

New Generation Modeler for LS-DYNA Material Parameter Conversion

H. Lobo, E. Strong

¹Matereality, New York, USA

1 Introduction

LS-DYNA software is supplied with a large number of material models that permit the description of a variety of materials and their physical behavior. The models range from the simple to the complex, depending on the material and the extent to which the physical behavior should be captured by the simulation. Some of the models directly accept material properties and require little or no material parameter conversion. Many commonly used material models, however, do not accept material data as measured, but require some form of conversion process to make the material parameters needed for the given material model. This process can be straightforward or complex depending on the model. In straightforward conversion processes, the material model parameters can be derived by a sequence of mathematical conversion steps. Complex conversion may require iteration, subjective data interpretation, and sometimes, the use of optimization routines to deliver parameters based on a best fit.

Comprehensive software documentation provides details of the requirements of each material model. Sometimes, however, there is uncertainty due to differing terminology or multiple interpretations of how to calculate a particular parameter. These can lead to loss in simulation fidelity and increased risk, particularly at a time when simulation is increasingly being relied upon for design decision making. Attempts to standardize material parameter conversion processes would be of great benefit to increase reliability and improve efficiency.

In previous work [1], we have shown the benefit of using software for the conversion process, where a dedicated code improved the efficiency and consistency of the conversion process while eliminating common errors, such as unit conversion or incorrect calculations. While these software modelers were effective, they did not address needs such as model extrapolation that is commonly used in crash modeling today.

In the current work, we describe a new software component that has been built to handle the unique, multivariate nature of LS-DYNA material models. Rate-dependent models require adjustment and tuning of many material parameters to fit the rate-dependent tensile properties. Drawing upon a robust back-end data model, a graphical user interface provides drag-and-drop capability to allow the user to perform tasks such as model extrapolation beyond tested data, modulus change, rate dependency tuning, and failure criteria adjustment, while maintaining self-consistency of the underlying material model. Because of the esoteric units often encountered in crash simulations, unit system conversions are also facilitated, eliminating error and ensuring that material inputs to simulation correctly reflect the intent of the CAE analyst. The utility of the Matereality CAE Modeler is illustrated with examples for LS-DYNA material models MAT_019, MAT_024 and MAT_089 LCSR. It should be clarified here that the purpose of this software is to aid in the mechanics of creation of a candidate material card. It does not provide an assurance of the quality of the material model. Validation is a useful next step to ensure that the simulation is producing a sufficiently accurate result [2]. The CAE Modeler can be used to create multiple candidate material cards with different options and settings for validation. Poorly performing material cards would be discarded.

2 Background

The CAE Modeler for LS-DYNA is a module within Matereality, an infrastructure software that allows enterprises to build material databases of all their material data and distribute appropriate information as needed to different kinds of engineers and other users. In the case of the CAE experts in the company, they would be able to access raw material data and use the CAE Modeler to create material parameters for the target simulation solver. In the case of LS-DYNA, for example, they could access rate-dependent tensile properties and create a MAT_024 material card.

