Sequential drop test simulations through automated process in Workbench LS-DYNA

Yury Novozhilov[1], Ulrich Stelzmann[1],

[1]CADFEM Germany GmbH

1 Introduction

Modeling drop tests of various goods has always been one of the classic problems for explicit structural dynamics. Current requirements for such tests often require the consideration of multiple sequential impacts, such as the procedures described by the International Safe Transit Association. CADFEM is investigating the possibilities of automating such sequential drop tests in the Workbench LS-DYNA environment. With the automated calculation of sequential drop tests, these can be integrated into optimization cycles.

2 Multi-stage analysis in LS-DYNA

In many applications that are typical for LS-DYNA, several processes must be simulated in a sequence so that the deformations are carried over and continued. There are generally two techniques for this: several restarts or the use of the dynain file.
At first glance, the dynain workflow seems much more flexible, because you build a new model each time and have almost complete freedom. However, this workflow was originally developed for forming simulation (**`*INTERFACE_SPRINGBACK`**) and there you have the special situation that only one part is deformed. In a drop test of a consumer good, such as a washing machine, the object under consideration consists of a large number of bodies. Using dynain, the stresses, strains and possibly also the history variables of the elements can be easily initialized, but more is required. Such devices usually contain a large number of contacts and other connecting elements such as welding points, springs and joints. These would also have to be initialized in order to pass on a preload state cleanly. Although the contact state can now also be initialized in the dynain, this is currently still very limited (mortar contact, very complex), other things are completely missing, such as joints or discrete beams.
A restart, we are mainly talking about a small restart here, does not have these problems for the time being. Here, all contacts and elements are initialized cleanly (except for some bugs). However, you have to accept some disadvantages:

- You have to continue working with the same executable and the same parallelization
- Only limited changes to the model are possible, in particular nothing new may be added.

Nevertheless, I have learned to appreciate the small Restart in many years of practice, because it offers some very helpful features:

- Initial velocities can be reset again and again.
- Parts, single elements or contacts can be deleted.
- And my favorite: the content of **`*DEFINE_CURVE`** can be changed later as long as the total number of data pairs does not change.

The fact that a small restart simply continues to write all result files also appears to be very useful, so that consistent post-processing is easily possible at the end.
We will therefore concentrate on the small restart.

3 Multiple load cases for drop tests

Drop tests are particularly important when it comes to packaging for transportation, i.e. we are concerned here with drop tests of packaged goods, typically packed in cardboard boxes. The position in which such a box hits the floor is likely to play a major role. On the one hand, we have 6 surfaces, which are likely to generate the hardest impact, 12 edges, which are probably somewhat softer, and 8 corners, which are likely to deform the most locally on impact. But which impact point is most critical for the packaged goods? This is difficult to estimate in advance, so all 26 possible positions would probably have to be tested individually.

© 2024 Copyright by Ansys Inc.

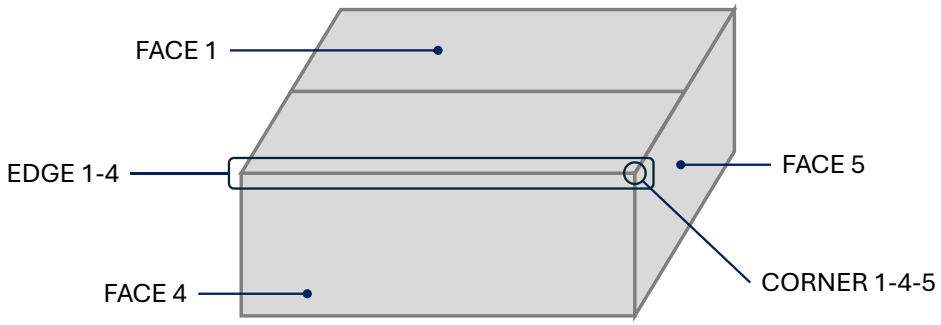


Fig.1: Typical markup of faces, edges, and corners for a consistent drop test

Many companies limit themselves to a smaller number. Nevertheless, these are many very similar analyses with exactly the same model but different case directions. The ***CASE** command of LS-DYNA can be used very nicely for this. It is probably easiest to define the different directions of fall using rigid plates or RIGIDWALLs as the ground arranged in space and the initial velocity from the height of fall. Optionally, the gravity could also be defined in the corresponding direction.
We use Ansys Workbench LS-DYNA for model creation and the CASE command is supported here in a very simple and user-friendly way. The model itself is not rotated, but a falling direction is defined at an angle in space, along with a wall that is at an angle in space.

