

LS-DYNA Structured ALE (S-ALE) Solver (1)

Mesh Generation and Input Setup

Hao Chen

Livermore Software Technology Corporation

A new Structured ALE solver is recently added into LS-DYNA. It targets to solve ALE problems with structured mesh. It supports all parallel versions of LS-DYNA, i.e., SMP, MPP and MPP Hybrid. The new S-ALE solver has several advantages over the generic ALE solver. It runs faster, uses less memory and more stable.

The new solver generates structured ALE mesh automatically. The elements do not have to be equal-sized. Instead a user can define element spacing according to needs of the specific engineering problems. Listing all the elements and nodes used for ALE mesh in the input deck is now avoided. Rather we can simply define the mesh geometry by a number of control point pairs and let the automated mesh generator do the work. For larger ALE problems, the input deck size will be greatly reduced. The execution time on reading in keyword and writing out structured input decks are minimized. Also changing mesh geometry becomes much simpler.

The new S-ALE solver is easy to use, especially for users acquainted to the old generic ALE solver. Only two new keywords have been added so far. They are *ALE_STRUCTURED_MESH and its helper *ALE_STRUCTURED_MESH_CONTROL_POINTS. Most other ALE keywords remain the same.

Mesh Generation

Structured mesh: Nowadays most ALE problems use structured meshes. Most likely mesh is evenly spaced along any given direction. Occasionally, certain part of the mesh needs to be finer. For example, in order to accurately describing the shock pressure during the initial phase of a detonation process, users would most likely refine the mesh around the detonation point. In either case, mesh only contains regular 8-node hexahedrons with all faces perpendicular to each other. We call such mesh structured mesh.

The old ALE solver relies on users to define all these elements and nodes. Mesh is first generated in pre-processor and then written out as input deck files. Next users might do some touchups here and there using a plain text editor. Then these input files are read in by LS-DYNA. For structured meshes, this three steps process could be simplified into one. That is to tell LS-DYNA the geometric info of the S-ALE mesh and let it generate the mesh at the input deck read-in stage. This way, not only relieves users from the unnecessary mesh generation process, but also saves a lot of human and machine time and storage.

The new S-ALE solver generates structured mesh automatically by using the two following keywords: *ALE_STRUCTURED_MESH and *ALE_STRUCTURED_MESH_CONTROL_POINTS. The whole purpose of *_CONTROL_POINTS card is to provide mesh geometry for *ALE_STRUCTURED_MESH card at three local directions. Its format is as follows.

* ALE_STRUCTURED_MESH_CONTROL_POINTS					
CPID			SFO		OFFO
A1			O1		
A2			O2		
...			...		

CPID is the ID of the CONTROL_POINTS card; each (A1, O1) defines a (node, coordinate) pair. For evenly spaced mesh along one direction, the CONTROL_POINTS card is sufficient with just two pairs of (node, coordinate). For example, (1, 0.0) and (11, 1.0) would place the first node at coordinate 0.0 and the 11th node at coordinate 1.0. Later the *ALE_STRUCTURED_MESH card will use that information to generate 10 evenly space elements along that direction.

The above case is the simplest. Now we study the extreme case on the other end. Let us say along certain direction none of elements share same length. To define N elements along that direction, we need N+1 pairs of (node, coordinate) to specify every node's coordinate.

*ALE_STRUCTURED_MESH card needs element spacing information along all three local coordinate axes. If none of the geometry definitions is the same, we need three different CONTROL_POINTS cards. Otherwise we can simply reuse the existing CONTROL_POINTS card id.

The format of *ALE_STRUCTURED_MESH is as follows.

* ALE_STRUCTURED_MESH					
MSHID	PID	NBID	EBID		
CPIDX	CPIDY	CPIDZ	NIDO	LCSID	

MSHID is the ID of the generated structured mesh. It is not used so far but reserved for the future implementations of multiple structured meshes. PID is the part ID for generated elements. NBID and EBID is the starting ID of the structured mesh nodes and elements, respectively. Please note to avoid conflict, we need NBID and EBID to be larger than the existing largest node ID and element ID, respectively.

CPIDX, CPIDY and CPIDZ are the CONTROL_POINTS card IDs along the local x, y and z direction. These CPIDX, CPIDY, CPIDZ values can be the same if the mesh geometry is the same along those directions.

NIDO and LCSID are used to define the mesh origin and local coordinate system. NIDO is the node ID of the mesh origin. We can describe the translational motion for the structured mesh by applying

prescribed motion to NID0 through *BOUNDARY_PRESCRIBED_MOTION. LCSID is the ID of *DEFINE_COORDINATE card. We recommend the usage of *DEFINE_COORDINATE_NODES with IFLAG=1 option for users want to have a rotating mesh during simulations.