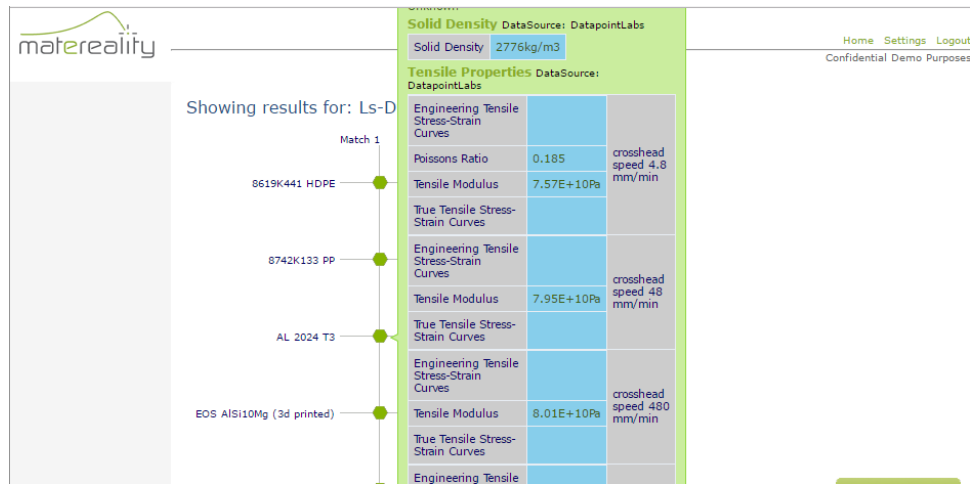


Fig.1: Mining the repository for materials with the required properties. Material models may have different material data requirements.

Other material cards could have different material data requirements, which would result in different search results. The assembled material data is processed using conversion algorithms and presented for inspection via a graphical user interface. The user can interact with the model, modifying and tuning it as needed. At the end, the Modeler writes a material file that can be immediately downloaded for use, or stored in the CAE Materials library of the Matereality software for distribution to other CAE users within the enterprise. Traceability of the source material data used to create the material card is maintained. To remove the potential of misapplication of material files, the material file is assigned to the same material as the original raw property data in Matereality software.

For distribution of CAE material files to CAE teams, the software creates the external material card libraries used by pre-processors such as Altair HyperWorks and ANSA from Beta CAE.

3 Software

The new Modeler utilizes a back-end data model to interlink all the material parameters that are used to describe the multivariate experimental data.

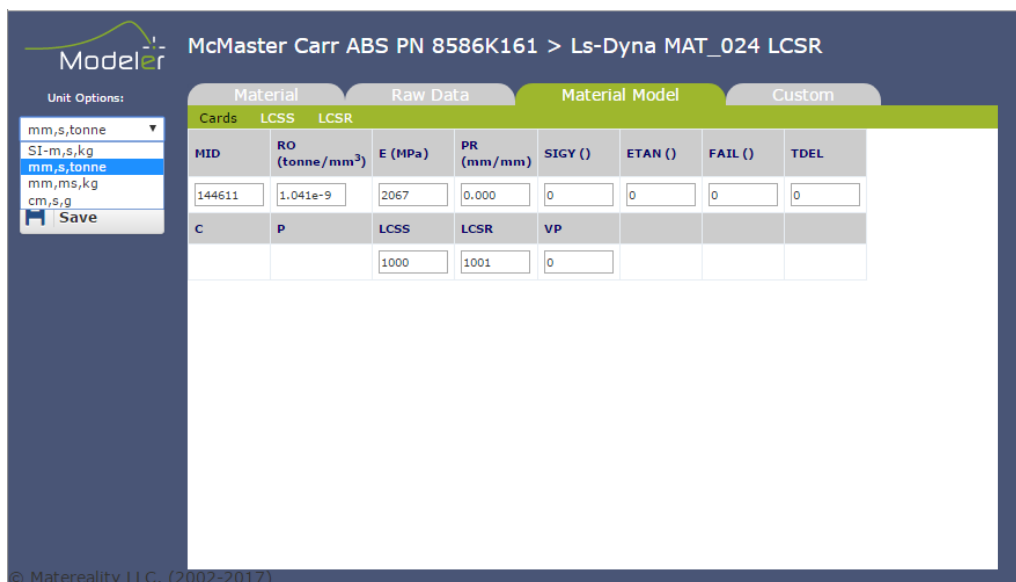


Fig.2: Selecting a unit system to ensure consistent and correct model creation

Unit system conversions are built in to ensure consistent units are used across the entire model, eliminating errors from inconsistent units. Missing data can be populated at this stage directly into the

Modeler forms. Default values are provided. Changing the modulus in this tab will affect the calculation of the plasticity curve.

Modeling of the curves is achieved by applying the material parameter conversion algorithms to the raw data as shown in Figure 3. To help with visualization, the material model is displayed in the context of the material data. Here, the user can easily assess the quality of fit.

