

# Getting Your Model ‘Right’ – Checking Before, During and After Your LS-DYNA® Analysis

Gavin Newlands

*Arup*

## Abstract

*To be able to capture more and more detail in an analysis, LS-DYNA models continue to increase in size and complexity. A model can now commonly have ten to twenty million elements and it is essential that checking procedures are used to ensure that the model is ‘right’. As element count increases, checking a LS-DYNA model can be a difficult process as many of the checks can be very complex. Having robust methods for checking a keyword deck both interactively and as part of any automatic process is therefore essential for quality models and correct results. Ultimately this will also give significant time savings.*

*In this paper, using the Oasys LS-DYNA Environment software we will look at various checking methods that can be used before, during, and after a LS-DYNA analysis. These will include:*

### Pre-Analysis Checking

- *Simple and detailed checks.*
- *User defined checks.*
- *Building checking into to an automatic process.*

### Checking During the Analysis

- *Viewing and investigating the information in the d3hsp file after analysis initialisation and using it to check and fix potential issues.*

### Post-Analysis Checking

*Sometimes, despite all our efforts an analysis does not behave as expected. We will give examples of typical analysis problems; how they can be viewed and investigated in post-processing and instantly fixed in the input deck.*

## Introduction

To be able to capture more and more detail in an analysis, LS-DYNA models continue to increase in size and complexity. A model can now commonly have ten to twenty million elements and it is essential that checking procedures are used to ensure that the model is ‘right’. As element count increases, checking a LS-DYNA model can be a difficult process as many of the checks can be very complex. Having robust methods for checking a keyword deck both interactively and as part of any automatic process is therefore essential for quality models and correct results. Ultimately this will also give significant time savings.

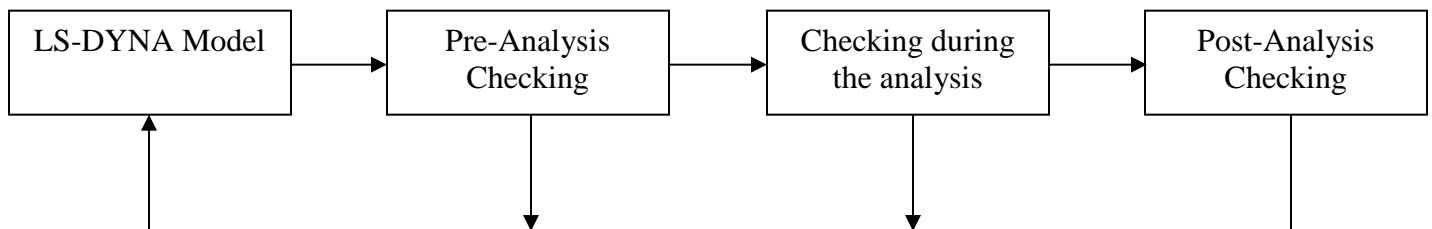
“Checking” can be construed in many different ways. In the context of this paper it considers everything you may do, as an individual or as part of an automatic process from the start of model creation to the end analysis results. Of particular note is the importance of having a process, whether automatic or otherwise.

For this paper, the checking process has broadly been split into three sections:

**Pre-analysis checking** – any checks you may wish to do before running your analysis in LS-DYNA.

**Checking during the analysis** – Any checks you may want to do whilst the model is running.

**Post-analysis checking** – checks on your models after the analysis is complete. This includes feedback to access initial model data.



The above diagram shows how checking fits into the whole process of creation and analysis of a LS-DYNA model. For the three sections of checking mentioned above, there is always feedback into the LS-DYNA input keyword model. Pre-analysis checking will lead to fixing keyword data before running. Checking during the analysis will give information on errors in your keyword deck, as well as giving information on mass and timestep, which may lead to changes to the input model. Post-analysis checking will provide further information that will incur changes to the input model (for example contact modification after reviewing instabilities).

In this paper we will show the benefits of using the Oasys LS-DYNA Environment software (PRIMER, D3PLOT, T/HIS and REPORTER) for checking in the areas outlined above.

## Pre-Analysis Checking

This is arguably the most important step of the checking process. Having confidence in the model you are submitting into LS-DYNA is important, and having a common process in place for checking before submission can lead to consistently “healthy” models across a larger team.

