*
c18 *constant*
al1 *alpha1* initial value of state variable 1
al2 *alpha2* initial value of state variable 2
al4 *alpha4* initial value of state variable 3
al5 *alpha5* initial value of state variable 4
al6 *alpha6* initial value of state variable 5
kap *kappa* initial value of state variable 6
exp *exponent* damage evolution
por *porosity* do initial damage

For material type **53** (Low Density Closed Cell Polyurethane Foam)
e *young's_modulus*
ca *constant*
cb *constant*
cc *constant*
p0 *pressure* initial foam pressure
phi *ratio* foam to polymer density
g0 *gamma0* initial volumetric strain
lcid *load_curve*

For material type **54** (Composite Damage Model (Chang matrix failure))
ex *direction* longitudinal direction
ey *dircetion* transverse direction

ez *dircetion* normal direction
vba *poisson's_ratio*
vca *poisson's_ratio*
vcb *poisson's_ratio*
gab *shear_modulus*
gca *shear_modulus*
gbc *shear_modulus*
aopt *option*
where the option can be one of

0	for by nodes
1	for by point and element center
2	for by normal vectors
3	by cross product with shell normal (Shell elements only)

xp *x-coordinate* aopt 1
yp *y-coordinate* aopt 1
zp *z-coordinate* aopt 1
ax *x-component* aopt 2
ay *y-component* aopt 2
az *z-component* aopt 2
vx *x-component* aopt 3
vy *y-component* aopt 3
vz *z-component* aopt 3
dx *x-component* aopt 2
dy *y-component* aopt 2
dz *z-component* aopt 2
dfailm *maximum_strain_for_matrix*
dfails *maximum_shear_strain*
tfail *step* time step for element deletion
ns *stress* nonlinear shear stress parameter
soft *factor* softening reduction factor for material strength in crashfront elements
frbt *softening* fiber tensile strength
ycfac *compression_strength* remainder longitudinal compression strength after compressive matrix
dfailt *strain* failure strain for tensile fiber mode
dfailc *strain* failure strain (negative) for compressive fiber mode
efs *effective_failure_strain*
xc *strength* longitudinal compressive strength
xt *strength* longitudinal tensile strength, a-axis
yc *strength* transverse compressive strength
yt *strength* transverse tensile strength, b-axis
sc *strength* shear strength, ab plane
bt *factor* weighting factor for shear term in tensile fiber mode

mangle *angle*
beta *list_of_material_angles* aopt 3

For material type **55** (Composite Damage Model (tsay-wu matrix failure))

ex *direction* longitudinal direction
ey *dircetion* transverse direction
ez *dircetion* normal direction
vba *poisson's_ratio*
vca *poisson's_ratio*
vcb *poisson's_ratio*
gab *shear_modulus*
gca *shear_modulus*
gbc *shear_modulus*
aopt *option*
where the option can be one of
0 for by nodes
1 for by point and element center
2 for by normal vectors
3 by cross product with shell normal (Shell elements only)
xp *x-coordinate* aopt 1
yp *y-coordinate* aopt 1
zp *z-coordinate* aopt 1
ax *x-component* aopt 2
ay *y-component* aopt 2
az *z-component* aopt 2
vx *x-component* aopt 3
vy *y-component* aopt 3
vz *z-component* aopt 3
dx *x-component* aopt 2
dy *y-component* aopt 2
dz *z-component* aopt 2
dfailm *maximum_strain_for_matrix*
dfails *maximum_shear_strain*
tfail *step* time step for element deletion
ns *stress* nonlinear shear stress parameter
soft *factor* softening reduction factor for material strength in crashfront elements
frbt *softening* fiber tensile strength
ycfac *compression_strength* remainder longitudinal compression strength after compressive matrix
dfailt *strain* failure strain for tensile fiber mode
dfailc *strain* failure strain (negative) for compressive fiber mode
efs *effective_failure_strain*

xc *strength* longitudinal compressive strength
xt *strength* longitudinal tensile strength, a-axis
yc *strength* transverse compressive strength
yt *strength* transverse tensile strength, b-axis
sc *strength* shear strength, ab plane
bt *factor* weighting factor for shear term in tensile fiber mode
mangle *angle*
beta *list_of_material_angles* aopt 3

For material type **57** (Low Density Urethane Foam)
e *young's_modulus*
ldns *load_curve* nominal stress vs. strain
tcut *stress* tension cut-off stress
hunl *factor* hysteretic unloading factor
beta *decay*
vc *viscous_coefficient* stress oscillations and shock waves
sf *factor* shape factor for unloading
fopt *option* failure option after cutoff stress is reached
where the failure option can be
0 tensile stress remains at cut-off value
1 tensile stress is reset to zero
kflg *flag* bulk viscosity activation flag
where the flag can be
0 no bulk viscosity (recommended)
1 bulk viscosity active
oe *modulus* optional young's relaxation modulus for rate effects
obeta *decay* optional decay constant
sc *stiffness_coefficient* contact interface stiffness

For material type **59** (Composite Failure Model - Plasticity Based)
ea *direction* longitudinal direction
eb *direction* transverse direction
ec *direction* normal direction
kf *modulus* bulk modulus of failed material
sr *factor* reduction factor (default=0.447) (shells)
sf *factor* softening factor (default=0.0) (shells)
vba *poisson's_ratio*
vca *poisson's_ratio*
vcb *poisson's_ratio*
sba *strength* in plane shear strength (bricks)
sca *strength* transverse shear strength (bricks)
scb *strength* transverse shear strength (bricks)

gab *shear_modulus*
gca *shear_modulus*
gbc *shear_modulus*
xc *strength* long. compressive strength, a-axis
yc *strength* trans. compressive strength, b-axis
zc *strength* normal compressive strength, c-axis
aopt *option*
where the option can be one of
0 for by nodes
1 for by point and element center
2 for by normal vectors
3 by cross product with shell normal(Shell elements only)
4 by normal vectors in cylindrical coordinates
xp *x-coordinate* aopt 1
yp *y-coordinate* aopt 1
zp *z-coordinate* aopt 1
ax *x-component* aopt 2
ay *y-component* aopt 2
az *z-component* aopt 2
dx *x-component* aopt 2
dy *y-component* aopt 2
dz *z-component* aopt 2
vx *x-component* aopt 3 & 4
vy *y-component* aopt 3 & 4
vz *z-component* aopt 3 & 4
px *x-component* aopt 4
py *y-component* aopt 4
pz *z-component* aopt 4
mopt *flag* material axes change flag for bricks
where flag can be
1 default
2 switch material axes a and b
3 switch material axes a and c
tsize *step* time step for automatic element deletion
alp *stress* nonlinear shear stress parameter
soft *factor* softening reduction factor for strength in crash
fbrt *strength* softening of fiber tensile strength
xt *strength* long. tensile strength, a-axis (bricks)
yt *strength* trans. tensile strength, b-axis (bricks)
zt *strength* normal tensile strength, c-axis (bricks)
xcs *strength* long. compressive strength, a-axis (shells)
xts *strength* long. tensile strength, a-axis (shells)

ycs *strength* trans. compressive strength, b-axis (shells)
yts *strength* trans. tensile strength, b-axis (shells)
sc *strength* shear strength, ab-plane (shells)
mangle *angle*
beta *list_of_material_angles* aopt 3

For material type **60** (Elastic with Viscosity)

e *young's_modulus*
visc *viscosity*
ca *viscosity_coefficient*
cb *viscosity_coefficient*
cc *viscosity_coefficient*
ldft *load_curve* defining factor vs. time
prs *ratio_list* ; Poisson's ratio (up to 8 values)
ts *temp* ; temperature values (up to 8 values)
viscs *viscosity_list* ; viscosity values (up to 8 values)
es *modulus_list* ; Young's modulus values (up to 8 values)
ctes *coef_list* ; coefficient of thermal expansion (up to 8 values)

For material type **61** (Maxwell/Kelvin Viscoelastic with Maximum Strain)

k *bulk_modulus* elastic
g0 *modulus* short-time shear modulus
ginf *modulus* long-time shear modulus
form *option* formulation option
where the option can be
0 Maxwell
1 Kelvin
mdec *decay*
kdec *relaxation*
sopt *option* strain output option
where the option can be
0 maximum principal strain
1 maximum magnitude of principal strain
2 maximum effective strain

For material type **62** (Viscous Foam, Ove Arup & Partners Model)

e1 *modulud* initial Young's modulus
n1 *exponent* power law for Young's modulus
v2 *viscous_coefficient*
e2 *modulud* elastic modulus for viscosity
n2 *exponent* power law for viscosity
pr *poisson's_ratio*

For material type **63** (Crushable foam)

e *young's_modulus*
pr *poisson's_ratio*
ldyv *load_curve* yield stress vs. volumetric strain
tcut *stress* cutoff value for tensile stress
visc *viscous_coefficient*

For material type **64** (Strain Rate Sensitive Power-Law Plasticity)

me *modulus* modulus of elasticity
pr *poisson's_ratio*
mk *material_constant*
sm *coef* strain hardening coefficient
sn *coef* strain rate sensitivity coefficient
is *initial_strain*
vp

For material type **65** (Modified Zerilli/Armstrong)

cg *constant*
ceps0 *constant*
cn *constant*
tr *room_temperature*
pc *pressure_cutoff*
sp *spall_type*
where the spall type can be
1 Minimum pressure limit
2 Maximum principal stress (default)
3 Minimum pressure cutoff
fs *strain* failure strain for erosion
c1 *constant*
c2 *constant*
c3 *constant*
c4 *constant*
c5 *constant*
c6 *constant*
b1 *constant*
b2 *constant*
b3 *constant*
g1 *constant*
g2 *constant*
g3 *constant*
g4 *constant*
vp

For material type **66** (Linear Stiffness/Linear Viscous 3d Discrete Beam)

tsr *stiffness*	translational stiffness along r-axis
tss *stiffness*	translational stiffness along s-axis
tst *stiffness*	translational stiffness along t-axis
rsr *stiffness*	rotational stiffness about r-axis
rss *stiffness*	rotational stiffness about s-axis
rst *stiffness*	rotational stiffness about t-axis
tvr *stiffness*	translational viscous damper along r-axis
tvs *stiffness*	translational viscous damper along s-axis
tvt *stiffness*	translational viscous damper along t-axis
rvr *damper*	rotational viscous damper about r-axis
rvs *damper*	rotational viscous damper about s-axis
rvt *damper*	rotational viscous damper about t-axis

For material type **67** (Nonlinear Stiffness/Viscous 3d Discrete Beam)

ldrr *load_curve*	r-axis force vs. r-axis displacement
ldss *load_curve*	s-axis force vs. s-axis displacement
ldtt *load_curve*	t-axis force vs. t-axis displacement
ldmrr *load_curve*	r-axis moment vs. r-axis rotation
ldmss *load_curve*	s-axis moment vs. s-axis rotation
ldmtt *load_curve*	t-axis moment vs. t-axis rotation
ldvrr *load_curve*	r-axis damping force vs. r-axis rotational velocity
ldvss *load_curve*	s-axis damping force vs. s-axis rotational velocity
ldvtt *load_curve*	t-axis damping force vs. t-axis rotational velocity
ldvmrr *load_curve*	r-axis damping moment vs. r-axis rotational velocity
ldvmss *load_curve*	s-axis damping moment vs. s-axis rotational velocity
ldvmtt *load_curve*	t-axis damping moment vs. t-axis rotational velocity

For material type **68** (Nonlinear Plastic/Linear Viscous 3d Discrete Beam)

tsr *stiffness*	translational stiffness along r-axis
tss *stiffness*	translational stiffness along s-axis
tst *stiffness*	translational stiffness along t-axis
rsr *stiffness*	rotational stiffness about r-axis
rss *stiffness*	rotational stiffness about s-axis
rst *stiffness*	rotational stiffness about t-axis
tvr *damper*	translational viscous damper along r-axis
tvs *damper*	translational viscous damper along s-axis
tvt *damper*	translational viscous damper along t-axis
rvr *damper*	rotational viscous damper about r-axis
rvs *damper*	rotational viscous damper about s-axis

rvt *damper* rotational viscous damper about t-axis
ldyr *load_curve* yield force vs. plastic displacement, r-axis
ldys *load_curve* yield force vs. plastic displacement, s-axis
ldyt *load_curve* yield force vs. plastic displacement, t-axis
ldmr *load_curve* yield moment vs. plastic rotation, r-axis
ldms *load_curve* yield moment vs. plastic rotation, s-axis
ldmt *load_curve* yield moment vs. plastic rotation, t-axis
fr *para* fr-fail optional parameter
fs *para* fs-fail optional parameter
ft *para* ft-fail optional parameter
mr *para* mr-fail optional parameter
ms *para* ms-fail optional parameter
mt *para* mt-fail optional parameter
ur *para* ur-fail optional parameter
us *para* us-fail optional parameter
ut *para* ut-fail optional parameter
tr *para* thetar-fail optional parameter
ts *para* thetas-fail optional parameter
tt *para* thetaj-fail optional parameter