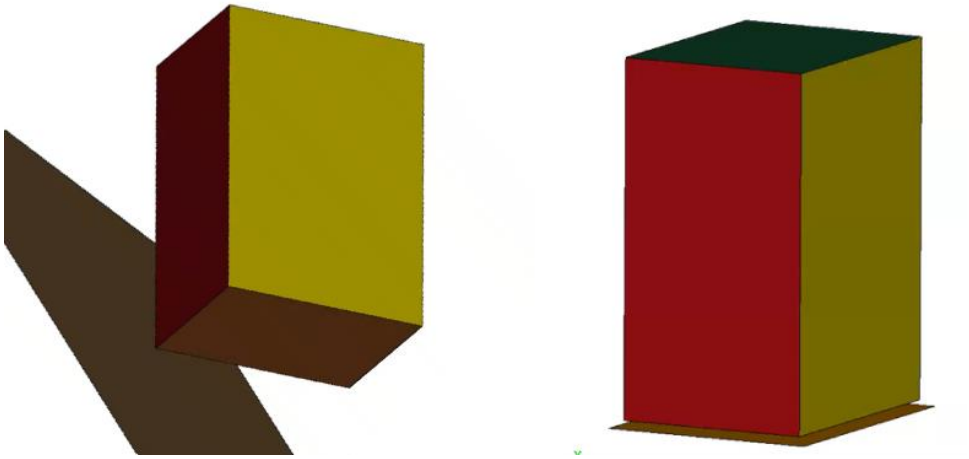
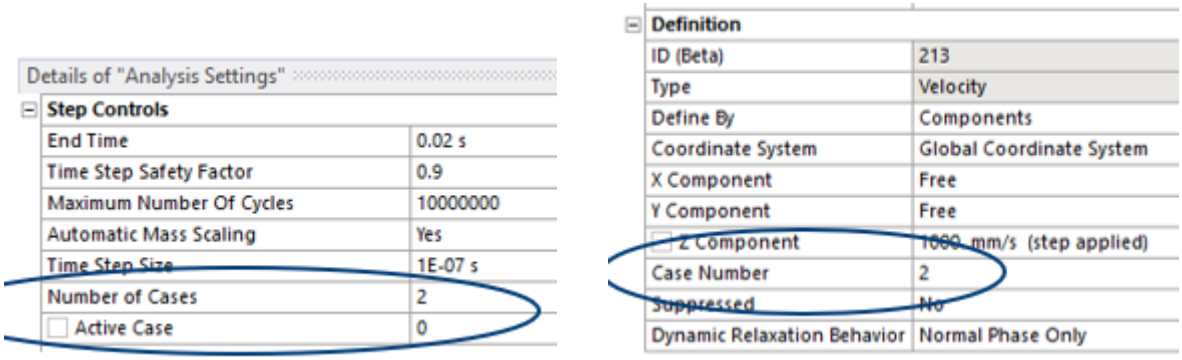


Fig.2: Two load cases with different falling directions using one simulation model

This is a nice application for the case control in Workbench:



Details of "Analysis Settings"

Step Controls	
End Time	0.02 s
Time Step Safety Factor	0.9
Maximum Number Of Cycles	10000000
Automatic Mass Scaling	Yes
Time Step Size	1E-07 s
Number of Cases	2
Active Case	0

Definition	
ID (Beta)	213
Type	Velocity
Define By	Components
Coordinate System	Global Coordinate System
X Component	Free
Y Component	Free
Z Component	1000. mm/s (step applied)
Case Number	2
Suppressed	No
Dynamic Relaxation Behavior	Normal Phase Only

Fig.3: Case control in Workbench LS-DYNA

These simulations can therefore be used to find out quite easily which drop position is most critical for the packaged component, whether due to local plastic deformation, the failure of connecting elements or simply high acceleration.

© 2024 Copyright by Ansys Inc.

4 Solving sequential load cases for drop tests

However, many regulations also require a packaged good to withstand several drop tests in sequence, e.g. first onto a surface, then onto an edge and then onto a corner, etc. As we have seen above, we prefer several small restarts for such sequential load cases. For an optimization or a sensitivity study, it would be desirable to be able to simply calculate all sequential drop tests one after the other without manual intervention. However, this presents a difficulty: Particularly in the case of edge or corner impacts, the entire body will somehow twist in space on impact. If you then need a defined fall position for the next impact, you would either have to turn the body back again first or define a new RIGIDWALL and corresponding initial velocities for the new fall direction. However, this would require manual intervention and thus interfere with the automatic running of an optimization loop. To avoid this, we have developed the following idea:
At the beginning, the model of the packaged body is created and one "rigid" plate is already defined for each desired impact position as the ground.