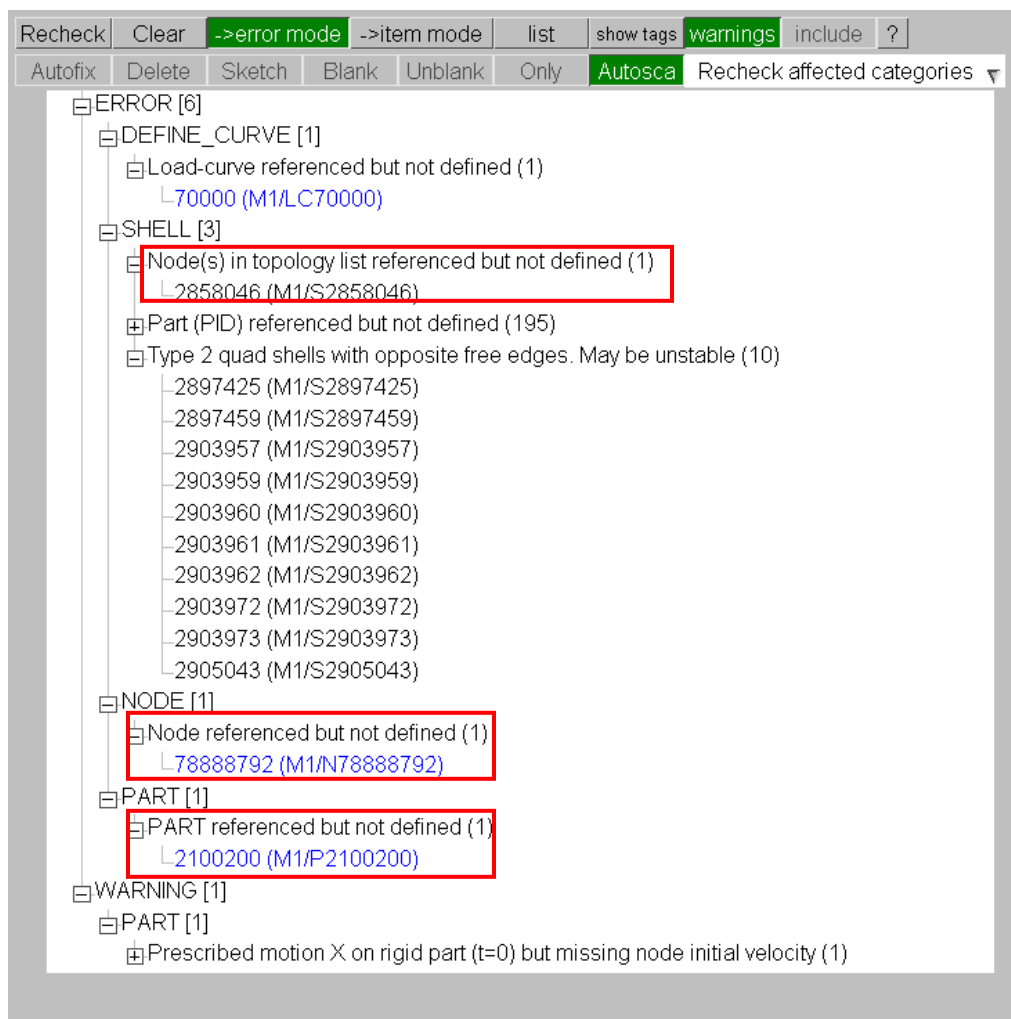
The pre-analysis checks can often rely on the pre-processor used, and the degree of checking varies from pre-processor to pre-processor. However, relying on pre-processor checks is often not enough, and experience is used to determine if a model is “healthy” or not. As engineer experience can vary, this is where the importance of having a checking process can be useful. First of all, we will look at the typical types of check carried out before submission into LS-DYNA.

### Simple Checks

Simple checks can be classified in many ways. Quite often they are using checks built into your pre-processor of choice. These will check things such as:

- Reference problems. Entity “A” refers to entity “B” but entity “B” does not exist.
- Element quality checks.
- Incorrect values in certain fields (for example a negative value when this is not valid).

The above are simple checks but extremely important – usually errors like the above will result in LS-DYNA not running. The following image shows output from a model check in Oasys PRIMER, which contains thousands of LS-DYNA specific checks.

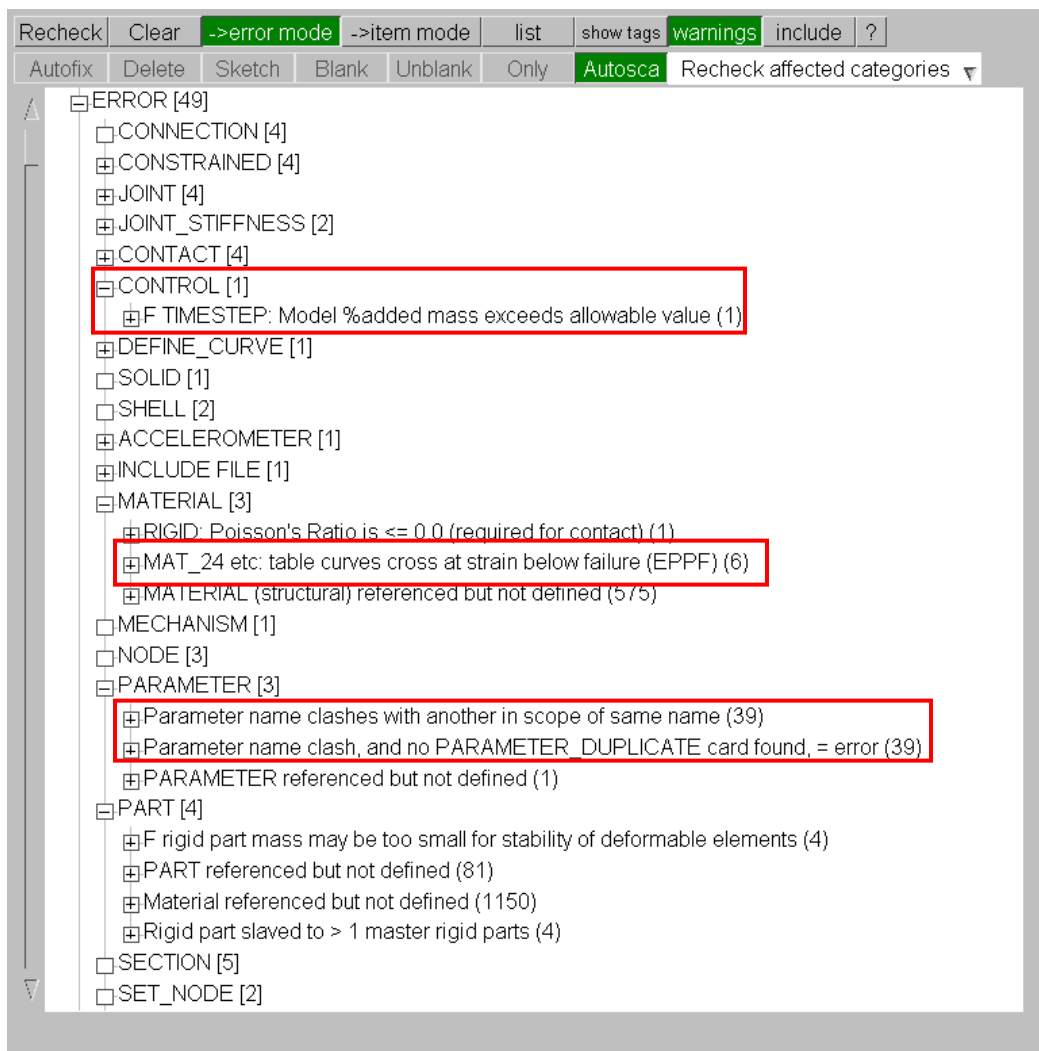


## More Detailed Checks

These are checks that go beyond the simple checks mentioned above. These can include:

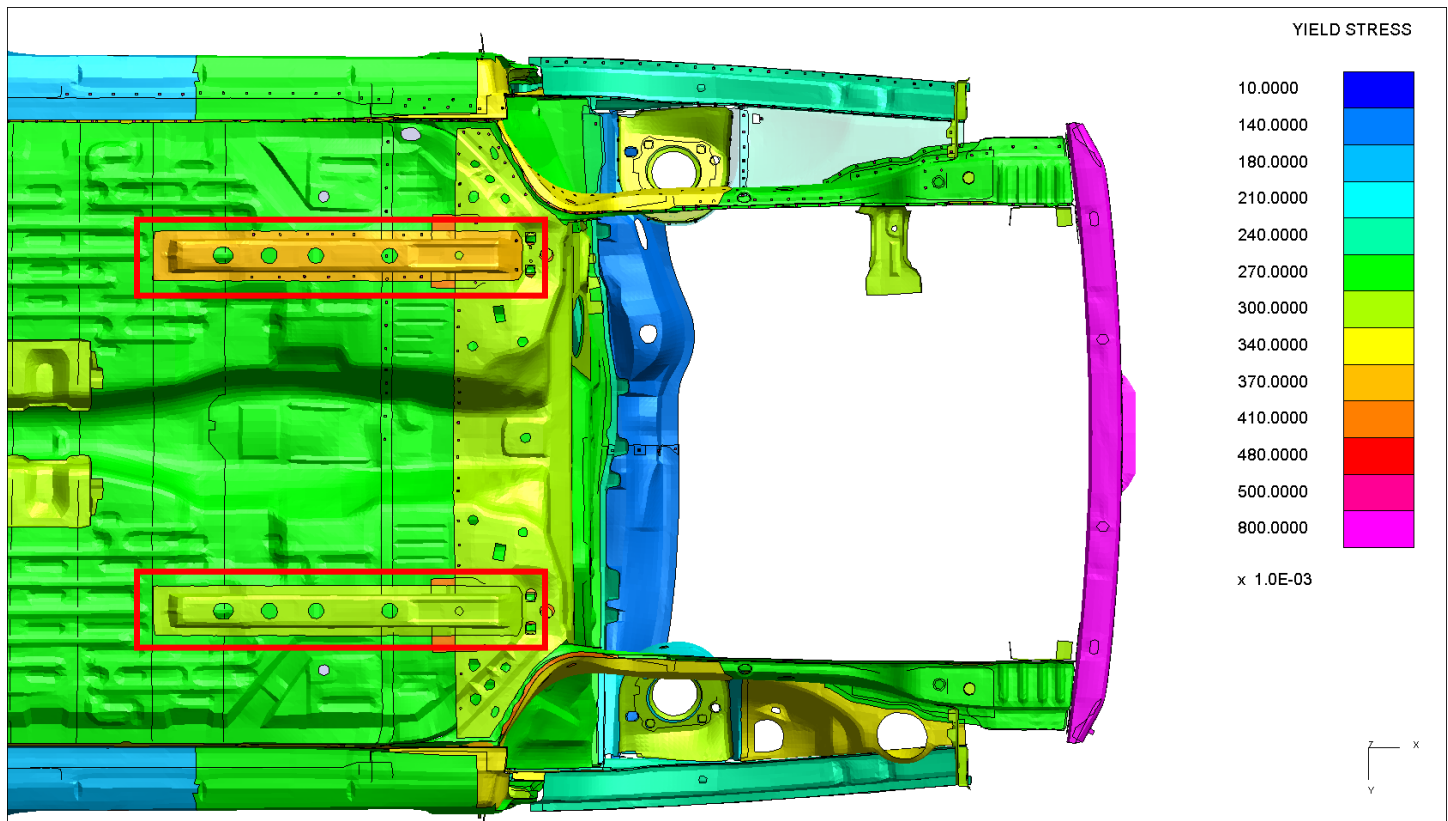
- Checks on curves referenced by particular material types.
- Rigid body constraint clash checks.
- Tied (or not tied) contact checks/penetration checks.

The above types of check are also important. If the issue does not lead to a termination when running in LS-DYNA it is important that the issue is picked up at the pre-analysis checking stage, as it may go unnoticed. LS-DYNA will run successfully and not produce any errors, but the result could be wrong or misleading. The following image again shows a check output from Oasys PRIMER, but this time highlighting some of the more specific/detailed checks, which are included in Oasys PRIMER based on many years' of experience using LS-DYNA.



## Visual Checking

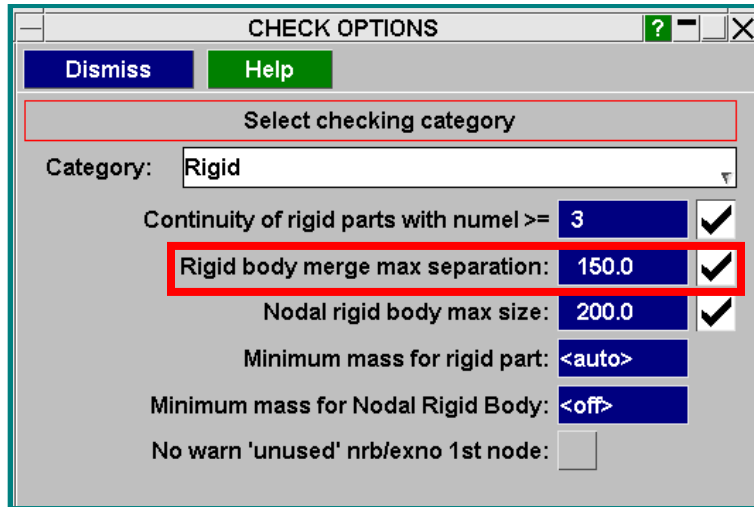
Visually checking your model is also important. Using pre-processor tools to check attachment between entities, or using contour plots to visually check data in your model can build confidence. The following image shows an example of this. Here, we are plotting material yield stress in Oasys PRIMER. The two highlighted reinforcements are contoured different colours, when you would expect the yield stress for these symmetric parts to be the same.