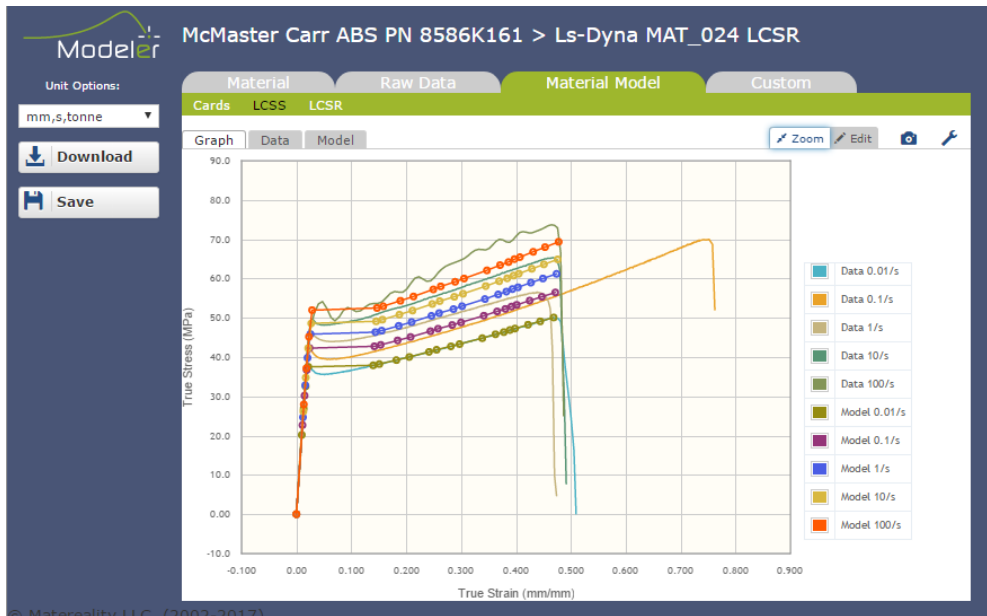


Fig.3: Automated first pass of the material model fit to the material data

The model can be tuned in a number of ways. Changing the modulus in the card will affect the initial response and also result in a recalculation of the plasticity data. It is often found necessary to extrapolate the plasticity data beyond the measured failure strain. Possible reasons for the need to do this are explained elsewhere [2].

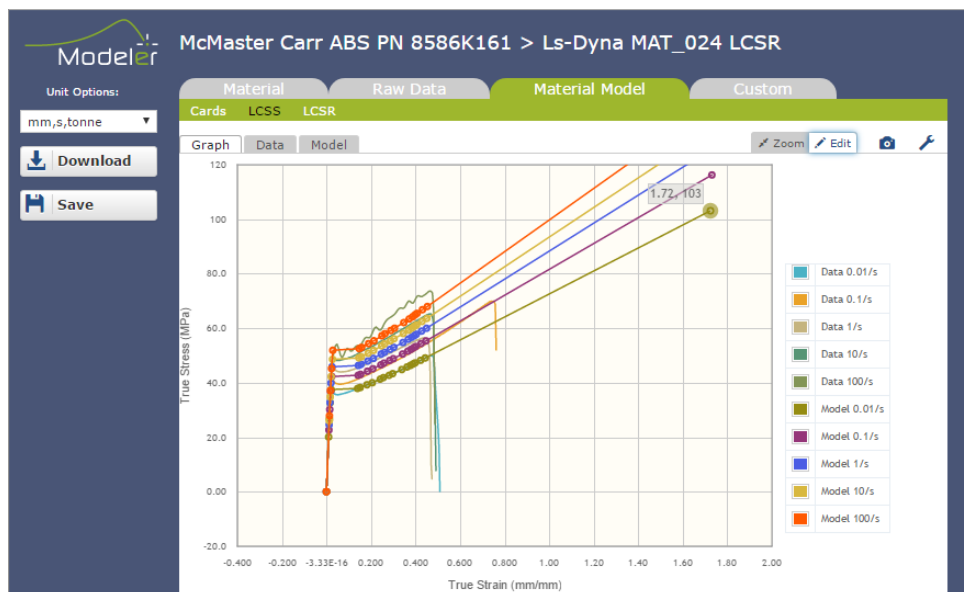


Fig.4: Extrapolation of plasticity curve using drag-and-drop.