For material type **69** (Side Impact Dummy Damper, sid Damper)
st *st* piston stroke
cd *d* piston diameter
cr *r* default orifice radius
ch *h* orifice controller position
ck *k* damping constant
cc *c* discharge coefficient
ckk *k* stiffness coefficient if piston bottoms out
cds *list* ; orifice locations relative to fixed end (no more than 16)
crs *list* ; radii corresponding to orifice locations
rho *rho* fluid density
c1 *c1* coefficient for linear velocity term
c2 *c2* coefficient for quadratic velocity term
ldfd *load_curve* force vs. piston displacement
lddd *load_curve* damping coefficient vs. piston displacement
d0 *s0* initial displacement (typically 0)
c3 *c3* coefficient for fluid inertia term
sf *scale_factor*
lvdc *coefficient*

For material type **70** (Hydraulic/Gas Damper Model)
c0 *c0* length of gas column

cn *n* adiabatic constant
p0 *p0* initial gas pressure
pa *pa* atmospheric pressure
ap *ap* piston cross sectional area
kh *kh* hydraulic constant
ldn *load_curve* orifice area vs. element deflection
fr *fr* return factor on orifice force
sclf *sclf* scale factor on force
clr *clearance*

For material type **71** (Cable Model)
e *young's_modulus*
ldn *load_curve* stress vs. strain

For material type **72** (Concrete Damage Model0
pr *pr* Constant Poission's Ratio Model
sigf *failure* Maximum Principal Stress Failure
a0 *cohesion* Cohesion
a1 *coefficient* Fiirst Pressure Hardening Coefficient
a2 *coefficient* Second Pressure Hardening Coefficient
b1 *factor* Damage Scaling Factor
a1f *coefficient* Pressure Hardening Coefficient for Failed Material
per *percent* Percent Reinforcement
er *elastic modulus* Elastic Modulus for Reinforcement
prr *poisson's ratio* Poisson's Ratio for Reinforcement
sigy *yield stress* Initial Yield Stress for Reinforcement
etan *tangent modulus* Tangent Modulus for Reinforcement
lcp *load curve* Load Curve Giving Rate Sensitivity for Principal Material
lcr *load curve* Load Curve giving Rate Sensitivity for Reinforcement
lambda *tabulated values (up to 13)* Damage Function
b3 *factor* Damage Scale Factor for Triaxial Tensile Path
a0y *cohesion* Cohesion for Yield Limit
a1y *hardening* Pressure Hardening for Yield Limit
eta *tabulated values (up to 13)* Scale Factor Function
b2 *factor* Damage Scaling Factor for Triaxial Tension Path
a2f *coefficient* Pressure Hardening Coefficient for Failed Material
a2y *coefficient* Pressure Hardening Coefficient for Yield Limit

For material type **73** (Low Density Viscoelastic Foam)
e *young's modulus* Young's Modulus
lcid *load_curve* Load Curve number of nominal stress versus strain
tc *cut-off_stress* Tension Cut-off Stress

hu *factor* Hysteretic Unloading Factor
beta *decay_constant* Decay Constant to Model Creep in Unloading
damp *coefficient* Viscous Coefficient for Stress Oscillations & Shock Waves
shape *factor* Shape Factor for Unloading
fail *flag* Failure Option After Cutoff Stres is Reached
where *flag* can be :
0 Tensile Stress remains at Cut-off Value
1 Tensile Stress is reset to zero
bvflag *flag* Bulk Viscosity Activation Flag
where *flag* can be :
0 No Bulk Viscosity (Recommended)
1 Bulk Viscosity Active
kcon *coefficient* Stiffness Coefficient for Contact Interface Stiffness

Viscoelastic Option

Calculate the Viscoelastic Constants

lcid2 *load_curve* Load Curve if Constants beta-t are determined via Least Squares Fit
bstart *initial_value* Starting Beta
nt *number_of_terms* Number of Terms in the Fit
tramp *time* Optional Ramp Time for Loading

List the Viscoelastic Constants

g1 *maxwell_constant* Maxwell Constant G1
beta1 *decay_constant* Decay Constant beta1
g2 *maxwell_constant* Maxwell Constant G2
beta2 *decay_constant* Decay Constant beta2
g3 *maxwell_constant* Maxwell Constant G3
beta3 *decay_constant* Decay Constant beta3
g4 *maxwell_constant* Maxwell Constant G4
beta4 *decay_constant* Decay Constant beta4
g5 *maxwell_constant* Maxwell Constant G5
beta5 *decay_constant* Decay Constant beta5
g6 *maxwell_constant* Maxwell Constant G6
beta6 *decay_constant* Decay Constant beta6

For material type **75** (Bikhu/Dubois Foam Model)
e *young's_modulus*
ld1 *load* curve pressure for plastic yielding vs. volumetric strain

ld2 *load* curve uniaxial yield stress vs. volumetric strain
visc *viscous_coefficient*
pcut *pressure_cutoff*
vpcut *cutoff*
tcut *cutoff*
vtcut *cutoff*
lcrate *load_curve*
pcoef *coeff*

For material type **76** (General Viscoelastic)

k *bulk_modulus* Constant Elastic Bulk Modulus
pcf
ef *flag*

Viscoelastic Option

Calculate the Viscoelastic Constants

lcid2 *load_curve* Load Curve if Constants beta-t are determined via Least Squares Fit
bstart *initial_value* Starting Beta
nt *number_of_terms* Number of Terms in the Fit
tramp *time* Optional Ramp Time for Loading

List the Viscoelastic Constants

g1 *maxwell_constant* Maxwell Constant G1
beta1 *decay_constant* Decay Constant beta1
g2 *maxwell_constant* Maxwell Constant G2
beta2 *decay_constant* Decay Constant beta2
g3 *maxwell_constant* Maxwell Constant G3
beta3 *decay_constant* Decay Constant beta3
g4 *maxwell_constant* Maxwell Constant G4
beta4 *decay_constant* Decay Constant beta4
g5 *maxwell_constant* Maxwell Constant G5
beta5 *decay_constant* Decay Constant beta5
g6 *maxwell_constant* Maxwell Constant G6
beta6 *decay_constant* Decay Constant beta6

Volumetric Relaxation Option

Calculate the Volumetric Relaxation

lcidk *load_curve* Load Curve if Constants beta-t are determined via Least Squares Fit
bstartk *initial_value* Volumetric Starting Beta
ntk *number_of_terms* Volumetric Number of Terms in the Fit
tramp *time* Volumetric Ramp Time for Loading

List the Viscoelastic Constants

k1 *maxwell_constant* Maxwell Constant K1
betak1 *decay_constant* Decay Constant beta-k1
k2 *maxwell_constant* Maxwell Constant K2
betak2 *decay_constant* Decay Constant beta-k2
k3 *maxwell_constant* Maxwell Constant K3
betak3 *decay_constant* Decay Constant beta-k3
k4 *maxwell_constant* Maxwell Constant K4
betak4 *decay_constant* Decay Constant beta-k4
k5 *maxwell_constant* Maxwell Constant K5
betak5 *decay_constant* Decay Constant beta-k5
k6 *maxwell_constant* Maxwell Constant K6
betak6 *decay_constant* Decay Constant beta-k6

For material type **77** (Hyperviscoelastic Rubber)

pr *poisson's_ratio* Poisson's Ratio
formf *flag* Formulation Flag
where *flag* can be :
0 Strain Energy Functional
n *order* Order of Fit To Experimental Data
nv *number_of_terms* Number of Terms in Fit
lcid2 *load_curve* Bt Least Square Fit Load Curve
sgl *specimen_gauge_length* Specimen Gauge Length
sw *specimen_width* Specimen Width
st *specimen_thickness* Specimen Thickness
lcid1 *load_curve* Force Versus Actual Change Load Curve
tramp *time* Ramp Time for Loading
c01 *coefficient* General Hyperelastic Coefficient C01
c11 *coefficient* General Hyperelastic Coefficient C11
c20 *coefficient* General Hyperelastic Coefficient C20
c02 *coefficient* General Hyperelastic Coefficient C02
g1 *maxwell_constant* Maxwell Constant G1
beta1 *decay_constant* Decay Constant beta1
g2 *maxwell_constant* Maxwell Constant G2
beta2 *decay_constant* Decay Constant beta2

g3 *maxwell_constant* Maxwell Constant G3
beta3 *decay_constant* Decay Constant beta3
g4 *maxwell_constant* Maxwell Constant G4
beta4 *decay_constant* Decay Constant beta4
g5 *maxwell_constant* Maxwell Constant G5
beta5 *decay_constant* Decay Constant beta5
g6 *maxwell_constant* Maxwell Constant G6
beta6 *decay_constant* Decay Constant beta6
lcid2 *load_curve* Bt Least Square Fit Load Curve
sgl *specimen_gauge_length* Specimen Gauge Length
sw *specimen_width* Specimen Width
st *specimen_thickness* Specimen Thickness
data 1 Biaxial Data
mu1 *coefficient* Ogden Coefficient Mu1
mu2 *coefficient* Ogden Coefficient Mu2
mu3 *coefficient* Ogden Coefficient Mu3
mu4 *coefficient* Ogden Coefficient Mu4
mu5 *coefficient* Ogden Coefficient Mu5
mu6 *coefficient* Ogden Coefficient Mu6
mu7 *coefficient* Ogden Coefficient Mu7
mu8 *coefficient* Ogden Coefficient Mu8
alpha1 *coefficient* Ogden Coefficient Alpha1
alpha2 *coefficient* Ogden Coefficient Alpha2
alpha3 *coefficient* Ogden Coefficient Alpha3
alpha4 *coefficient* Ogden Coefficient Alpha4
alpha5 *coefficient* Ogden Coefficient Alpha5
alpha6 *coefficient* Ogden Coefficient Alpha6
alpha7 *coefficient* Ogden Coefficient Alpha7
alpha8 *coefficient* Ogden Coefficient Alpha8
g *shear_modulus*
sigf s*tress_limit*
bstart *value*

For material type **78** (Solid/Concrete)

g *shear_modulus* Shear Modulus
k *bulk_modulus* Bulk Modulus
lcpv *load_curve* Pressure/Volumetric Strain Load Curve
lcfp *load_curve* Plastic Strain/Pressure Load Curve
lcrp *load_curve* Plastic Strain/Residual Strength Load Curve
pc *pressure_cutoff* Pressure Cutoff for Plastic Strain
out *flag* Plastic Strain Output
where *flag* can be

0 for VolumetricPlastic Strain
1 for DeviatoricPlastic Strain

b *factor* Residual Strength Factor After Cracking

fail *flag* Failure of Element Flag

where *flag* can be

0 for No Failure
1 for After cut-off is reached, Element is Eroded
2 for After cut-off is reached, Tension is no longer carried

Choose only one:

lcvm *load_curve* Yield vs. Von Mises pressure Load Curve
lcyp *load_curve* Second Stress Invariant, J2, Yield/Pressure Load Curve

For material type **79** (Hysteretic Soil)

k0 *bulk_modulus* Bulk Modulus at the Reference Pressure
p0 *pressure_cutoff* Cutoff/Datum Pressure
b *exponent* Exponent to Pressure-Sensitive Moduli
a0 *constant* Yield Function Constant a0
a1 *constant* Yield Function Constant a1
a2 *constant* Yield Function Constant a2
df *flag* Damping Factor

where *flag* can be:

0 No Damping
1 Maximum Damping

rp *pressure* Reference Pressure
gam1 *strain* Shear Strain Gamma1
gam2 *strain* Shear Strain Gamma2
gam3 *strain* Shear Strain Gamma3
gam4 *strain* Shear Strain Gamma4
gam5 *strain* Shear Strain Gamma5
tau1 *stress* Shear Stress Tau1
tau2 *stress* Shear Stress Tau2
tau3 *stress* Shear Stress Tau3
tau4 *stress* Shear Stress Tau4
tau5 *stress* Shear Stress Tau5
lcid *load_curve* Load Curve Defining Shear Strain versus Shear Stress
sfld *factor* Scale Factor to Apply to Shear Stress in LCID