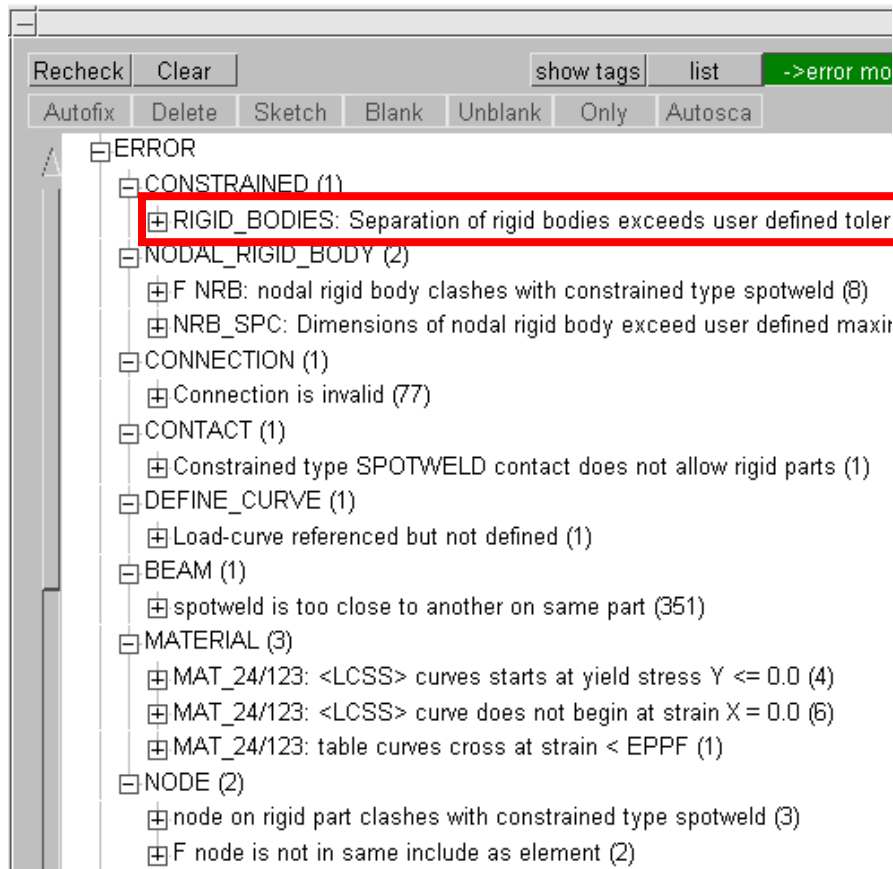
## User defined checks

The issues with the above pre-processor checks are that you are reliant on the checking tools within the pre-processor. This paper considers checking using Oasys PRIMER, which includes many checking tools specifically for LS-DYNA models, however even then there may be some checks missing specific to a particular user's/companies' requirements.

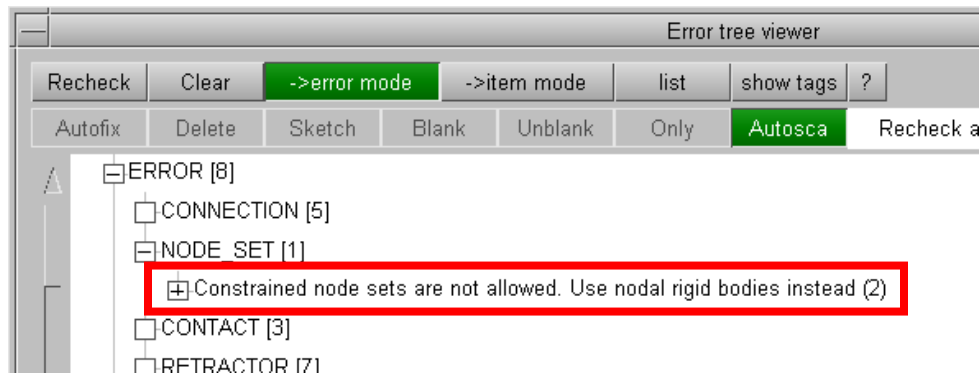
To that end, it is often possible to create user defined checks. Using Oasys PRIMER, there are different ways of doing this. One way is that many of the checks can be turned on/off and have user inputs to modify how the check works. One example for this is a separation check for \*CONSTRAINED\_RIGID\_BODY definitions. You may want to know of any \*CONSTRAINED\_RIGID\_BODY definition where the distance between the parts referenced is above a specified value. This allows you to detect modelling mistakes which will be valid when running in LS-DYNA but will give incorrect results. In Oasys PRIMER these can be set on the check options panel:



Which will then show up when running a normal model check:



Another way of introducing a user defined check is through scripting. This may be within a pre-processor or using an external script, but scripting allows you to interrogate data in any way you want, and to flag up any issues found in the model. These types of checks are useful when looking for issues such as company specific checks, where you want members of your team to follow particular modelling guidelines. One example is the use of \*CONSTRAINED\_NODE\_SET. It is perfectly legal to use this in a LS-DYNA analysis, but you may want to flag up its usage, as \*CONSTRIANED\_NODAL\_RIGID\_BODY may be the card that is actually required (as \*CONSTRAINED\_NODE\_SET does not have rotational coupling). Since this will not generally be picked up by a pre-processor (as it is valid in LS-DYNA) then a user script could be written to catch this. In Oasys PRIMER, user scripts can be written to check entities in any way. These checks can be incorporated into the main checking tool and easily rolled out across all members of a team. For the above example, the output would look something like this:



Using this method you can add multiple checks for each entity type (for example node, shell, part etc.). The check script will be run for every entity, just like the default Oasys PRIMER checks. You also have the option of adding a “general” check, not linked to a specific entity type that can do anything you want – an example being checking the overall mass of the model against certain targets.