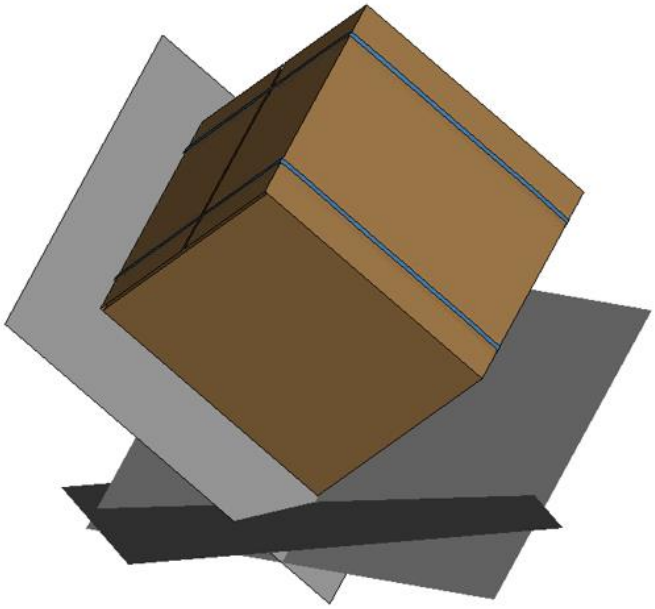


Fig.4: packaged model with three impact planes for surface, edge and corner impact

In the first calculation run, however, the falling body is only in contact with the first impact plane. The other impact planes are connected to the falling body in such a way that they move with it and, in particular, rotate with it. This means that these future impact planes always remain in the same position relative to the falling body. In a first restart, the first impact plane is then deactivated, the second impact plane is fixed in space and the falling body is now given its initial velocity perpendicular to this new impact plane. This procedure can be continued for any number of impact positions with a corresponding number of impact planes.
The challenge here is how to connect several rigid planes to the falling body in such a way that they move with it, but do not influence its movement themselves. This is easily possible in LS-DYNA. All we need is a (small) rigid body on or in the falling body. This could, for example, be the base of an accelerometer (***element_seatbelt_accelerometer**). The individual impact planes are initially not rigid bodies but separate shell elements with ***MAT_NULL**. Exactly at the positions of the four corner nodes of these null shells there are again coincident nodes which are not connected to any elements, and which belong to the rigid accelerometer as extra nodes. The movement of these extra nodes can always be read out using ***DEFINE_CURVE_FUNCTION**. These curves can be used immediately to move the corner nodes of the null shells using ***BOUNDARY_PRESCRIBED_MOTION**. In this way, the null shells follow the movement of the drop body exactly without influencing it. It is important to know that the combination of ***DEFINE_CURVE_FUNCTION** and ***BOUNDARY_PRESCRIBED_MOTION** will only work for velocities, not for displacements. Instead of the ***DEFINE_CURVE_FUNCTION** also ***SENSOR** keywords can be used to grab the motion of the extra nodes. This would give the additional possibility to filter the signal before using it in the ***BOUNDARY_PRESCRIBED_MOTION**. This could improve the stability but will also increase the complexity.

© 2024 Copyright by Ansys Inc.

If one of these null shells is then required as the impact plate for the next drop, deactivate it's ***BOUNDARY_PRESCRIBED_MOTION** and convert it to a rigid body using ***DEFORMABLE_TO_RIGID_AUTOMATIC** and connect it to a master rigid body, which was already modeled in the beginning and is fixed in space.
This type of modeling may seem somewhat complex at first glance, but the Workbench LS-DYNA user interface can also be helpful here. Of course, it would be most convenient if this type of modeling were already pre-programmed, e.g. in the form of a drop test module.
Unfortunately, there is one showstopper in this procedure: As the orientation in space for each impact surface is only determined by the simulation, the initial velocity must be defined perpendicular to the corresponding impact surface at the start of each new drop test. This could be realized very easily with a moving coordinate system and the keyword ***CHANGE_VELOCITY_GENERATION**. According to the manual, the ***CHANGE_VELOCITY_GENERATION** has the same features as the ***INITIAL_VELOCITY_GENERATION**, i.e. you should actually be able to use a local coordinate system here. Unfortunately, however, this is apparently not implemented. With the ***CHANGE_VELOCITY_GENERATION**, all inputs are only ever understood in the global coordinate system. This would require manual conversion and input and would disrupt the automatic workflow. Fortunately, however, the conversion from the local direction of fall to the global coordinates can be implemented within the LS-DYNA workbench using Python programming.