For material type **80** (Ramberg-Osgood Plasticity)

gy *strain* Reference Shear Strain
ty *stress* Reference Shear Stress
a *coefficient* Stress Coefficient

r *exponent* Stress Exponent
k *elastic_modulus* Elastic Bulk Modulus

For material type **81** (Plastic with Damage)
e *young's_modulus* Young's Modulus
cc *parameter* Strain Rate Parameter C
p *parameter* Strain Rate Parameter p
pr *poisson's_ratio* Poisson's Ratio
sigy *yield_stress* Yield Stress
eppf *strain* Plastic Strain at Failure
tdel *time_step* Automatic Element Deletion Time Step
lcsr *load_curve* Load Curve to Scale Yield Stress
eppfr *strain* Plastic Strain at Rupture
vp
lcdm *load_curve*
numint *num_fail*
Effective Stress/Effective Plastic Strain
by Load Curve
lcss *load_curve* Effective Stress/Effective Plastic Strain Load Curve
etan *tangent_modulus* Tangent Modulus
by Strain & Stress Values (up to 8 values)
eps *list_of_effective_plastic_strains* Effective Plastic Strain Values
es *list_of_effective_plastic_stress* Yield Stress Values

For material type **83** (Fu-Chang"s Foam with Rate Effects)
e *young's_modulus* Young's Modulus for Tensile Strains
ed *coefficient* Stiffness Coefficient for Contact Interface Stiffness
tc *stress* Tension Cut-off Stress
fail *flag* Failure Option After Cutoff Stress is Reached
where *flag* can be
0 Tensile Stress remains in Cut-off Value
1 Bulk Viscosity Active

damp *coefficient* Viscous Coefficient
tbid *load_curve* Stress-Strain/Strain Rate Table
bvflag *flag* Bulk Viscosity Activation Flag
where *flag* can be :
0 No Bulk Viscosity (Recommended)
1 Bulk Viscosity Active
d0 *constant* Material Constant d0
n0 *constant* Material Constant n0
n1 *constant* Material Constant n1

n2 *constant* Material Constant n2
n3 *constant* Material Constant n3
c0 *constant* Material Constant c0
c1 *constant* Material Constant c1
c2 *constant* Material Constant c2
c3 *constant* Material Constant c3
c4 *constant* Material Constant c4
c5 *constant* Material Constant c5
aij *constant* Material Constant aij
sij *constant* Material Constant sij
ratemin *strain_rate* Minimum Strain Rate
ratemax *strain_rate* Maximum Strain Rate
sflag *flag*
rflag *flag*
tflag *flag*
pvid *load_curve*
sraf *flag*

For material type **86** (Orthotropic-Viscoelastic)

ea *constant* Ea
eb *constant* Eb
ec *constant* Ec
vf *friction* Volume Friction of Viscoelastic Material
k *bulk_modulus* Elastic Bulk Modulus, K
g0 *shear_modulus* Short-Time Shear Modulus, G0
ginf *shear_modulus* Long-Time Shear Modulus, Ginf
b *decay_constant* Decay Constant, Beta
prba *poisson's_ratio* Vba, Poisson's ratio
prca *poisson's_ratio* Vca, Poisson's ratio
prcb *poisson's_ratio* Vcb, Poisson's ratio
gab *constant* Gab
gbc *constant* Gbc
gca *constant* Gca
aopt *option*

where the option can be one of

0 for by nodes
1 for by point and element center
2 for by normal vectors
3 by cross product with shell normal(Shell elements only)

xp *x-coordinate* aopt 1
yp *y-coordinate* aopt 1
zp *z-coordinate* aopt 1

ax *x-component* aopt 2
ay *y-component* aopt 2
az *z-component* aopt 2
dx *x-component* aopt 2
dy *y-component* aopt 2
dz *z-component* aopt 2
vx *x-component* aopt 3
vy *y-component* aopt 3
vz *z-component* aopt 3
beta *beta* aopt 3 Material angle

For material type **87** (Cellular Rubber)

pr *poisson's_ratio* Poisson's Ratio
p0 *pressure* Initial Air Pressure
phi *ratio* Ratio of Cellur Rubber to Rubber Density
ivs *strain* Initial Volumetric Strain
g *modulus* Optional Shear Relaxation Modulus for Rate Effects
b *constant* Optional Decay Constant
Computed Viscoelastic Parameters
n *order* Order of Fit
sgl *specimen_gauge_length* Specimen Gauge Length
sw *specimen_width* Specimen Width
st *specimen_thickness* Specimen Thickness
lcid *load_curve* Force/Actual Change in Guage Length Load Curve
n 0 Viscoelastic Parameters
c01 *coefficient* General Hyperelastic Coefficient C01
c11 *coefficient* General Hyperelastic Coefficient C11
c20 *coefficient* General Hyperelastic Coefficient C20
c02 *coefficient* General Hyperelastic Coefficient C02

For material type **88** (Mechanical Threshold Stress)

siga *sigma* Dislocation Interactions with Long Range Barriers (force/area)
sigi *sigma* Dislocation Interactions with Interstitial Atoms (force/area)
sigs *sigma* Dislocation Interactions with Solute Atoms (force/area)
sig0 *sigma* Initial Value of Sigma at Zero Plastic Strain (force/area)
hf0 *constant* Dislocation Generation Material Constant (force/area) HF0
hf1 *constant* Dislocation Generation Material Constant (force/area) HF1
hf2 *constant* Dislocation Generation Material Constant (force/area) HF2
sigs0 *sigma* Saturation Threshold Stress at 0 K (force/area)
edots0 *strain_rate* Reference Strain Rate EDOTS0 (1/time)
burg *magnitude* Magnitude of Burger's Vector (interatomic slip distance), (distance)

capa *constant* Material Constant CAPA
boltz *constant* Boltzmann's Constant (energy/degree)
sm0 *shear_modulus* Shear Modulus at Zero Degrees Kelvin
sm1 *shear_modulus* Shear Modulus Constant (force/area) SM1
sm2 *shear_modulus* Shear Modulus Constant (force/area) SM2
edot0 *strain_rate* Reference Strain-Rate EDOT0 (1/time)
g0 *energy* Normalized Activation Energy for Dislocation/Dislocation interaction
pinv *constant* Material Constant PINV
qinv *constant* Material Constant QINV
edoti *strain_rate* Reference Strain Rate EDOTI (1/time)
g0i *energy* Normalized Activation Energy for Dislocation/Interstitial Interaction
pinvi *constant* Material Constant PINVI
qinvi *constant* Material Constant QINVI
edots *strain_rate* Reference Strain-Rate EDOTS (1/time)
g0s *energy* Normalized Activation Energy for Dislocation/Solute Interaction
pinvs *constant* Material Constant PINVS
qinvs *constant* Material Constant QINVS
rhocpr *product* Product of Density and Specific Heat
temprf *initial_temperature* Initial Element Temperature in Degrees K
bulk *bulk_modulus* Bulk Modulus for Shell Elements
alpha *constant* Material Constant ALPHA
eps0 *factor* Factor to Normalize Strain Rate

For material type **90** (Acoustic)
ss *speed* Sound Speed
b *factor* Damping Factor
cf *flag* Cavitation Flag
where *flag* can be
0 Off
1 On
atmos *pressure* Atmospheric Pressure
grav *constant* Gravitational Acceleration Constant
fsp *x_coord y_coord z_coord* Coordinates of Free Surface point
fsn *x_dir_cos y_dir_cos z_dir_cos* Direction Cosines of Free Surface Normal Vector

For material type **96** (Brittle Damage)
e *young's_modulus* Young's Modulus
frarf *friction* Friction of Reinforcement in Section
erf *young's_modulus* Young's Modulus of Reinforcement
ysrf *yield_stress* Yield Stress of Reinforcement

ehrf *hardening_modulus* Hardening Modulus of Reinforcement
fsrf *failure_strain* True Failure Strain of Reinforcement
pr *poisson's_ratio* Poisson's Ratio
tlimit *tensile_limit* Tensile Limit
slimit *shear limit* Shear Limit
ftough *fracture_toughness* Fracture Toughness
sreten *shear_retention* Shear Retention
visc *viscosity* Viscosity

For material type **100 beam elfom spw** (Spot Weld)
e *young's_modulus* Young's modulus
pr *poisson's_ratio* Poisson's ratio
es *yield_stress* Yield Stress
etan *hardening_modulus* Hardening modulus
tsms *time_step_size* Time step size for mass scaling
fs *failure_strain* Failure strain for eroding elements
nrrf *resultant_at_failure* Force resultant Nrrf at failure
nrsf *resultant_at_failure* Force resultant Nrsf at failure
nrtf *resultant_at_failure* Force resultant Nrtf at failure
mrrf *resultant_at_failure* Moment resultant Mrrf at failure
mssf *resultant_at_failure* Moment resultant Mssf at failure
trrf *resultant_at_failure* Moment resultant Trrf at failure

For material type **103** (Anisotropic Viscoplastic)
e Young's Modulus
pr Poisson's Ratio
sigy Initial Yeild Stress
Choose a Method
flag 0 List all Material Parameters
qr1 *parameter* Isotropic Hardening Paameter, Qr1
cr1 *parameter* Isotropic Hardening Paameter, Cr1
qr2 *parameter* Isotropic Hardening Paameter, Qr2
cr1 *parameter* Isotropic Hardening Paameter, Cr2
qx1 *parameter* Kinematic Hardening Paameter, Qx1
cx1 *parameter* Kinematic Hardening Paameter, Cx1
qx2 *parameter* Kinematic Hardening Paameter, Qx2
cx2 *parameter* Kinematic Hardening Paameter, Cx2
vk *parameter* Viscous Material Parameter Vk
vm *parameter* Viscous Material Parameter Vm
flag 1 Load curve and Optionally the Viscous Parameters
lcss *load_curve* Load Curve Number

alpha *alpha* Distribution of Hardening Used in the Curve-Fitting
vk *parameter* Viscous Material Parameter Vk
vm *parameter* Viscous Material Parameter Vm
flag 1 Load Table (automatically calculates the Viscous Parameters)
lcss *load_table* Load Table Number
alpha *alpha* Distribution of Hardening Used in the Curve-Fitting
Choose the Same Element Type
R00 for Shell
R45 for Shell
R90 for Shell
F for Brick
G for Brickh
H for Brick
L for Brick
M for Brick
N for Brick
aopt *option*
where the *option* can be one of

0	by nodes
1	by point and element center
2	by normal vectors
3	by cross product with shell normal(Shell elements only)
4	by normal vectors in cylindrical coordinates

xp *x-coordinate*	aopt 1 & 4
yp *y-coordinate*	aopt 1 & 4
zp *z-coordinate*	aopt 1 & 4
ax *x-component*	aopt 2
ay *y-component*	aopt 2
az *z-component*	aopt 2
dx *x-component*	aopt 2
dy *y-component*	aopt 2
dz *z-component*	aopt 2
vx *x-component*	aopt 3 & 4
vy *y-component*	aopt 3 & 4
vz *z-component*	aopt 3 & 4
p1 *x-component*	aopt 4
p2 *y-component*	aopt 4
p3 *z-component*	aopt 4

beta angle
betal list_angle ;
plas
s11 *stress* for plas

s22 *stress* for plas
s33 *stress* for plas
s12 *stress* for plas

For material type **126** (Metallic Honeycomb)

e *young's_modulus* Young's Modulus (required)
pr *poisson's_ratio* Poisson's Ratio (required)
sigy *yield_stress* Yield Stress for Fully Compacted Honeycomb (required)
lca *load_curve* Sigma-aa versus Normal Strain-aa Load Curve (required)
lcb *load_curve* Sigma-bb versus Normal Strain-bb Load Curve (default LCA)
lcc *load_curve* Sigma-cc versus Normal Strain-cc Load Curve (default LCA)
lcs *load_curve* Shear Stress versus Relative Volume/Volumetric Strain Load Curve (default LCA)
vf *relative_volume* Relative Volume at which the Honeycomb is Fully Compacted (required)
eaau *elastic_modulus* Elastic Modulus Eaau in Uncompressed Configuration (required)
ebbu *elastic_modulus* Elastic Modulus Ebbu in Uncompressed Configuration (required)
eccu *elastic_modulus* Elastic Modulus Eccu in Uncompressed Configuration (required)
gabu *elastic_modulus* Elastic Shear Modulus Gabu in Uncompressed Configuration (required)
gbcu *elastic_modulus* Elastic Shear Modulus Gbcu in Uncompressed Configuration (required)
gcau *elastic_modulus* Elastic Shear Modulus Gcau in Uncompressed Configuration (required)
mu *coefficient* Material Viscosity Coefficient (default .05)
bulk *flag* Bulk Viscosity Flag (default 0)

where *flag* can be :