One other aspect of running checks in a pre-processor is configuration. We have seen above how it is useful to add user-defined checks, but it is also useful to turn on/off checks that natively exist in the pre-processor. This can be done in Oasys PRIMER so that any check and warning can be promoted/demoted as you see fit. This allows you to tailor the checking report to your specifications.

### ***Checking as part of an automatic process***

As mentioned, in a larger team/organization it can be difficult to ensure the same checks are applied across engineers and models – the model checking process for one person could be very different to the model checking process for someone else.

The above leads to the introduction of a checking process. This can be a guidance document that engineers can follow. It can also be an automatic process built into your pre-processor or pre-submission scripts. This can be more difficult to implement, but can give significant rewards in terms of consistency of model quality.

Some examples of items you may want to add to an automatic process to compliment pre-processor checking are:

- Highlighting important errors and warnings.
- Introducing new checks not covered by the pre-processor.
- Company specific checking – make sure people are following company specific rules.
- Extracting checking information from different sources – output files as well as input.

The following is an example of the Oasys PRIMER checking “dashboard” which pulls together checking information from a variety of sources. Some of the sources are again scripting based.





The example shows various checks and whether the model passes the check or not. The scripting based checks are as follows:

Model Metrics – checks model mass and timestep/added mass information against company guidelines.

Control cards – Check if the control cards being used, or the values on the control cards follow your company guidelines.

ELFORM check – checks to see if there are any type 16 parts with < 5 integration points (company specific check).

Instrumentation – You may have an include file that always has to be included in your model that contains instrumentation information. This check looks to see if that include is present or not.

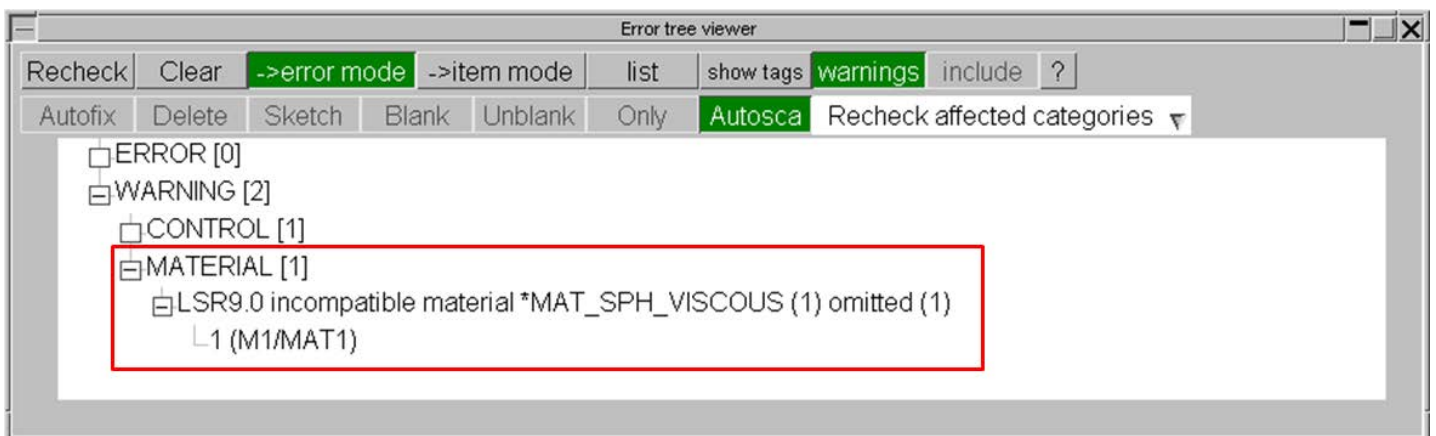
MAT 100 check – check to see if we have any nodes on MAT 100 parts not in a tied contact.

MAT rigid constraint checks – checks to see if we have any MAT 100 cards with rigid constraint values set.

As you can see with the above, they are not things that would cause LS-DYNA to terminate in error, so may not be things that are picked up generally by a pre-processor's built in checks, but they are useful tools that can be used to ensure consistency in your output and lead to "healthier" models. The dashboard checks can be used to supplement the user defined script based checks mentioned earlier. The dashboard checks are intended to give an overview of the model health.

### ***Keyword checking against LS-DYNA version***

Another area of checking to consider is validity of the keywords and fields you use. Often, the pre-processor you use will support LS-DYNA keywords for the latest released version of LS-DYNA, whereas you will be using an older version of LS-DYNA for consistency of results throughout a project. It is useful to be able to check the data you are using to ensure the keywords and fields are valid in the version of LS-DYNA you will be running in. One example here is receiving a model from a supplier which may use keywords/fields that are not supported in the version of LS-DYNA you are running in. Oasys PRIMER will do this, both in the standard model check, but also when writing out your model.



## Checking During the Analysis

When your model has been submitted to LS-DYNA, there are various checks you can carry out whilst the model is running to ensure everything is OK. Some of these things are covered in the following section (post-processing checking), so in this section we will concentrate on the output to the various text information files that are produced by LS-DYNA. These include the d3hsp file, the .log file and the various messXXX files. These files contain lots of useful information, and it is often useful to build investigation of this information into your checking process. Information included in these files include:

- Timestep information.
- Contact penetration information.
- Added mass information.
- Convergence information (implicit).