The Modeler provides a drag-and-drop feature which makes it easy to perform this step. At this stage of development, the extrapolation is visual in nature. For the LS-DYNA MAT_024 model with LCSR rate dependency option, extrapolating the quasi-static curve will automatically display the extrapolation

for the other strain rates (Figure 4). On the other hand, if the LS-DYNA MAT_024 model with LCSS rate dependency option is used, the extrapolation needs to be applied for each strain rate.

The Modeler can also handle rate dependency tuning. In the current example (Figure 4), it may be noted that the material model is underpredicting behavior at the highest strain rate. Now, the drag-and-drop feature can be used to increase the stress ratio at 100/s to bring the material model in line with the measured data (Figure 5). In this way, a high degree of model consistency is achieved, even while working in a multidimensional plane.

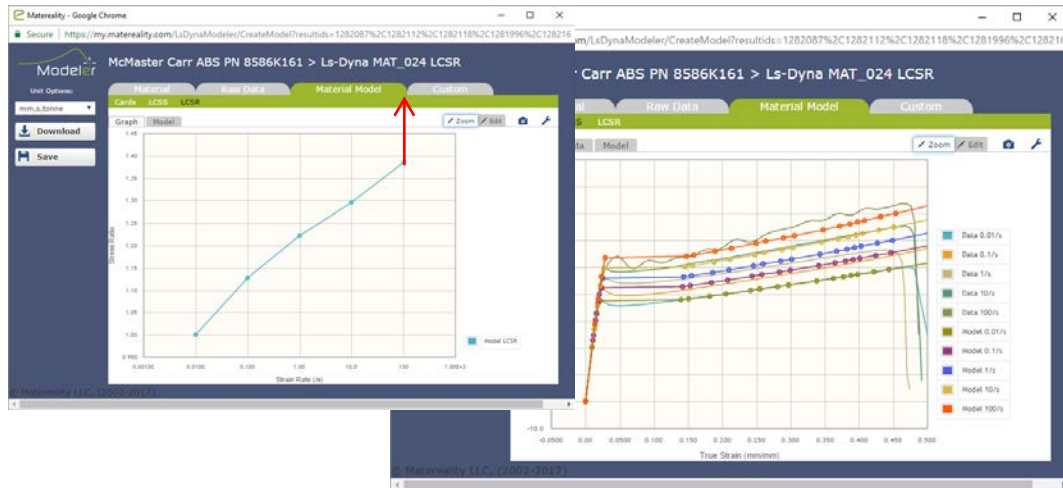


Fig.5: Changing the stress ratio modifies the display of the material curves.

Moving further to LS-DYNA MAT_089, which additionally allows the modeling of rate dependency of failure strain, this feature is particularly useful in cases where the material becomes brittle with increasing strain rate. Again, a drag-and-drop method can be used on the LCFAIL plot. Although not described, in this paper, it is easy to see how these features have also been exploited for the MAT_019 bilinear rate dependent material model, which includes failure as a function of strain rate .