0 No Bulk Viscosity (Recommended)
1 Bulk Viscosity Active

lcab *load_curve* Sigma-ab versus Shear Strain ab Load Curve (default LCS)
lcbc *load_curve* Sigma-bc versus Shear Strain bc Load Curve (default LCS)
lcca *load_curve* Sigma-ca versus Shear Strain ca Load Curve (default LCS)
lcsr *load_curve* Strain Rate Effects Load Curve (optional)
tsef *strain* Tensile Strain at Element Failure (element will erode)
ssef *strain* Shear Strain at Element Failure (element will erode)
aopt *option*

where the *option* can be one of

0 by nodes
1 by point and element center

2 by normal vectors
3 by cross product with shell normal(Shell elements only)

xp *x-coordinate* aopt 1
yp *y-coordinate* aopt 1
zp *z-coordinate* aopt 1
ax *x-component* aopt 2
ay *y-component* aopt 2
az *z-component* aopt 2
dx *x-component* aopt 2
dy *y-component* aopt 2
dz *z-component* aopt 2
vx *x-component* aopt 3
vy *y-component* aopt 3
vz *z-component* aopt 3

For material type **134** (Viscoelastic Fabric)
k Constant Elastic Bulk Modulus
cs Compressive Stress
Viscoelastic Option
Calculate Viscoelastic Constants
lcid2 *load_curve* Bt Least Square Fit Load Curve
bstart *initial_value* Starting Beta
nt *number_of_terms* Number of Terms in Fit
tramp *time* Ramp Time for Loading
List the Viscoelastic Constants
g1 *maxwell_constant* Maxwell Constant G1
beta1 *decay_constant* Decay Constant beta1
g2 *maxwell_constant* Maxwell Constant G2
beta2 *decay_constant* Decay Constant beta2
g3 *maxwell_constant* Maxwell Constant G3
beta3 *decay_constant* Decay Constant beta3
g4 *maxwell_constant* Maxwell Constant G4
beta4 *decay_constant* Decay Constant beta4
g5 *maxwell_constant* Maxwell Constant G5
beta5 *decay_constant* Decay Constant beta5
g6 *maxwell_constant* Maxwell Constant G6
beta6 *decay_constant* Decay Constant beta6
Volumetric Option
Calculate the Volumetric Relaxation
lcidk *load_curve* Volumetric Load Curve for Least Square Fit
bstartk *initial_value* Volumetric Starting Beta

ntk *number_of_terms* Number of Terms in Fit
trampk *time* Volumetric Ramp Time for Loading
List the Volumetric Relaxation Constants
k1 *maxwell_constant* Maxwell Constant K1
betak1 *decay_constant* Decay Constant betak1
k2 *maxwell_constant* Maxwell Constant K2
betak2 *decay_constant* Decay Constant betak2
k3 *maxwell_constant* Maxwell Constant K3
betak3 *decay_constant* Decay Constant betak3
k4 *maxwell_constant* Maxwell Constant K4
betak4 *decay_constant* Decay Constant betak4
k5 *maxwell_constant* Maxwell Constant K5
betak5 *decay_constant* Decay Constant betak5
k6 *maxwell_constant* Maxwell Constant K6
betak6 *decay_constant* Decay Constant betak6

lsdyna select LS-DYNA output format

lsdyna *option*
where *option* can be
keyword keyword format
structured structed format (obsolete)

lsdyopts LS-DYNA analysis and database options

lsdyopts *options* ;
where *options* can be
nosu CONTROL_ACCURACY
inn
pidosu *set_id*
factin *initial_relaxation_factor* CONTROL_ADAPSTEP
dfactr *incremental_increase*
adpfreq *time* CONTROL_ADAPTIVE
adptol *tolerance*
adpopt *type*
where *type* can be
1 - angle change in degrees per adaptive refinement relative to surrounding elements for each element to be refined.
2 - total angle change in degrees relative to the surrounding element for each element to be refined.

7 - 3D r-adaptive remeshing for solid elements.
8 - 2D r-adaptive remeshing for axisymmetric and plane strain solid elements
maxlvl *max_number_of_levels*
tbirth *time*
tdeath *time*
lcadp *load_curve*
gsam
mnelsz *size*
npss *number_of_passes*
ireflg *level*
adpene *distance*
adpth *thickness*
imem *percentage*
orient
maxel *number_of_elements*
dct *type* CONTROL_ALE
where *type* can be
1 - lagrangian (default)
2 - eulerian
3 - arbitrary lagrangian eulerian
4 - eulerian ambient
nadv *cycles*
meth *method*
where *method* can be
1 - donor cell + half index shitft (first order accurate)
2 - van leer + half index shift (second order)
3 - van leer
afac *weight*
bfac *weight*
cfac *weight*
dfac *weight*
efac *weight*
tbeg *time*
tend *time*
aafac *factor*
vfact *factor*
vlimit *limit*
ebc *flag*
where *flag* can be
0 Off
1 On with stick condition
2 On with slip condition

q1 *quadratic_viscosity_coefficient* CONTROL_BULK_VISCOSITY
q2 *linear_viscosity_coefficient*
ibq *type*
where *type* can be
-1 standard (also additional shels)
1 standard (default)
itsflg *flag* CONTROL_CFD_AUTO
where *flag* can be
0 IAUTO=1 for fixed time step size
1 Fixed time step based on DTINIT
2 Time step based on CFL/stability for INSOL=3
3 Automatic time step selection
epsdt *tolerance*
dtsf *scale_factor*
adtmax *time_step_size*
insol *solver_type* CONTROL_CFD_GENERAL
dtinit *initial_time_step*
cfl *maximum_advective_grid-CFL*
ickdt *Reynolds_and_advective_CFL_check_interval*
iacurc *accuracy_flag*
mimass *mass_matrix_formula* CONTROL_CFD_MOMENTUM
where mass_matrix_formula can be
0 IMASS=1 (default)
1 Lumped mass matrix
2 Consistent mass matrix
3 Higher-order mass matrix
iadvec *balancing_tensor_diffusivity_flag*
where *balancing_tensor_diffusivity_flag* can be:
0 IADVEC=10 for forward-Euler with BTD (default)
-1 IADVEC=0 for forward-Euler without BTD
10 forward-Euler with BTD
40 fully-implicit with simplified trapezoid rule
ifct *advective_flux_limiting_advection_scheme_toggle*
where *advective_flux_limiting_advection_scheme_toggle* can be
0 IFCT=1 (default)
1 Advective flux limiting is on
-1 Advective flux limiting is off
divu *RMS_divergence_tolerance*
thetak *viscous_terms_time_weighting*
thetaa *advection_terms_time_weighting*
thetaf *body_forces_time_weighting*
msol *momentum_equations_solver_type*

where *momentum_equations_solver_type* can be:

0 MSOL=20 (default)

20 Jacobi preconditioned conjugate gradient method

30 Jacobi preconditioned conjugate gradient squared method (default when IADVEC=40)

maxit *maximum_number_of_iterations*

ichkit *convergence_check_interval*

idiag *diagnostic_information_output_toggle*

ihist *convergence_history_file_generation_toggle*

eps *convergence_criteria*

ihg *hourglass_stabilization_type*

where *hourglass_stabilization_type* can be:

0 IHG=1 (default)

1 LS-DYNA CFD viscous hourglass stabilization

2 γ-hourglass stabilization viscous form

ehg *hourglass_stabilization_multiplier*

ipsol *pressure_solver_type* CONTROL_CFD_PRESSURE

where the *pressure_solver_type* can be:

0 IPSOL=22 for serial, IPSOL=21 for MPP (default)

10 Sparse direct solver

11 PVS direct solver

20 Jacobi preconditioned conjugate gradient method

21 SSOR preconditioned conjugate gradient method

22 SSOR preconditioned conjugate gradient using the Eisenstat transformation

maxitr *maximum_number_of_pressure_solver_iterations*

ichcit *convergence_criteria_check_interval*

idiag

ihst

epsp *convergence_criteria*

nvec *number_of_A-conjugate_vectors*

istab *stabilization_type*

where *stabilization_type* can be:

0 ISTAB=1

1 Local jump stabilization

2 Global jump stabilization

-1 No stabilization is active

pbeta *stabilization_parameter*

sid *set_id*

plev *hydrostatic_pressure_level*

plcid *hydrostatic_pressure_load_curve*

itemp *energy_equation_solver_type* CONTROL_CFD_TRANSPORT

nspec *number_of_species_transport_equations_activated*

imss *mass_matrix_formulation*
ibaltd *balancing_tensor_diffusivity_flag*
iaflx *advective_flux_limiting_flag*
thetk *viscous/diffusion_weighting_term*
thtaa *advection_term_time_weighting*
thetf *body_forces_time_weighting*
itsol *equation_solver_type*
mxiter *maximum_number_of_iterations*
ickint *convergence_criteria_check_interval*
idiagn *diagnostic_information_output_flag*
ichist *convergence_history_file_generation_flag*
epst *convergence_criteria*
ihgt *hourglass_stabilization_type*
ehgt *stabilization_parameter*
itrb *turbulence_model_flag* CFD_TURBULENCE
smagc *Smagorinsky_constant*
sn1-sn8 *optional_seed_nodes*
icoarse *coarsening_toggle* (CONTROL_COARSEN)
fangl *allowable_flatness_angle*
sn1-sn8 *optional_seed_nodes*
slsfac *sliding_interface_penalties_scale_factor* CONTROL_CONTACT
rwpnal *rigid_wall_penalties_scale_factor*
islchk *initial_penetration_check*
shlthk *shell_thickness*
penopt *penalty_stiffness_value_option*
thkchg *shell_thickness_changes*
orien *contact_interface_segment_reorientation_flag*
dkeep *flag*
usrstr *storage_per_contact_interface*
usrfrc *storage_per_contact_interface*
nsbcs *number_of_cylces*
interm *intermittent_searching_flag*
xpene *multiplier*
tfst *actual_shell_thickness_flag*
itftss *time_step_size_flag*
itfpsn *bypass_projection_flag*
dsfric *default_static_coefficient_of_friction*
ddfric *default_dynamic_coefficient_of_friction*
dedc *default_exponential_decay_coefficient*
dvfc *default_viscous_friction_coefficient*
dth *default_contact_thickness*
dthsf *thickness_scale_factor*

dpensf *default_local_penalty_scale_factor*
ignore *initial_penetrations_flag*
frceng *frictional_energy_flag*
unleng *factor* CONTROL_COUPLING
untime *factor*
unforc *factor*
timidl *time*
flipx *flag* (**0**-off**, 1**-on)
flipy *flag* (**0**-off**, 1**-on)
flipz *flag* (**0**-off**, 1**-on)
sybcyl *interval*
cputim *seconds* CONTROL_CPU
nrcyck *iterations* CONTROL_DYNAMIC_RELAXATION
drtol *tolerance*
drfctr *factor*
drterm *time*
tssfdr *factor*
irelal *flag* (**0**-off, **1**-on)
edttl *tolerance*
idrflg *flag*
where *flag* can be
1 - activate dynamic relaxation
2 - initialize to a prescribed geometry
hgen *flag* (**1**-off, **2**-on) CONTROL_ENERGY
rwen *flag* (**1**-off, **2**-on)
slnten *flag* (**1**-off, **2**-on)
rylen *flag* (**1**-off, **2**-on)
expsh CONTROL_EXPLOSIVE_SHADOW
ihq *type* CONTROL_HOURGLASS
where *type* can be
1 - standard ls-dyna3d
2 - flanagan-belytschko integration
3 - flanagan-belytschko integration with exact volume
4 - stiffness form of type 2 (flanagan-belytschko)
5 - stiffness form of type 3 (flanagan-belytschko)
6 - Belytschko-Bindeman [1993]
8 - Applicable to the type 16 fully integrated shell element
qh *coefficient*
n36flg
iautf *time_step_control_flag* CONTROL_IMPLICIT_AUTO
where *time_step_control_flag* can be:
0 - constant time step size

1 - automatically adjusted time step size

iteropt *optimum_iteration_count*

iterwin *allowable_iteration_window*

dtmini *minimum_allowable_time_step*

dtmaxi *maximum_allowable_time_step*

inal *flag* CONTROL_IMPLICIT_DYNAMICS

where *flag* can be:

0 static,

1 dynamic, Newmark

2 dynamic, modal

newgam *constant*

newbet *constant*

neig *number_of_eigenvalues* CONTROL_IMPLICIT_EIGENVALUE

center *frequency*

lflag *flag* (**0**-left end point is -infinity, **1**-left end point is lftend)

lftend endpoint

rflag *flag* (**0**-right end point is infinity, **1**-right end point is rhtend)

rhtend endpoint

eigmth flag (**1**-subspace iteration, **2**-block shift and Lanczos)

shfscl *scale*

imflag *flag* CONTROL_IMPLICIT_GENERAL

where flag can be:

0 explicit

1 implicit

2 explicit & implicit

dt0 *initial_implicit_time_step*

imform *flag* (**1**-fully integrated formulation, **2**-original formulation)

nsbs *number_of_springback_steps*

istress *flag* (**1**-include initial stress, **2**-ignore initial stress)

cnstn *indicator_for_consistent_tangent_stiffness* (**0**-do not use, **1**-use)

form *element_formulation* (**0**-type 16, **1**-type 6)

nsolvr *flag* CONTROL_IMPLICIT_SOLUTION

where *flag* can be

1 - linear

2 - nonlinear with BFGS updates (default)

3 - nonlinear with Broyden updates

4 - nonlinear with DFP updates

5 - nonlinear with Davidon updates

6 - nonlinear with BFGS updates + arclength

7 - nonlinear with Broyden updates + arclength

8 - nonlinear with DFP updates + arclength

9 - nonlinear with Davidon updates + arclength

ilimit *iteration_limit*
maxref *stiffness_reformation_limit*
dctoln *displacement_convergence_tolerance*
ectoln *energy_convergence_tolerance*
lstoln *line_search_convergence_tolerance*
dnorm *flag* (**1**-vs. current time step, **2**-vs. total)
diverg *flag* (**1**-reform stiffness, **2**-ignore divergence)
istif *flag* (**1**-reform stiffness at start of each step, **n**-reform at each "n"th step)
nlprt *flag* (**1**-print to screen and files, **2**-print to files)
arcctl *arc_length_control_node*
arcdir *flag* (**1**-global X-translation, **2**-global Y-translation, **3**-global Z-translation)
arclen *arc_length_size*
arcmth *flag* (**1**-Crisfield, **2**-Ramm)
arcdmp *flag* (**2**-off, **1**-on)
lsolvr *flag* CONTROL_IMPLICIT_SOLVER

where *flag* can be

1 - direct, sparse, incore (default)
3 - direct, sparse, double precision
4 - SMP parallel multi-frontal sparse solver #2
5 - SMP parallel multi-frontal sparse solver #2, double precision
6 - BCSLIB-EXT, direct, sparse, double precision
10 - iterative, best of currently available
11 - iterative, Conjugate Gradient method
12 - iterative, CG, Jacobi preconditioner
13 - iterative, CG, Choleski preconditioner
14 - iterative, Lanczos method
15 - iterative, Lanczos, Jacobi preconditioner
16 - iterative, Lanczos, Choleski preconditioner

lprint *flag*

where flag can be:

0 - no printing
1 - summary statistics on memory, cpu time, and iteration count
2 - more statistics
3 - even more statistics

negev *flag* (**1**-stop on negative eigenvalue, **2**-print warning, continue)
sorder *option* (**0**-method set automatically, **1**-MMD, **2**-Metis)
drcm *method* (**1**-add stiffness, **2**-generate geometry based drilling constraint, **3**-neither)
drcprm *constraint_parameter*
autospc *switch* (**1**-automatically scan, **2**-do not scan)
autotol *tolerance*
ias *flag* (**1**-active, **2**-inactive) CONTROL_IMPLICIT_STABILIZATION
ascale *stabilization_scale_factor*

strttim *stabilization_start_time*
endtime *stabilization_end_time*
nmem *percentage* CONTROL_NONLOCAL
npopt *flag* CONTROL_OUTPUT
where *flag* can be
0 - no suppression
1 - nodal coordinates, element connectivities, rigid wall definitions and initial velocities are not printed
neecho *flag*
where *flag* can be
0 - all data printed
1 - nodal printing is suppressed
2 - element printing is suppressed
3 - both node and element printing is suppressed
nrefup *flag* (**0**-no update, **1**-update)
iaccop *flag* (**0**-no average, **1**-averaged between output intervals)
ofifs *interval*
ipnint *flag* (**0**-print 100 elements with smallest time steps, **1**-governing time steps printed)
ikedit *interval*
iflush *number_of_time_steps*
iprtf *print_flag*
where *print_flag* can be:
0 - write part data into both MATSUM and RBDOUT
1 - write data into RBDOUT only
2 - write data into MATSUM file only
3 - do not write data into RBDOUT and MATSUM
ncpu *number_of_cpus* CONTROL_PARALL-EL
numrhs *number* (**0** - same as ncpu, **1** - write only one)
iconst *flag* (**1**-on, **2**-off, for a faster solution)
ipllacc *flag* (**0**-off, **1**-on)
remin *minimum_edge_length* CONTROL_REMESHING
remax *maximum_edge_length*
lmf *flag* (**0**-explicit penalty, **1**-implicit with Lagrange multipliers) CONTROL_RIGID
jntf *stiffness* (0-incremental update, 1-total formulation)
orthmd *flag* (**0**-true, 1-false)
partm *flag* (**0**-true, 1-false)
sparse *flag* (**0**-false, 1-true)
wrpang *degrees* CONTROL_SHELL
itrist *flag* (**0**-no sorting required, **1**-full sorting)
irnxx *option*
where *option* can be

-2 - unique nodal fibers
-1 - compute normals each cycle
0 - default set to -1
1 - compute on restarts
n - compute every n cycles

istupd *flag* (**0**-no change, **1**-membrane straining causes thickness change)
theory *theory*
where *theory* can be
1 - Hughes-Liu
2 - Belytschko-Tsay (default)
3 - bciz triangular shell
4 - c0 triangular shell
5 - Belytschko-Tsay membrane
6 - s/r Hughes Liu
7 - s/r co-rotational Hughes Liu
8 - Englemann-Whirley shell
9 - fully integrated Belytschko-Tsay membrane
10 - Belytschko-Wong-Chiang
11 - fast (co-rotational) Hughes-Liu
12 - plane stress (x-y plane)
13 - plane strain (x-y plane)
14 - axisymmetric solid (y-axis of symmetry) - area weighted
15 - axisymmetric solid (y-axis of symmetry) - volume weighted
16 - fully integrated shell element
17 - discrete Kirchhoff triangular shell (DKT)
18 - discrete Kirchhoff linear shell either quad or triangular
20 - C^0 linear shell element with drilling stiffness

bwc *option*
where *option* can be
1 - Belytschko-Wong-Chiang warping stiffness added
2 - Belytschko-Tsay (default)

miter *option*
where *option* can be
1 - iterative plasticity with 3 secant iterations (default)
2 - full iterative plasticity
3 - radial return noniterative plasticity

shproj *flag* (**0**-drill projection, **1**-full projection)
rotascl *scale_factor*
intgrd *rule* (**0**-Gauss integration, **1**-Lobatto integration)
lamsht *flag* (**0**-do not update shear corrections, **1**-activate laminated shell theory)
esort *flag* (**0**-no sorting required, **1**-full sorting) CONTROL_SOLID
ianprc *procedure* CONTROL_SOLUTION

where *procedure* can be:

0 - Structural analysis only

1 - thermal analysis only

2 - coupled structural thermal analysis

4 - incompressible/low-Mach CFD analysis only

5 - coupled incompressible fluid-structure interaction

ncbs *number_of_cycles* CONTROL_SPH

boxid *id*

sphdt *death_time*

idim *dimension* (**3** - 3D problems, **2** - 2D problems, **-2** - 2D axisymmetric)

struct CONTROL_STRUCTURED

iterm

subcyl CONTROL_SUBCYCLE

endtim *time* CONTROL_TERMINATION

endcyc *cycle*

dtmin *factor*

endeng *percent_change*

endmas *percent_change*

mxmrts *number_time_steps* CONTROL_THERMAL_NONLINEAR

ctolt *tolerance*

divcp *value*

atype *type* CONTROL_THERMAL_SOLVER

where *type* can be:

0 - steady state analysis

1 - transient analysis

ptype *type*

where *type* can be:

0 - linear problem

1 - nonlinear problem with material properties evaluated at gauss point temperature

2- nonlinear problem with material properties evaluated at element average temp.

thslvr *type*

where *type* can be:

1 - actol: symmetric direct solver

2 - dactol: nonsymmetric direct solver

3 - dscg: diagonal scaled conjugate gradient iterative (default)

4 - iccg: Incomplete Choleski conjugate gradient iterative

cgtol *tolerance*

gpt *number* (0-default is set to 8, 1-one point quadrature is used)

eqheat *value*

fwork *fraction*

sbc *constant*

ktst *time_step_control* CONTROL_THERMAL_TIMESTEP
where *time_step_control* can be:
0 - fixed time step
1 - variable time step
tipt *parameter* (0.0-set to 0.5 - Crank-Nicholson scheme, 1.0-fully implicit)
itst *timestep*
tmint *timestep*
tmaxt *timestep*
dtempt *temperature_change*
tscpt *parameter*
dtinit *time_step_size* CONTROL_TIMESTEP
scft *scale_factor*
isdo *flag*
where *flag* can be:
0 - characteristic length = area/(min. of longest side or longest diagonal)
1 - characteristic length = area/(longest diagonal)
2 - based on bar wave speed
3 - timestep size based on maximum eigenvalue
tslimt minimum_time_step
dt2ms *time_step_size*
lctm *load_curve_id*
erode *flag* (0-no, 1-yes)
ms1st (0-no, 1-yes)
dt2msf *scale_factor*
iddlc *load_curve_ID* (DAMPING_GLOBAL)
valdmp *system_damping_constant*
stx *scale_factor_on_global_x_translational_damping*
sty *scale_factor_on_global_y_translational_damping*
stz *scale_factor_on_global_z_translational_damping*
srx *scale_factor_on_global_x_rotational_damping*
sry *scale_factor_on_global_y_rotational_damping*
srz *scale_factor_on_global_z_rotational_damping*
dmpid *material_id* (DAMPING_PART_MASS)
lcid *load_curve_id*
sflc *scale_factor_for_load_curve*
lcflg *separate_global_scale_factors_flag*
idstf *part_ID* (DAMPING_PART_STIFFNESS)
drayl *Rayleigh_damping_coefficient*
cdamp *fraction_of_critical_damping* (DAMPING_RELATIVE)
frq *frequency*
pidrb *material_id_rigid_body*
psid *material_set_id*

sphmem *n* number of neighbors (SPH)

sphform *flag*

where *flag* can be

0 default formulation

1 renormalization approximation

2 symmetric formulation

3 symmetric reformulation

4 tensor formulation

5 fluid particle formulation

6 fluid particle w/ renormalization

sphstart *time*

sphmaxv *velocity*

sphcont *flag*

where *flag* can be

0 particle approximation is defined

1 particle approximation is not calculated

sphderiv *type*

where *type* can be

0 default

1 cubic root formula

sphini *flag*

where flag can be

0 bucket sort based algorithm

1 global computation

abstat *time_interval* Airbag statistics file creation timestep (DATA BASE)

abstatbn *flag* Flag for binary airbag statistics file

where the *flag* can be

0 ascii

1 binary

2 both

avsflt *time_interval* AVS database file creation timestep

avsfltbn *flag* Flag for binary AVS database file

where the *flag* can be

0 ascii

1 binary

2 both

bndout *time_interval* Boundary condition forces and energy file creation timestep