You will also get error and warning messages from LS-DYNA's internal checking. These are extremely useful for identifying and tracking down problems in your model, because as well as the message you also get references to entity labels that the problem relates to. Here is one common example:

```

*** Error 40456 (SOL+456) (processor # 0)
    NaN detected.

*** Warning 11080 (KEY+1080)
    *RIGIDWALL_GEOMETRIC_FLAT ID 1999 has length less than or equal t
    LENL = 0.0000E+00; LENM = 0.0000E+00;

*** Warning 70025 (OTH+25) (processor # 0)
    Memory is set 378702 words short
    increase the memory size to 52000000

*** Warning 70025 (OTH+25) (processor # 0)
    Memory is set 7192330 words short
    increase the memory size to 59192330

*** Warning 70025 (OTH+25) (processor # 0)
    Memory is set 821210 words short
    increase the memory size to 63192330

*** Warning 10177 (KEY+177)
    nodal rigid body and part share the same ID 18101

*** Warning 10177 (KEY+177)
    nodal rigid body and part share the same ID 100000

*** termination due to out-of-range moments
    number of nodes has out-of-range moments          402
Node list:
10500941 10500942 10500943 10500944 10500945 10500946 10500947
10500949 10500959 10500960 10500963 10500964 10500965 10500966
10500968 10500969 10500970 10500971 10500981 10500983 10500985
10500988 10500990 10500991 10500993 10500996 10500999 10501001
10501002 10501004 10501005 10501006 10501007 10501008 10501009

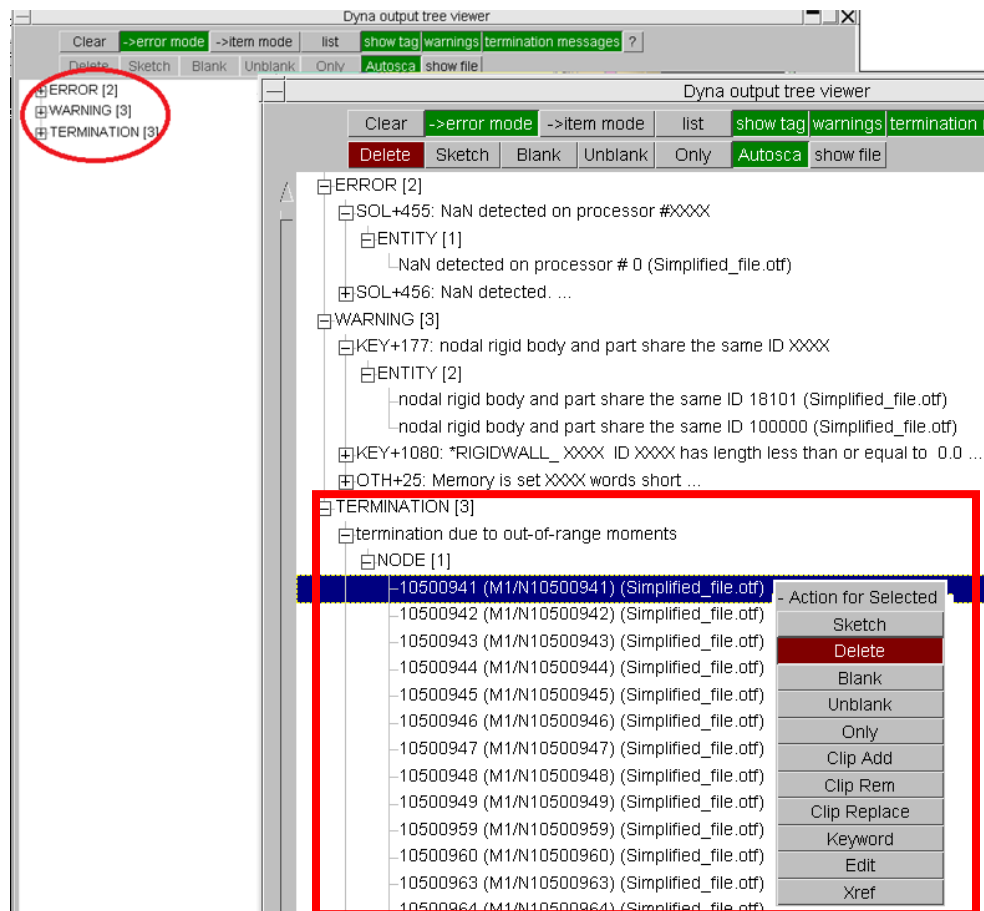
```

In the above, the analysis has terminated due to out of range moments, and a list of nodes is given that are problematic. These can now be investigated.

One aspect to the above is the process of investigating these messages. It can often be time consuming to follow this procedure:

1. Open the output file in a text editor.
2. Search for errors and warnings.
3. Make a note of entity ID's relating to errors and warnings.
4. Investigate those entities in a pre-processor.

This procedure is often worth automating and incorporating into your checking process. In Oasys PRIMER there is a tool for scanning and reading the above information from these files and showing the information to the user in a tree view. This allows you to see all the errors and warnings in one panel, and investigate the entities referenced directly in the model:



The above means you can directly visualise/edit any entity from the error/warning message given in the output files – i.e. you don't just get the messages, the referenced entities are mined from the output files and you have direct access to view and edit these entities from the list of error/warning messages.

The above can also be easily carried out automatically as part of a batch automatic process if required.

## Post-Analysis Checking

Checking your model after the analysis is an important part of the checking process. Are the results realistic? Is there any instability in the model? There are many things that require checking at this stage, so in this paper we will consider two aspects of this:

- Automatic post-processing.
- Checking output results and relating them back to the original input deck (feedback).