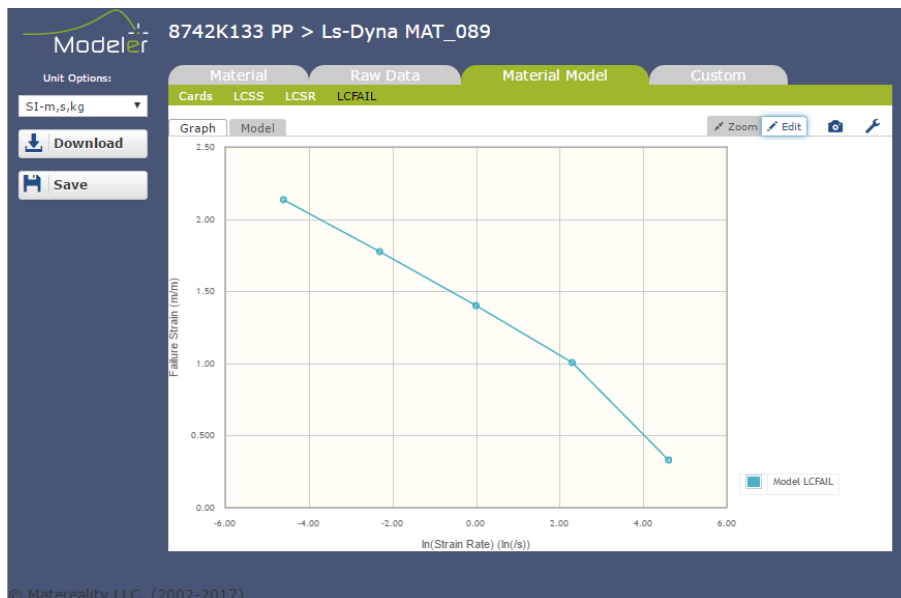


Fig.6: Modeling rate dependent failure strain in MAT_089 using drag and drop

As in previous versions of the Modeler, the software writes the material file ready for use in LS-DYNA. The material file also documents source data, material name and other characteristics for traceability purposes.

Some CAE experts have their own methods for material parameter conversion. In these cases, the Custom tab provides an Export Data button to download all the material data needed for the material parameter conversion in a standard Excel format. It is easy to adapt the expert user's method to read data from this standardized source, thereby linking external conversion methods to the data source. Such externally created material cards can then be uploaded to the enterprise CAE Materials library and are linked to the material name and originating source data for traceability purposes.

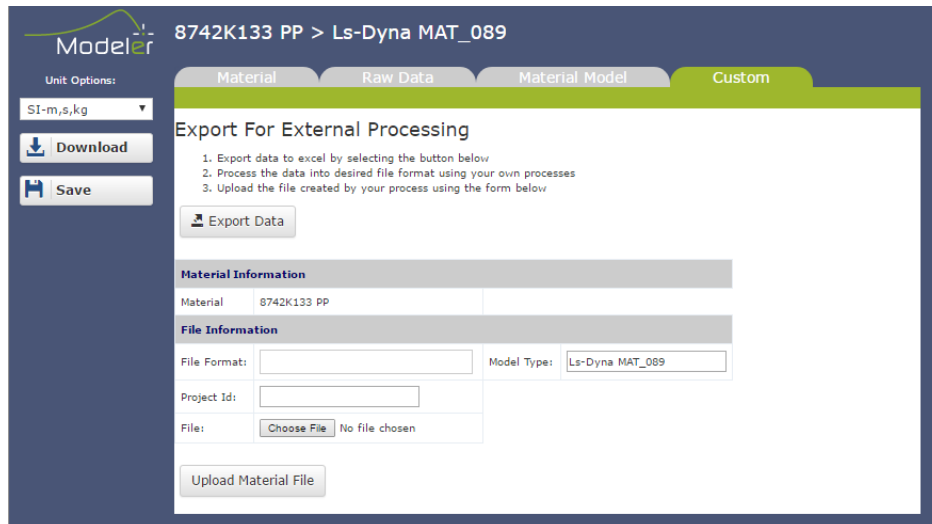


Fig.7: Using the Custom feature to download required data for external processing

4 Summary

We present an improved CAE Modeler software for material parameter conversion, which takes into consideration the unique, multivariate nature of LS-DYNA material models. Rate-dependent models require adjustment and tuning of many material parameters to fit the rate-dependent tensile properties. Drawing upon a robust back-end data model, a graphical user interface provides drag-and-drop capability to allow the user to perform tasks such as model extrapolation beyond tested data, modulus change, rate dependency tuning, and failure criteria adjustment, while assuring self-consistency of the underlying material model. Unit system conversions are also facilitated, eliminating error and ensuring that material inputs to simulation correctly reflect the intent of the CAE analyst. Matereality CAE Modelers can be used to generate material parameters for the following: MAT_019, MAT_024, MAT_089 LCSR.

5 Literature

- [1] Lobo H., 12th International LS-DYNA Users' Conference Proceedings (2012).
- [2] Lobdell M., Croop B., Lobo H., 14th International LS-DYNA Users Conference (2016)