bndoutbn *flag* Flag for binary boundary condition forces and energy file

where the *flag* can be

0 ascii

1 binary

2 both

defgeo time_*interval* Deformed geometry file creation timestep

defgeobn *flag* Flag for binary deformed geometry file

where the *flag* can be

0 ascii

1 binary

2 both

deforc *time_interval* Discrete element file creation timestep

deforcbn *flag* Flag for binary discrete element file

where the *flag* can be

0 ascii

1 binary

2 both

elout *time_interval* Element data file creation timestep

eloutbn *flag* Flag for binary element data file

where the *flag* can be

0 ascii

1 binary

2 both

gceout *time_interval* Geometric contact entities file creation timestep

gceoutbn *flag* Flag for binary rigid geometric contact entities file

where the *flag* can be

0 ascii

1 binary

2 both

glstat *time_interval* Global data file creation timestep

glstatbn *flag* Flag for binary global data file

where the *flag* can be

0 ascii

1 binary

2 both

h3out *time_interval*

h3outbn *flag*

where the *flag* can be

0 ascii

1 binary

2 both

jntforc *time_interval* Joint force file creation timestep

jntforcbn *flag* Flag for binary joint force file

where the *flag* can be

0 ascii

1 binary

2 both

matsum *time_interval* Material energies file creation timestep

matsumbn *flag* Flag for binary material energies file

where the *flag* can be

0 ascii

1 binary

2 both

movie *time_interval* MOVIE database file creation timestep

moviebn *flag* Flag for binary MOVIE database file

where the *flag* can be

0 ascii

1 binary

2 both

mpgs *time_interval* MPGS database file creation timestep

mpgsbn *flag* Flag for binary MPGS database file

where the *flag* can be

0 ascii

1 binary

2 both

ncforc *time_interval* Nodal interface forces file creation timestep

ncforcbn *flag* Flag for binary nodal interface forces file

where the *flag* can be

0 ascii

1 binary

2 both

nodfor *time_interval* Nodal force groups file creation timestep

nodforbn *flag* Flag for binary nodal force groups file

where the *flag* can be

0 ascii

1 binary

2 both

nsid *nodal_set_id*

cid *local_coordinate system_id*

nodout *time_interval* Nodal point data file creation timestep

nodoutbn *flag* Flag for binary nodal point data file

where the *flag* can be

0 ascii

1 binary

2 both

rbdout *time_interval* Rigid body data file creation timestep

rbdoutbn *flag* Flag for binary rigid body data file

where the *flag* can be

0 ascii
1 binary
2 both

rcforc *time_interval* Resultant interface forces file creation timestep

rcforcbn *flag* Flag for binary for resultant interface forces file

where the *flag* can be
0 ascii
1 binary
2 both

rwforc *time_interval* Wall forces file creation timestep

rwforbn *flag* Flag for binary wall forces file

where the *flag* can be
0 ascii
1 binary
2 both

sbtout *time_interval* Seat belt file creation timestep

sbtoutbn *flag* Flag for binary seat belt file

where the *flag* can be
0 ascii
1 binary
2 both

secforc *time_interval* Cross section forces file creation timestep

secforcbn *flag* Flag for binary cross section forces file

where the *flag* can be
0 ascii
1 binary
2 both

sleout *time_interval* Sliding interface energy file creation timestep

sleoutbn *flag* Flag for binary sliding interface energy file

where the *flag* can be
0 ascii
1 binary
2 both

spcforc *time_interval* SPC reaction forces file creation timestep

spcforbn *flag* Flag for binary SPC reaction forces file

where the *flag* can be
0 ascii
1 binary
2 both

sphout *time_interval* SPH data file creation timestep

sphoutbn *flag* Flag for binary SPH data file

where the *flag* can be

0 ascii
1 binary
2 both

ssstat *time_interval* Subsystem data file creation timestep

ssstatbn *flag* Flag for binary subsystem data file

where the *flag* can be
0 ascii
1 binary
2 both

swforc *time_interval* Nodal constraint reaction forces file creation timestep

swforcbn *flag* Flag for binary nodal constraint reaction forces file

where the *flag* can be
0 ascii
1 binary
2 both

tprint *time_interval* Thermal output file creation timestep

tprintbn *flag* Flag for binary thermal output file creation timestep

where the *flag* can be
0 ascii
1 binary
2 both

trhist *time_interval* Tracer particle history file creation timestep

trhistbn *flag* Flag for binary tracer particle history file creation timestep

where the *flag* can be
0 ascii
1 binary
2 both

iflagadm *flag*

where the *flag* can be
0 do not write (default)
1 write to file "d3mnf"

m_units *flag*

where the *flag* can be
-1 kilogram
-2 gram
-3 megagram (metric ton)
-4 lbf*sec^2/in (psi-com/atible)
-5 slug
-6 pound_mass

l_units *flag*

where the *flag* can be

-1 meter
-2 centimeter
-3 milimeter
-4 inch
-5 foot

t_units *flag*

where the flag can be
-1 second
-2 milisecond
-3 minute
-4 hour

d3drlf *time options*

where *options* can be
nr *n* number of running restart file
ibinary *flag*
where *flag* can be
0 64 bit format
1 32 bit ieee

d3dump *time form* ;

where the *form* can be
nr *n* number of running restart file

d3mean *time form options* ;

where the *form* can be
lcdt *n* load_curve_id
nr *n* number of running restart file
where *options* can be
istats *type*
where *type* can be
0 do not collect any statistics
1 generate mean quantities
2 generate additional 3nd moments
3 generate additional higher-order moments
tstart *time*
iavg *interval*

d3part *time form options* ;

where the *form* can be
lcdt *n* load_curve_id
nr *n* number of running restart file
where *options* can be
beam *type*
where *type* can be
0 discrete spring/damper elements are added

1 no discrete spring/damper elements are added
2 similar to option 0

npltc *n* number of plots
psetit *id* part id
neiph *n* number of integration point variables for solid elements
neips *n* number of integration point variables for shell elements
maxint *n* number of shell integration points
strflg *flag*

where *flag* can be
0 dump strain tensors (off)
1 dump strain tensors (on)

sigflg *flag*

where *flag* can be
1 include stress tensors
2 exclude stress tensors

epsflg *flag*

where *flag* can be
1 include effective plastic strains
2 exclude effective plastic strains

rltflg *flag*

where *flag* can be
1 include stress resultants
2 exclude stress resultants

engflg *flag*

where *flag* can be
1 include internal energe and thickness
2 exclude internal energe and thickness

cmpflg *flag*

where *flag* can be
0 global
1 local

ieverp *flag*

where *flag* can be
0 more than one state can be on each plotfile
1 only one state can be on each plotfile

beamip *n* number of beam integration points
dcomp *flag*

where *flag* can be
1 off (default), no rigid body data compression
2 on, rigid body data compression active
3 off, option(1) but no nodal vel. and accel.
4 on, option(2) but no nodal vel. and accel.

shge *flag*

where *flag* can be

1 output shell hourglass energy (off)

2 output shell hourglass energy (on)

stssz *flag*

where *flag* can be

1 off (default)

2 on (output time step size)

3 output mass, added mass, or time step size

n3thdt *flag*

where *flag* can be

1 write material energy (off)

2 write material energy (on)

nintsld *n* number of solid element integration points

lsavs [*option flag*] ;

where *option* can be

0 node

1 brick

2 beam

3 shell

4 thick shell

where *flag* can be

0 table 1

1 table 2

3 table 3

lsmovie [*option flag*] ;

where *option* can be

0 node

1 brick

2 beam

3 shell

4 thick shell

where *flag* can be

0 table 1

1 table 2

3 table 3

lsmpgs [*option flag*] ;

where *option* can be

0 node

1 brick

2 beam

3 shell

4 thick shell

where *flag* can be

0 table 1

1 table 2

3 table 3

ssstatex *id* part set id

ibinary *flag*

where *flag* can be

0 64 bit format

1 32 bit ieee

d3plot *time form options* ;

where the *form* can be

lcdt *n* load_curve_id

nr *n* number of running restart file

where *options* can be

beam *type*

where *type* can be

0 discrete spring/damper elements are added

1 no discrete spring/damper elements are added

2 similar to option 0

npltc *n* number of plots

iform *flag*

where *flag* can be

0 LS-DYNA database format

1 ANSYS database format

2 both LS-DYNA and ANSYS database format

ibinary *flag*

where *flag* can be

0 64 bit format

1 32 bit ieee

neiph *n* number of integration point variable for solid elements

neips *n* number of integration point variable for shell elements

maxint *n* number of shell integration points

strflg *flag*

where *flag* can be

0 dump strain tensors (off)

1 dump strain tensors (on)

sigflg *flag*

where *flag* can be

1 include stress tensors

2 exclude stress tensors

epsflg *flag*

where *flag* can be

1	include effective plastic strains
2	exclude effective plastic strains

rltflg *flag*

where *flag* can be

1	include stress resultants
2	exclude stress resultants

engflg *flag*

where *flag* can be

1	include internal energe and thickness
2	exclude internal energe and thickness

cmpflg *flag*

where *flag* can be

0	global
1	local

ieverp *flag*

where *flag* can be

0	more than one state can be on each plotfile
1	only one state can be on each plotfile

beamip *n* number of beam integration points

dcomp *flag*

where *flag* can be

1	off (default), no rigid body data compression
2	on, rigid body data compression active
3	off, option(1) but no nodal vel. and accel.
4	on, option(2) but no nodal vel. and accel.

shge *flag*

where *flag* can be

1	output shell hourglass energy (off)
2	output shell hourglass energy (on)

stssz *flag*

where *flag* can be

1	off (default)
2	on (output time step size)
3	output mass, added mass, or time step size

n3thdt *flag*

where *flag* can be

1	write material energy (off)
2	write material energy (on)

nintsld *n* number of solid element integration points

lsavs [*option flag*] ;

where *option* can be

0 node
1 brick
2 beam
3 shell
4 thick shell
where *flag* can be
0 table 1
1 table 2
3 table 3

lsmovie [*option flag*] ;
where *option* can be
0 node
1 brick
2 beam
3 shell
4 thick shell
where *flag* can be
0 table 1
1 table 2
3 table 3

lsmpgs [*option flag*] ;
where *option* can be
0 node
1 brick
2 beam
3 shell
4 thick shell
where *flag* can be
0 table 1
1 table 2
3 table 3

ssstatex *id* part set id

d3thdt *time form options* ;
where the *form* can be
lcdt *n* load curve id
nr *n* number of running restart file
where *options* can be
iform flag
where flag can be
0 LS-DYNA database format
1 ANSYS database format
2 both LS-DYNA and ANSYS database format

ibinary *flag*

where *flag* can be

0 64 bit format

1 32 bit ieee

neiph *n* number of integration point variable for solid elements

neips *n* number of integration point variable for shell elements

strflg *flag*

where *flag* can be

0 dump strain tensors (off)

1 dump strain tensors (on)

sigflg *flag*

where *flag* can be

1 include stress tensors

2 exclude stress tensors

epsflg *flag*

where *flag* can be

1 include effective plastic strains

2 exclude effective plastic strains

rltflg *flag*

where *flag* can be

1 include stress resultants

2 exclude stress resultants

engflg *flag*

where *flag* can be

1 include internal energe and thickness

2 exclude internal energe and thickness

cmpflg *flag*

where *flag* can be

0 global

1 local

ieverp *flag*

where flag can be

0 more than one state can be on each plotfile

1 only one state can be on each plotfile

beamip *n* number of beam integration points

dcomp *flag*

where *flag* can be

1 off (default), no rigid body data compression

2 on, rigid body data compression active

3 off, option(1) but no nodal vel. and accel.