*DEFINE_COORDINATE_NODES card defines a local coordinate system through 3 user-defined nodes. With prescribed motion applied on those three nodes, the generated structured mesh then rotates with the local coordinate system during the run.

Here PID deserves some further explanation. The PART in *ALE_STRUCTURED_MESH card is nothing but a collection of nodes and elements in the generated structured mesh. It does not, and should not contain any material properties. Rather, it simply represents a spatial domain where all the ALE materials (ALE Multi-Material Group, shortened as AMMG) flow in and out. In the context of S-ALE, we name this part MESH PART. This part contains only geometry information in contrast to the regular PART definition in Lagrange analysis which is a collection of geometry (*NODE+*ELEMENT) AND material information (*MAT_+*EOS_+*HOURGLASS_).

Hence, users only need to give an unused PART ID for the PID field in *ALE_STRUCTURED_MESH card. It is not needed to define a *PART card for this PID; S-ALE will automatically define the PART and tag it as a S-ALE mesh part.

Another kind of PART used in S-ALE is MATERIAL PART. Contrary to MESH PART, MATERIAL PART doesn't contain any geometry information. It serves as a wrapper to contain a collection of (*MAT_+*EOS_+*HOURGLASS_). To use PART definition as the material wrapper is rather a reluctant choice. It is simply because there is no other definition existed in LS-DYNA includes all three entities of a complete material definition, i.e., *MAT, *EOS and *HOURGLASS. The MATERIAL PART ID is only used when defining *ALE_MULTI-MATERIAL_GROUP card. It should not appear elsewhere in the input deck.

In S-ALE solver, we enforce a one to one correspondence between MATERIAL PARTs and ALE Multi-Material Groups (AMMGs). We believe this would bring much less confusion in setting up S-ALE models. Please note this one to one correspondence is unique to S-ALE solver; the general ALE solver doesn't have this requirement.

S-ALE Input Setup Mesh

In this section we use a simple example to show how to set up a S-ALE input deck. The input files used here can be downloaded at the following address: <http://ftp.lstc.com/anonymous/outgoing/hao/sale/>. Other S-ALE examples and documentations can also be found there.

Take a simplified underwater explosion problem. Let us say we have all other parts of the problem setup already such as the Lagrange structure mesh, material and all the control cards. Now we start the S-ALE setup.

Step 1: S-ALE Mesh Generation

1. Set up the following three * ALE_STRUCTURED_MESH_CONTROL_POINTS cards to provide spacing info along the three local axes.

* ALE_STRUCTURED_MESH_CONTROL_POINTS					
3001					
1			-12.0		
25			12.0		

There are 25 nodes along x-direction from -12.0 to 12.0; 24 elements evenly spaced.

* ALE_STRUCTURED_MESH_CONTROL_POINTS					
3002					
1			0.0		
13			12.0		

There are 13 nodes along y-direction from 0.0 to 12.0; 12 elements evenly spaced.

* ALE_STRUCTURED_MESH_CONTROL_POINTS					
3003					
1			0.0		
17			16.0		

There are 17 nodes along z-direction from 0.0 to 16.0; 16 elements evenly spaced.

2. Generate Mesh.

* ALE_STRUCTURED_MESH					
MSHID	PID	NBID	EBID		
1	9	200001	200001		
CPIDX	CPIDY	CPIDZ	NIDO	LCSID	
3001	3002	3003	199997	890	

The PID is 9; starting NODE ID and ELEMENT ID are both 200001; CONTROL_POINTS ID along three axes are 3001, 3002 and 3003; the mesh origin is at node 199997; location coordinate system ID is 890 and defined as follows.

* DEFINE_COORDINATE_NODES					
LCSID	NID1	NID2	NID3	IFLAG	
890	199998	199999	200000		

Here, IFLAG=0 means the local coordinate system is set up at the initial time step and won't be updated during the run.

Step 2: Define S-ALE multi-materials

1. Define Material Parts: We have the following three materials in S-ALE mesh, high explosive, water below structure and water above structure.

* PART					
Water below plate					
PID	SECID	MID	EOSID	HGID	
10	10	10	10	10	

*SECTION_SOLID					
SECID	FORM				
10	11				

*MAT_NULL					
MID	RO	PC	MU		
10	1.00				

*EOS_GRUNISEN					
EOSID	C	S1	S2	S3	GAMA0
10	0.148	1.75	0.0	0.0	0.28

*HOURGLASS					
HGID	FORM	FACTOR			
10	1	1.0E-6			

Only water material definition is given above; the other two are skipped.