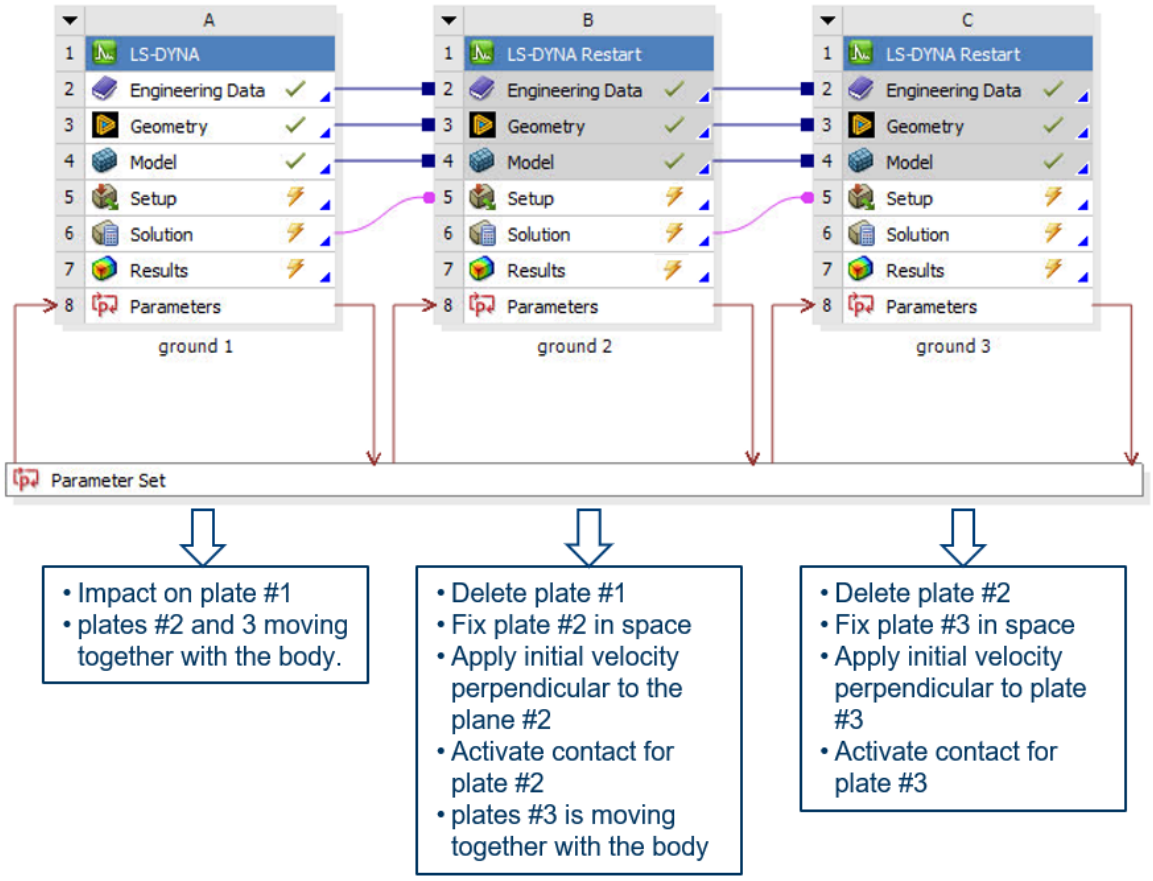


Fig.5: Workflow on the Workbench project page for automated run of three sequential drop tests

5 Summary

An idea is presented of how several sequential drop tests can be simulated in an automated process in such a way that no manual intervention is required. The LS-DYNA workbench is used for modeling.
Building more accurate models of the actual testing process naturally opens the way to optimizing packaging and improving the sustainable consumption of packaging materials.
Based on the results of that work, suggestions are made for improving the Workbench environment to automate and simplify the simulation of sequential drop tests.

© 2024 Copyright by Ansys Inc.