4 on, option(2) but no nodal vel. and accel.

shge *flag*

where *flag* can be

1 output shell hourglass energy (off)

2 output shell hourglass energy (on)

stssz *flag*

where *flag* can be

1 off (default)

2 on (output time step size)

3 output mass, added mass, or time step size

n3thdt *flag*

where *flag* can be

1 write material energy (off)

2 write material energy (on)

nintsld *n* number of solid element integration points

lsavs [*option flag*] ;

where *option* can be

0 node

1 brick

2 beam

3 shell

4 thick shell

where *flag* can be

0 table 1

1 table 2

3 table 3

lsmovie [*option flag*] ;

where *option* can be

0 node

1 brick

2 beam

3 shell

4 thick shell

where *flag* can be

0 table 1

1 table 2

3 table 3

lsmpgs [*option flag*] ;

where *option* can be

0 node

1 brick

2 beam

3 shell

4 thick shell

where *flag* can be
0 table 1
1 table 2
3 table 3

ssstatex *id* part set id

ibinary *flag*
where *flag* can be
0 64 bit format
1 32 bit ieee

runrsf *time options* ;
where *options* can be
nr *n* number of running restart file

intfor *time option* ;
where the *option* can be
lcdt *n* load curve id
nr *n* number of running restart file

xtfile [*nodal_set_id1 coordiante_system_id*] ;

d3crck *time option* ;
where the *option* can be
lcdt *n* load curve id
nr *n* number of running restart file

cplane *options* ;
where *options* can be
psid *id* part set id.
xct *x* x-coordiante of tail of normal vector
yct *y* y-coordiante of tail of normal vector
zct *z* z-coordiante of tail of normal vector
xch *x* x-coordiante of head of normal vector
ych *y* y-coordiante of head of normal vector
zch *z* z-coordiante of head of normal vector
xhev *x* x-coordiante of head of edge vector
yhev *y* y-coordiante of head of edge vector
zhev *z* z-coordiante of head of edge vector
lenl *l* length of edge, in l direction
lenm *l* length of edge, in m direction
id id rigid body or accelerometer id
type *flag*
where *flag* for local system type can be
0 rigid body
1 accelerometer

idhead *options* ;
where *options* can be

csid *id* cross section id
heading *descriptor* cross section descriptor

fsi *option* [*option1*];
where *option* can be
dt output interval
where *option1* can be
id *id* surface id
sid *id* set id
stype *flag*
where *flag* can be
0 part set
1 part
2 segment set

iflagspr *flag*
where *flag* for output element nodal force vector can be
0 off
1 output for deformable nodes
2 output for materials, subset for nike3d file

superpl *timestep*

anprc *flag* specify the thermal solver type
where the *flag* can be
0 structured analysis only
1 thermal analysis only
2 coupled structured thermal analysis

yldb *yield* (BRODE)

hiteb *height*

xb0 *x*

yb0 *y*

zb0 *z*

tb0 *time*

lcb1 *load_curve_#*

lcb2 *load_curve_#*

clb *factor*

ctb *factor*

cpb *factor*

gvst *acceleration dir material_list ; mass1 depth1 ...* ; (BODY LOADS)
where *dir* can be
x x-direction
y y-direction
z z-direction

xgrav *scale load_curve_#*

xgdr *load_curve_#*

ygrav *scale load_curve_#*
ygdr *load_curve_#*
zgrav *scale load_curve_#*
zgdr *load_curve_#*
xvel *scale load_curve_#*
xavdr *load_curve_#*
xavcr *x0 y0 z0*
yvel *scale load_curve_#*
yavdr *load_curve_#*
yavcr *x0 y0 z0*
zvel *scale load_curve_#*
zavdr *load_curve_#*
zavcr *x0 y0 z0*
usaco usa coupling
lsnwds *megawords* defines the memory size to use in mgawords (KEYWORD)
hgen *flag* calculating hourglass energy
where the *flag* can be
1 off
2 on,
mrpc *flag* material repositioning flag for madymo/m-cal3d coupling
where the *flag* can be
0 off
1 on, section 68 input data is read
2 no repositioning of nodes
spbk *options* ; (Spring back)
where an *option* can be
opt1 *type*
where *type* can be
Lsdyna
Nastran
Seamless
opt2 *thic*
where *thic* can be
thickness
nothickness
mts *material_list* ;
nshv *n*
ftype *file_type*
where the *file_type* can be
0 ascii
1 binary
2 ascii and binary

nd1 *node_# constrains* ;
nd2 *node_# constrains* ;
nd3 *node_# constrains* ;
nd4 *node_# constrains* ;

where a *constraint* can be

dx	constrain in the x-direction
dy	constrain in the y-direction
dz	constrain in the z-direction
rx	constrain in the x-rotation
ry	constrain in the y-rotation
rz	constrain in the z-rotation

icsc implicit coupling for springback calculations

lsdythmt LS-DYNA thermal material properties

lsdythmt *material_# material_type parameter_list* ;

where the *material_type* can be one of the types listed below and the *parameter_list* depending on *material_type* and
where the *material_#* associates these thermal properties with a numbered material.

For material type 1 Isotropic

tro *density*
tgrlc *load_curve_#*
tgmult *multiplier*
hc *capacity*
tc *conductivity*

For material type 2 Orthotropic

tro *density*
tgrlc *load_curve_#*
tgmult *multiplier*
hc *capacity*
k1 *conductivity*
k2 *conductivity*
k3 *conductivity*
aopt *option*

where the *option* can be one of

0	by nodes
1	by point and element center
2	by normal vectors

xp *x-coordinate* aopt 1

yp *y-coordinate* aopt 1
zp *z-coordinate* aopt 1
ax *x-component* aopt 2
ay *y-component* aopt 2
az *z-component* aopt 2
dx *x-component* aopt 2
dy *y-component* aopt 2
dz *z-component* aopt 2

For material type 3 Temperature Dependent Isotropic

tro *density*
tgrlc *load_curve_#*
tgmult *multiplier*
temp *list_temperatures*
hc *list_capacities*
k *list_conductivities*

For material type 4 temperature dependent orthotropic

tro *density*
tgrlc *load_curve_#*
tgmult *multiplier*
temp *list_temperatures*
hc *list_capacities*
k1 *list_conductivities*
k2 *list_conductivities*
k3 *list_conductivities*
aopt *option*

where the *option* can be one of

0 by nodes
1 by point and element center
2 by normal vectors

xp *x-coordinate* aopt 1
yp *y-coordinate* aopt 1
zp *z-coordinate* aopt 1
ax *x-component* aopt 2
ay *y-component* aopt 2
az *z-component* aopt 2
dx *x-component* aopt 2
dy *y-component* aopt 2
dz *z-component* aopt 2

For material type 5 Temperature Dependent Isotropic with Phase Change

tro *density*
tgrlc *load_curve_#*
tgmult *multiplier*
temp *list_temperatures*
hc *list_capacities*
k *list_conductivities*
solt *temperature*
liqt *temerature*
lh *latent_heat*

For material type 6 Temperature Dependent Isotropic by Load Curves
tro *density*
tgrlc *load_curve_#*
tgmult *multiplier*
hclc *load_curve_#*
tclc *load_curve_#*

offset add offset to numbered entities in the output

offset [*type offset*] ;
where *type* can be

nodes	node numbers
bricks	brick elements
shells	shell elements
beams	beam elements
tshells	thick shells
nsetoff	node sets
fsetoff	face sets if
esetoff	element sets
partoff	parts
lcrsyoff	local coordinate systems

Remarks

This command is useful if you want to create partial LS-DYNA input files. By applying the proper offsets to the different entities, then you can easily combine several files for LS-DYNA by simply concatenating the files or using the include feature in LS-DYNA. This command is found in the Elements menus.

si assign sliding interface to region (part phase)

si *region sliding_# type options*

where

sliding_# reference number for the interface

type **m** for master and **s** for slave

options this depends on the *type*.

If the *type* is **s**, then the *options* can be

[*normal_failure_stress_or_force shear_failure_stress_or_force*
[*exponent_for_normal_force exponent_for_shear_force*]]

fsf *Coulomb_friction_scale viscous_friction_scale*

where the first pair of parameters must be specified in order to specify the second pair of parameters.

If the *type* is **m**, then *options* can be

fsf *Coulomb_friction_scale viscous_friction_scale*

Remarks

This command, and its relative **sii**, specify that faces in the mesh are part of a sliding interface. You can use these commands to assign a shell or brick face to a sliding interface definition. In order to define the properties of the sliding interface, first use the command **sid**. **Sid** defines the properties of the sliding interface that you refer to in **si** and **sii**.

Surfaces from 3D solid brick elements have an obvious orientation pointing outward. However, this is not the case with sliding interfaces on 2D shell surfaces. You must provide information about how to orient them. That is the purpose of the **orpt** command.

During the node merging process using, using **stp** for example, **True*Grid*®** will not merge nodes on opposite sides of a sliding interface.

Use the merge phase command **co** with the **si** option to view the numbered sliding interfaces and their orientation.

si select nodes for the slave side (merge phase)

si *type interface_# boundary options ;*

where

type can be one of:

n *node_numbe*r	to select a single node
rt *x y z*	to select a node close to a Cartesian point
cy *rho theta z*	to select a node close to a cylindrical point

sp *rho theta phi* to select a node close to a spherical point
nset *name_of_set* to select an entire node set
fset *face_set* to select a face set

boundary can be one of

m master side of the interface
s slave side of the interface

options can be

fail *norm_failure_force shear_failure_force*
exp *norm_failure_exp shear_failure_exp*
fsf *coulomb_friction_scale viscous_friction_scale*

Remarks

The global properties of a sliding interface are defined using the **sid** command. Some interface types allow for nodes on the slave side. Most require face sets for both the master and slave sides. The dummy sliding interface type, which is used to control the merging without the side effect of causing a sliding interface definition in the output, allows for nodes on both the master and slave side.

sii assign sliding interfaces (part phase)

sii *progression sliding_# type options*

where

sliding_# sliding interface reference number
type **m** for master and **s** for slave
options this depends on the *type*.

If the *type* is **s**, then *options* can be

[normal_failure_stress_or_force shear_failure_stress_or_force
[exponent_for_normal_force exponent_for_shear_force]]
fsf Coulomb_friction_scale viscous_friction_scale
where the first pair of parameters must be specified in order to specify the second pair of parameters.

If the *type* is **m**, then *options* can be

fsf Coulomb_friction_scale viscous_friction_scale

sid sliding interface definition

sid *slide_#* **lsdsi** *type options* ;

where *type* can be:

1 for Sliding without penalties
p1 for Symmetric sliding with penalties

2 for Tied
3 for Sliding, impact, friction
a3 for Sliding, impact, friction, no segmentation orientation
4 for Single surface contact
5 for Discrete nodes impacting surface
a5 for Discrete nodes impacting surface, no segmentation orientation
6 for Discrete nodes tied to surface
7 for Shell edge tied to shell surface
8 for Nodes spot welded to surface
9 for Tiebreak interface
10 for One way treatment of sliding, impact, friction
a10 for One way treatment, no segmentation orientation
13 for Automatic single surface with beams and arbitrary orientations
a13 for Automatic single surface with beams and arbitrary orientations with extra search for airbag contact
14 for Surface to surface eroding contact
15 for Single surface eroding contact
16 for Node to surface eroding contact
17 for Surface to surface symmetric/asymmetric constraint method
18 for Node to surface constraint method (Taylor and Flanagan 1989)
19 for Rigid body to rigid body contact with arbitrary force/deflection curve
20 for Rigid nodes to rigid body contact with arbitrary force/deflection curve
21 for Rigid body to rigid body contact with arbitrary force/deflection curve (one way treatment)
22 for Single edge treatment for shell surface edge to edge treatment
23 for Simulated draw bead
24 for Automatic surface to surface tiebreak
25 for Automatic one way surface to surface tiebreak
34 for Automatic general
35 for Automatic general interior
36 for Force transducer constraint
37 for Force transducer penalty
38 for Forming node to surface
39 for Forming one way surface to surface
40 for Forming surface to surface
41 for Discrete nodes impacting surface w/ interference
42 for One way treatment of sliding, impact, friction w/ interference
43 for Spotweld
44 for Spotweld with torsion
45 for Sliding, impact, friction w/ interference
46 for Tiebreak nodes only
47 for Tied with failure

rebar for rebar in concrete 1D sliding

and an *option* can be:

lcrsgo *load_curve_#* for optional load curve defining the resisting stress vs. gap opening

isrch *flag* small penetration in contact search

where *flag* can be

0 for check is off

1 for check is on

2 for check is on, shortest diagonal used

visdam *percent* for viscous damping coefficient in percent of critical

kpf *flag* kinematic partition factor for constraint

where *flag* can be

0 for fully automatic treatment

1 for one way treatment with slave nodes constrained to master surface

-1 for one way treatment with master nodes constrained to slave surface

lcair *load_curve_#* load curve defining airbag thickness

penmax *penetration* maximum penetration

thkopt f*lag* thickness option

where *flag* can be

0 for default from the control cards

1 for thickness is not considered

2 for thickness is considered but rigid bodies are excluded

3 for thickness is considered including rigid bodies

4 for thickness effects are not included

lcfpb *load_curve_#* force vs. penetration behavior load curve

fcm *flag* force calculation method

where *flag* can be

1 for total normal force on surface vs. max. penetration of any node

2 for normal force on each node vs. penetration of node through the surface

3 for normal pressure vs. penetration of node into surface

4 for total normal force vs. max. soft penetration

unstf unloading stiffness

lcbcrf *load_curve_#* load curve giving the bending component of the retaining force

lcnflen *load_curve_#* load curve giving the normal force per unit draw bead length as a function of displacement

dbd *depth* draw bead depth

scllc *factor* scale factor for load curve

nitdb #*_iterations* number of integration points laong the draw bead

slvmat *material_list* ; automatic slave segment materials

sypl slave, do not include faces with normal boundary constraints

serin slave erosion/interior node option

sadjmat slave storage is allocated so that eroding contact can occur

scoufsf *factor* coulomb friction scale factor
svfsf *factor* viscous friction scale factor
snffs *stress_or_force* slave normal stress at failure
ssffs *stress_or_force* slave shear stress at failure
senf *exponent* slave exponent for normal force
sesf *exponent* slave exponent for shear force
mstmat *material_list* ; automatic master segment materials
mypl master, do not include faces with normal boundary constraints
merin master erosion/interior node option
madjmat master storage is allocated so that eroding contact can occur
mcoufsf *factor*
mvfsf *factor*
mnffs *stress_or_force* master normal stress at failure
msffs *stress_or_force* master shear stress at failure
menf *exponent* master exponent for normal force
mesf *exponent* master exponent for shear force
scoef *coefficient* static coefficient of friction
dcoef *coefficient* dynamic coefficient of friction
decay *coefficient* dynamic decay coefficient
incslv include slave side in printed and binary force interface file
incmst include master side in printed and binary force interface file
sfsps *factor* scale factor on default slave penalty stiffness
sfmps *factor* scale factor on default master penalty stiffness
vfcoef *coefficient* coefficient for viscous friction
thss *thickness* optional thickness for slave surface
thms *thickness* optional thickness for master surface
sthss *factor* scale factor for slave surface thickness
sthms *factor* scale factor for master surface thickness
btime *time* birth time
dtime *time* death time
softc *flag* soft constraint option

where *flag* can be:

0 for penalty formulation
1 for soft constraint formulation

ssoftc *factor* scale factor for constraint forces of soft constraint option
maxpcss *coordinate* maximum parametric coordinate in segment search
srchdp *depth* search depth in automatic contact
ncybs *cycles* number of cycles between bucket sorts
ncyup *cycles* number of cycles between contact force updates for penalty formulations
diskoc disable logic in thickness offset contact to avoid shooting nodes
concon *conductance* contact conductance

radcon *conductance* radiation conductance
gapcs *size* gap critical size
ctofst
atbo *args*

where *args* can be

1 for Slave nodes in contact and which come into contact will permanently stick. Tangential motion is inhibited.