* PART					
High explosive					
PID	SECID	MID	EOSID	HGID	
11	10	11	11	10	

* PART					
Water above plate					
PID	SECID	MID	EOSID	HGID	
12	10	12	12	10	

2. Include all S-ALE Material Parts in *ALE_MULTI-MATERIAL_GROUP card

* ALE_MULTI-MATERIAL_GROUP					
PID	TYPE				
10	1				
11	1				
12	1				

In the above card, PART 10 represents material “water below”, marked as AMMG #1; PART 11 “high explosive”, AMMG #2 and PART 12 “water above”, AMMG #3. Note here all those PARTs are only to provide material definitions; no mesh info at all.

Step 3: Define the initial configuration of all ALE multi-materials

In theory, any given element inside the S-ALE mesh could be fully or partially occupied by any ALE multi-material(s) defined. Naturally comes the question that how we record such information. We introduce an additional element history variable called “volume fraction”. We calculate volume fractions for each AMMG in each ALE element and record them accordingly. Later we use these volume fractions to reconstruct the material interfaces inside each element.

So, it is required for users to provide the initial values of those volume fractions; same as users need to provide values for initial stresses. We use the following command to generate those values.

*INITIAL_VOLUME_FRACTION_GEOMETRY					
SID	IDTYP	BAMMG	NTRACE		
9	1	1			
TYPE	FILLOPT	FAMMG			
6		2			
X0	Y0	Z0	R		
0.0	0.0	0.0	2.0		
TYPE	FILLOPT	FAMMG			
3		3			
X0	Y0	Z0	XCOS	YCOS	ZCOS
0.0	0.0	12.0	0.0	0.0	15.0

Here we first fill the MESH PART 9 completely with AMMG 1 (water below); then we switch the water to AMMG 2 (High explosive) inside a sphere with radius 2.0 and centered at (0.0, 0.0, 0.0); at last we switch AMMG 1 (water under) to AMMG 3 (water above) above a plate located at (0.0, 0.0, 12.0).

Step 4: Miscellaneous

1. *CONTROL_ALE and *INITIAL_DETONATION

* CONTROL_ALE					
DCT	NADV	METH	AFAC		
		1	-1		
START	END	AAFAC	PRIT		

The only option users need to care is METHOD while using the S-ALE solver. Currently, S-ALE supports two advection methods: Donor Cell (1st order) and Van Leer (2nd order). AFAC=-1 is mandatory for using multi-material ALE formulations. It means mesh smoothing is turned off.

* INITIAL_DETONATION					
PID	X	Y	Z	LT	
9	0.0	0.0	0.0		

*INITIAL_DETONATION is to set up the detonation center. Please note the PID here needs to refer to the MESH PART. As we know that MATERIAL PARTs do not contain any elements.

2. Boundary Conditions

As for the boundary condition, it is pretty much the same as in a typical Lagrange or ALE problem.

*BOUNDARY_SPC to apply fixed or symmetric boundaries; *BOUNDARY_PRESCRIBED_MOTION to apply velocity boundaries; *LOAD_SEGMENT to apply pressure boundaries.

There is one unique problem only S-ALE solver faces. It is how to define Node Sets and Segment Sets with node ids not ready. For now, the solution we provide is to use *DEFINE_BOX.

* DEFINE_BOX						
BOXID	XMIN	XMAX	YMIN	YMAX	ZMIN	ZMAX
1	-12.0	12.0	0.0	0.01	0.0	16.0

First we define a box at the -Y face of the S-ALE mesh.

*SET_NODE_GENERAL					
SID					
1					
OPTION	E1				
BOX	1				

Then we include all the nodes inside the box to make NODE SET 1.

* BOUNDARY_SPC_SET					
ID	LCSID	X	Y	Z	
1		0	1	0	

Finally we lock Y direction at all nodes in NODE SET 1. Now we get symmetric condition applied.

3. Fluid Structure Interaction (FSI)

S-ALE uses exactly the same FSI algorithm and keyword ALE solver uses, with only one tiny modification to the MCOUP field. Now MCOUP can be a PART SET which includes all the AMMGs coupling to the structure. Here PARTSET 1011 contains MATERIAL PART 10 (water below, AMMG1) and 11(high explosive, AMMG 2). This way, we can avoid the unnecessary *SET_MULTI-MATERIAL_GROUP_LIST card.

* CONSTRAINED_LAGRANGE_IN_SOLID							
SLAVE	MASTER	SSTYP	MSTYP	NQUAD	CTYPE	DIREC	MCOUP
1	9	0	1	2	4	1	1011