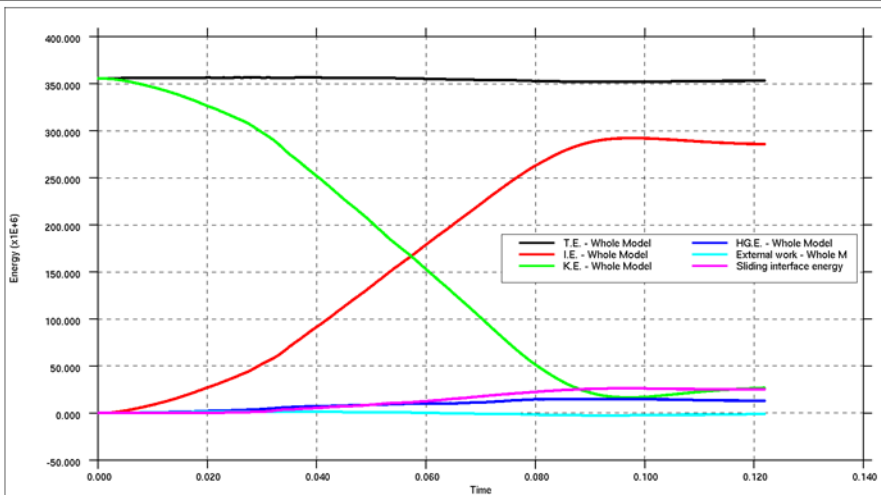
### *Automatic post-processing*

Some level of automatic post-processing is often greatly beneficial. Firstly, in retrieving key results without having to go through the manual process of extracting them each time (for example extracting occupant injury results in a vehicle crash analysis) will eliminate possible user errors and save time. In the context of checking, automatic post processing can allow you to very quickly view key metrics which can tell you how “healthy” your analysis was. Below is an example of a simple report (produced by Oasys REPORTER) automatically at the end of an analysis:

The report gives some key information about the analysis (version of LS-DYNA used, number of cores, timestep/add mass information). It also gives an energy balance graph. Automatically producing this type of information so the engineers can quickly review the key “model health” information can be a key time saving exercise. Oasys REPORTER also has many standard templates for EuroNCAP, USNCAP, IIHS etc. to allow you to automatically post-process standard loadcases in a few simple steps.

### Model Data Check :

|                      |  |                              |
|----------------------|--|------------------------------|
| Model TITLE :        | ori_dummy1 + HUMANETICS SID-IIS SBL-D VERSION V3.3.3 S3 (MM,KG,MS) |                              |
| Machine              | atnode26 using 8 CPUs  |                              |
| Platform             | Linux RHEL 5.4   | Platform-MPI 8.1.1 Xeon64 uo |
| LS-Dyna version :    | mpp s R7.1.2 (95028)   | Single precision (I4R4)      |
| Start time :         | 10/17/2016 16:02:17  |                              |
| End time :           | 10/17/2016 21:27:43  |                              |
| Elapsed Time         | 5 hours 25 min. 26 sec.  |                              |
| CPU time :           | 5 hours 25 minutes 26 seconds                                      |                              |
| Total mass :         | 0.22721014E+01   |                              |
| Initial added mass : | 1.9199E-01   |                              |
| Final added mass :   | 1.9548E-01   |                              |
| Termination status : | Normal   |                              |
| Model Directory :    | C:/Users/Sujeevan.Gnanasegara/Desktop/Models_01/QA/RUN_04/         |                              |
| Keyword file :       | input_sb4.key  |                              |



input\_sb4

Insert your company logo here

1 of 3

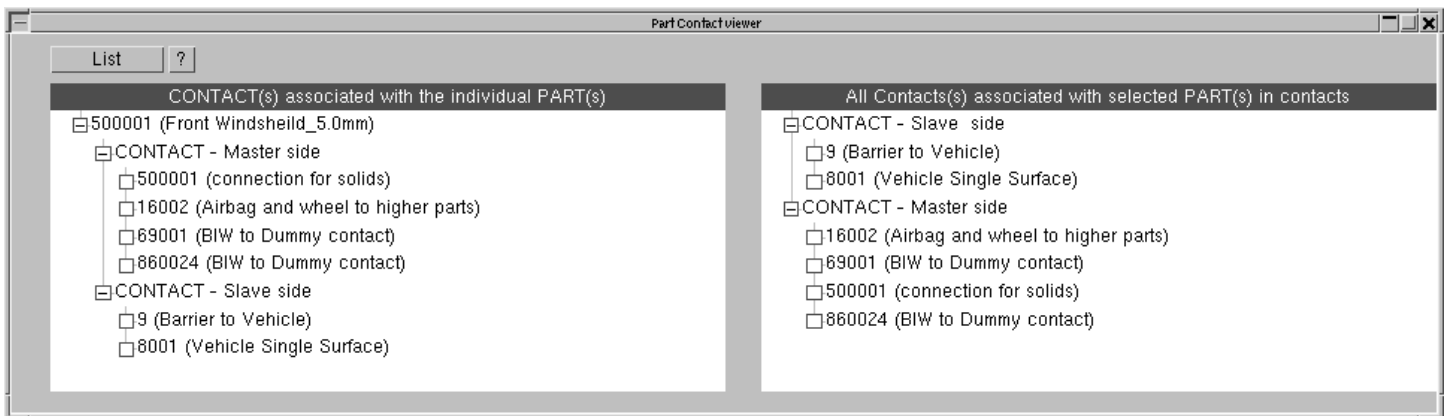
### Feedback of results to input deck

Invariably there will be a degree of manual checking when viewing analysis results. In these situations you often need to refer back to the input keyword deck for model information. In the latest release of the Oasys LS-DYNA environment software, the pre and post processing tools can be linked so that information is shared. This allows more efficient checking. Consider the following examples:

## Example 1 – Model fails to run – excessive displacements/velocities of nodes

You are running a LS-DYNA model. The model has terminated due to out of range velocities. The first thing would be to look at the nodes mentioned in the output file – as described in the previous section. Next, you may look at the d3plot output files to identify problems. After you have identified the problem parts/elements you want to find the cause as quickly as possible. This can be time consuming as the sources of information are not coupled together natively (keyword file is separate to d3plot file which is separate to binout file for example). In the Oasys software you can couple together the pre and post processor packages to share information back and forth, which makes it easier and quicker to investigate issues in your model. For the above example you can:

- 1) Identify the problem part in d3plot file in Oasys D3PLOT (Oasys program for viewing d3plot output data).
- 2) Right click on part->open part contact information directly in Oasys PRIMER. This accesses the keyword model information and the tool within Oasys PRIMER to identify all the contacts that the problem part is referenced by. This is then presented in a tree format panel:



- 3) You can now easily bring up contact energies for only the ones related to the problem part in Oasys T/HIS – a graphical plotter, which is also linked to Oasys D3PLOT.
- 4) If any of the energies show spikes then the contact keyword data can be directly accessed from energy curve.
- 5) If the contact keyword information looks OK, you can then turn your attention to the contents of the contact. Looking at internal and hourglass energy of parts is easy with D3PLOT and T/HIS linked.
- 6) If a part is attracting a lot of hourglass energy you can then directly access the part keyword information – the part may have type 2 shells – this could be the cause of the instability.