2 normal_stress shear_stress for Tiebreak is active for nodes which are initially in contact. Until failure, tangential motion is inhibited.

3 normal_stress shear_stress for Same as 1st option but with failure after sticking

4 for Tiebreak is active for nodes which are initially in contact but tangential motion with frictional sliding is permitted.

5 plastic_stress load_curve for Tiebreak is active for nodes which are initially in contact. Damage is defined by a load curve.

6 distance for Tiebreak is active for brick and thick shell nodes which are initially in contact. damage is a linear function between points.

lcid1 *load_curve* load curve dynamic interface stiffness
lcid2 *load_curve* load curve transient interface stiffness
isym *option_#* symmetric plane option
i2d3d *option_#* segment searching option
sldthk *thickness* solid element thickness
sldstf *thickness* solid element stiffness
igap *option_#* flag implicit convergence behavior
ignore *option_#* ignore initial penetration in automatic interfaces
edge *distance* edge to edge penetration check
rbrad *radius*
rbstr *strength*
rbshr *modulus*
rbumax *strain*
rbexp *exponent*

Remarks

Sliding interfaces or contact surfaces are constructed in 3 steps. These steps can be done in any order.

1. define the properties
2. select the slave side
3. select the master side, if applicable

The **sid** command is used to define the properties. The **si** and **sii** commands are used in the part phase or the merge phase to select the nodes or faces that form the master and slave sides of the interface.

The options for LS-DYNA are large and unique so they have been singled out and fall under the **lsdsi** option of the **sid** command. Some of these sliding interfaces require nodes on the slave side while others require only a set of faces on the slave side. This definition is required so that the proper data can be written to the output file.

You can construct a node set or a face set. This will be written to the output file as a set. Then it is a simple matter to add the keyword command to the output file using a text editor to transform that set into a contact surface or sliding interface. This approach has the problem that nodes may be merged across the two sides because they are not defined as sliding interfaces.

When nodes are merged, nodes across a sliding interface will not be merged. When a merge command is first issued in the merge phase, a table is written listing the number of nodes and faces associated with each sliding interface.

The **dummy** type interface is actually used to avoid merging of nodes. A sliding interface of this type is not written to the output file.

The nodes and faces of a sliding interface or contact surface can be viewed in the merge phase using the **si** option of the **co** command.

If the output option has been selected prior to using the dialogue box to make a selection, only the options available to that output option will be displayed in the dialogue box.

Example

```
sid 13 lsdsi 9 lcrsgo 1 visdam .34 isrch 1 ; ; ;
```

sind shell user-defined integration rules

sind *rule* s_1 t_1 w_1 s_2 t_2 w_2 ... s_n t_n w_n ;

where

rule	is the integration rule number, 0 or 1. If 1, then all other arguments are ignored.
t_i w_i m_i	are local coordinate, integration weight, and material number

spd define the properties of a set of springs or dampers

spd *spring/damper_# option type parameters*

where *option* can be

dro *flag*

where flag can be:
0 for a linear spring/damper
1 for a torsional spring/damper

dmf *dynamic_factor*
tv *test_velocity*
cl *clearance*
fd *failure_deflection*
dlc *limit_compression*
dlt *limit_tension*

where *type* is the spring or damper's material model and
where *parameters* is a list of corresponding parameters, as in the following:

le *stiffness* for linear elastic
lv *damping* for linear viscous
iep *elastic tangent yield* for isotropic elastic
ne *ld_curve_#* for nonlinear elastic
nv *ld_curve_#* for nonlinear viscous
gn *loading_# unloading_# hardening tension compression* for general nonlinear
dhpt for a dashpot
mv for a three parameter maxwell viscoelastic
itc for a inelastic tension or compression only
se *elastic_value damping stress* for scalar elastic
mus *l0 vmax sv a fmax tl tv fpe lmax ksh* for muscle

where

l0 for initial muscle length
vmx for maximum CE shortening velocity
sv for scale factor for Vmax vs. activs state
a for activation level vs. time function
fmax for peak isometric force
tl for active tension vs. length function
tv for active tension vs. velocity function
fpe for force vs. length function
lmax for relative length
ksh for exponential rise constant

Remarks

A spring or damper is defined using either the **spdp** command forming a set of springs/dampers between two surfaces, or using the **spring** command to create a single spring at a time. In each case, the definition of a spring includes a reference to a material definition **spd** number.

spdp assigns a face to be half of a set of spring/damper pairs

spdp *region spring/damper_#* **M** *scale_factor options* ;
or
spdp *region spring/damper_#* **S** *options* ;

where

spring/damper_# is the user-defined set number
M or *S* indicates Master or Slave
scale_factor is a scaling factor for the set
options can be

dx to constrain spring in the x-direction,
dy to constrain spring in the y-direction,
dz to constrain spring in the z-direction,
rx to constrain spring about the x-axis,
ry to constrain spring about the y-axis,
rz to constrain spring about the z-axis,
orop *flag* orientation option,

where flag can be:
0 for spring/damper acts along the axis
1 for deflection/rotations are measured and force/moments applied along the following vector
2 for deflection/rotations are measured and force/moments applied along the projection of the spring/damper onto the plane with the following normal

prflg *flag* print flag,

where flag can be:
0 for forces are printed in DEFORC file
1 for forces are not printed in DEFORC file

ofsi *offset* initial offset,
xco *x-component* x-component of the orientation vector,
yco *y-component* y-component of the orientation vector, and/or
zco *z-component* z-component of the orientation vector.

Remarks

A face can be either a master or slave side to this set of springs. Then each node on the slave side is paired with a node on the master side to form a spring/damper pair. Use the **spd** command to define the properties of the spring/damper. The degrees of freedom to be coupled with the spring/damper are specified for all of the nodes within the specified region.

spring create/modify a spring (part phase)

spring *spring_# options ;*

where

spring_# is the number of the spring that you are creating, and

options can be:

sinc *increment* increment the spring number when replicating the part,

v1 *i j k* vertex of the part as the first node,

pm1 *pointmass_#* point mass as the first node,

pminc1 *increment* first point mass number when making part replications,

dx1 constrain spring in the x-direction at the first node,

dy1 constrain spring in the y-direction at the first node,

dz1 constrain spring in the z-direction at the first node,

rx1 constrain spring about the x-axis at the first node,

ry1 constrain spring about the y-axis at the first node,

rz1 constrain spring about the z-axis at the first node,

v2 *i j k* vertex of the part as the second node,

pm2 *pointmass_#* point mass as the second node,

pminc2 *increment* increment the second point mass number when making part replications,

dx2 constrain spring in the x-direction at the second node,

dy2 constrain spring in the y-direction at the second node,

dz2 constrain spring in the z-direction at the second node,

rx2 constrain spring about the x-axis at the second node,

ry2 constrain spring about the y-axis at the second node,

rz2 constrain spring about the z-axis at the second node,

sddn *spd_#* material properties,

sminc *increment* increment the SPD number when making part replications,

amp *scale_factor* scale factor for the material properties,

orop *flag* orientation option,

where flag can be:

0 for spring/damper acts along the axis

1 for deflection/rotations are measured and force/moments applied along the following vector

2 for deflection/rotations are measured and force/moments applied along the projection of the spring/damper onto the plane with the following normal

prflg *flag* print flag,

where flag can be:

0 for forces are printed in DEFORC file

1 for forces are not printed in DEFORC file

ofsi *offset* initial offset,

xco *x-component*	x-component of the orientation vector,
yco *y-component*	y-component of the orientation vector, and/or
zco *z-component*	z-component of the orientation vector.

Remarks

This command creates or modifies a spring, with the options, in order to specify the direction of the spring and the material. A node defining the end of the spring can be a vertex of the present part or a point mass (see **npm** and **pm**). The **spring** command is usually invoked twice to generate a single spring, once for each node of the spring. This can be done across several parts or in the Merge Phase. This spring is replicated along with the part (see **lrep**, **grep** and **pslv**). Use the **spd** command to define the properties of the spring.

spring create/modify a numbered spring

spring *spring_# options ;*

where an *option* can be

n1 *node_#*	assign a structure node as the first node
pm1 *pointmass_#*	assign a point mass as the first node
dx1	constrain spring in the x-direction at the first nod
dy1	constrain spring in the y-direction at the first node
dz1	constrain spring in the z-direction at the first node
rx1	constrain spring about the x-axis at the first node
ry1	constrain spring about the y-axis at the first node
rz1	constrain spring about the z-axis at the first node
n2 *node_#*	assign a structure node as the second node
pm2	assign a point mass as the second node
dx2	constrain spring in the x-direction at the second node
dy2	constrain spring in the y-direction at the second node
dz2	constrain spring in the z-direction at the second node
rx2	constrain spring about the x-axis at the second node
ry2	constrain spring about the y-axis at the second node
rz2	constrain spring about the z-axis at the second node
sddn *spd_#*	specify the material properties
orop *flag*	assign an orientation option

where *flag* can be:

0	spring/damper acts along the axis
1	deflection/rotations are measured and force/moments applied along the following vector
2	deflection/rotations are measured and force/moments applied along the projection of the spring/damper onto the plane with the following normal

3 deflection/rotations are measured and force/moments applied along the vector between the following two nodes

prflg *flag* assign a print flag

where *flag* can be:

0 forces are printed in DEFORC file

1 forces are not printed in DEFORC file

ofsi *offset* assign an initial offset

xco *x-component* assign a x-component of the orientation vector

yco *y-component* assign a y-component of the orientation vector

zco *z-component* assign a z-component of the orientation vector

n3 *node_#* assign a third node for orientation

n4 *node_#* assign a fourth node for orientation

Remarks

This command specifies the direction of the spring and the material. The nodes defining the ends of the spring can be from a node in the mesh or a point mass (see **npm** and **pm**). The **spring** command is usually invoked two times to generate a single spring, once for each node of the spring. This can be done across several parts or in the Merge Phase. All of the options are not needed by all output options.

spwd spot weld property definitions

spwd *id options* ;

where *id* is a positive integer identifying the spot weld properties definition

where an *option* can be

sn nornal_force

ss shear_force

n normal_force_exponent

m shear_force_exponent

tf failure_time

ep effective_plastic_strain

nf #_force_vectors

tw time_window

Remarks

This command defines the properties of a spotweld. These definitions are needed by the **spw** command and the associated interactive command **spotweld**.

vd define a volume

vd *number box_type xm ym zm xx yx zx option* ;

where *number* is a unique volume number

where the *box_type* **box** has one of the *options*

adaptive *material level*	for BOX_ADAPTIVE
coarsen *inout_flag*	for BOX_COARSEN

where the *box_type* **sphbox** has the *option*

motion *xn yn yz load_curve_# flag*

where flag can be

0	velocity
1	displacement

IV. INDEX