## Example 2 – Interrogating model data

You may be looking at the performance of your model, perhaps looking at load-paths through your model. You are looking at the native stresses and strains, but you want to confirm how close to yield/failure you are. By linking directly to the material information for parts from your Oasys D3PLOT session, you are able to plot utilisation. Maybe you see some high utilisation in particular areas, but it looks like some of your connections are not working as intended. You can investigate these directly using the “attached” feature in Oasys D3PLOT which has information about connection types (rigid bodies, NRB’s, tied contacts) transferred from the keyword model in Oasys PRIMER to Oasys D3PLOT. At this stage you may want to look at more information about the

parts in question, so from Oasys D3PLOT you can directly bring up the part table in Oasys PRIMER for the selected parts, which contains a great deal of information on the parts (material properties, mass, timestep, added mass, element formulation, number of integration points etc.) – an example of this is shown below.

| Part ID | Part type | Sect ID | Sect Gauge  | Mat ID  | Mass (Dyna added)<br>[0.144073] | Sect Elform | Sect NIP    | Smallest TS         |
|---------|-----------|---------|-------------|---------|---------------------------------|-------------|-------------|---------------------|
| P100146 | SHELL     | 100156  | 2.000000    | 575422  | 7.03735e-05                     | 16          | 5           | 4.120e-07 (S:10688) |
| P100160 | SHELL     | 100160  | 2.000000    | 1210007 | 2.59326e-06                     | 16          | 5           | 7.872e-07 (S:10198) |
| P100161 | SHELL     | 100161  | 2.000000    | 1210007 | 2.11428e-07                     | 16          | 5           | 8.542e-07 (S:10198) |
| P100165 | SHELL     | 100165  | 2.000000    | 575422  | 1.49942e-05                     | 16          | 5           | 6.704e-07 (S:10185) |
| P100167 | SHELL     | 100167  | 2.000000    | 575422  | 3.95853e-06                     | 16          | 5           | 6.380e-07 (S:10177) |
| P100168 | SHELL     | 100168  | 2.000000    | 575422  | 2.47566e-06                     | 16          | 5           | 5.573e-07 (S:10175) |
| P100268 | SHELL     | 100268  | 2.000000    | 575422  | 5.54898e-05                     | 16          | 5           | 2.551e-07 (S:10424) |
| P100270 | SHELL     | 100270  | 2.000000    | 575422  | 3.16412e-05                     | 16          | 5           | 5.984e-07 (S:10418) |
| P101960 | SHELL     | 100268  | 2.000000    | 20002   | <undefined>                     | 16          | 5           | <undefined>         |
| P101961 | SHELL     | 100165  | 2.000000    | 20002   | <undefined>                     | 16          | 5           | <undefined>         |
| P150902 | SOLID     | 150902  | <undefined> | 910002  | 0.063788                        | 1           | <undefined> | 1.716e-07 (H:10000) |
| P150903 | SOLID     | 150903  | <undefined> | 910003  | 0.038938                        | 1           | <undefined> | 2.165e-07 (H:12069) |
| P150904 | SOLID     | 150904  | <undefined> | 910004  | 0.021149                        | 1           | <undefined> | 2.589e-07 (H:12048) |
| P150905 | SOLID     | 150905  | <undefined> | 910005  | 0.0121955                       | 1           | <undefined> | 2.341e-07 (H:12046) |
| P150906 | SOLID     | 150906  | <undefined> | 910006  | 0.00355463                      | 1           | <undefined> | 3.111e-07 (H:12047) |
| P150912 | BEAM      | 101635  | <undefined> | 606036  | 0.000592898                     | 9           | <undefined> | 8.288e-08 (B:23500) |
| P151002 | SHELL     | 151001  | 2.000000    | 1100019 | 0                               | 16          | 5           | 9.328e-07 (S:15168) |
| P151003 | SHELL     | 151002  | 8.000000    | 1100019 | 0                               | 16          | 5           | 1.427e-06 (S:15190) |
| P151901 | SOLID     | 151901  | <undefined> | 1100022 | 0                               | 1           | <undefined> | 1.595e-05 (H:15102) |

## Summary

This paper has shown that checking a LS-DYNA model can be a complex process, and should be considered at all stages of the analysis process:

**Pre-analysis** – Quite often at this stage the internal checks of a pre-processor are relied upon, but they should also be supplemented with company specific checks and automatic checking processes to ensure consistent, “healthy” models before submission into LS-DYNA.

**Checking during the analysis** – LS-DYNA’s internal checks and outputs are a useful source of information on model health. Processing this information as part of the checking process can be beneficial.

**Post-analysis** – This can sometimes be overlooked due to the pressure to produce results. Spending time investigating model issues at this stage can be hugely beneficial though, particularly at the start of model development. Having access to all model data at this stage and linking back to the keyword input model can also be beneficial.

Overall having a checking process that can be followed through from model creation, during analysis, post analysis and then fed back to the input model can lead to consistently healthy models across engineers in larger teams. Automating that process somewhat can also eliminate human error and again promote consistency, as well as reducing “wasted” analysis hours on problematic keyword decks.