SANDIA REPORT

SAND2014-4868
Unlimited Release
July, 2014

Development of an Improved MATLAB GUI for the Prediction of Coefficients of Restitution, and Integration into LMS

Renee N. Baca, Michael L. Congdon, and Matthew R. Brake

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online order:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2014-4868
Unlimited Release
July, 2014

Development of an Improved MATLAB GUI for the Prediction of Coefficients of Restitution, and Integration into LMS

Renee N. Baca, Michael L. Congdon, and Matthew R. Brake
Component Science & Mechanics
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS1070

Abstract

In 2012, a Matlab GUI for the prediction of the coefficient of restitution was developed in order to enable the formulation of more accurate Finite Element Analysis (FEA) models of components. This report details the development of a new Rebound Dynamics GUI, and how it differs from the previously developed program. The new GUI includes several new features, such as source and citation documentation for the material database, as well as a multiple materials impact modeler for use with LMS Virtual.Lab Motion (LMS VLM), and a rigid body dynamics modeling software. The Rebound Dynamics GUI has been designed to work with LMS VLM to enable straightforward incorporation of velocity-dependent coefficients of restitution in rigid body dynamics simulations.

ACKNOWLEDGMENTS

We would like to thank Matthew Williams (2615) for working closely with us to provide feedback from a rigid body dynamics simulation viewpoint. Douglas VanGoethem (1526) also provided key feedback about the functionality of the program and the detail of the material database. We would also like to thank Dannelle Aragon (1554) for reading over the report and providing useful critiques.

CONTENTS

FIGURES

NOMENCLATURE

CoR	Coefficient of Restitution
DLL	Dynamic Linked Library
DOE	Department of Energy
FEA	Finite Element Analysis
GUI	Graphical User Interface
IFA	Inertial Force Applicator
LMS VLM	LMS Virtual.Lab Motion
MMIM	Multi-Material Impact Modeler
SNL	Sandia National Laboratories
UDF	User Defined Force

1. INTRODUCTION

The coefficient of restitution (CoR) is a derived quantity that is used to describe impact. While highly approximate, CoR models are often used to model contact in multi-body systems such as StrongLinks and other components. In order to improve the accuracy of these models, a Rebound Dynamics Graphical Users Interface (GUI) was proposed [1] to enable easy calculation of velocity-dependent CoR curves for metals with arbitrary geometries. This Rebound Dynamics GUI is based on a well-validated, state-of-the-art model for elastic-plastic contact [2]. This model has since been used to show the effect of the choice of contact model on impact dynamics [3,4], and has specifically shown that predictions of failure are significantly different than using a traditionally static CoR model compared to a higher fidelity model, such as [2,5].

1.1 Motivation behind the additions

In 2012, a rebound dynamics GUI was created to predict accurately and efficiently the CoR. This GUI contained a materials database as well as a CoR model [1]. The CoR modeler gathered the users' inputs (geometric and material properties) and utilized them in a contact model [2], outputting CoR as a function of velocity (mm/s). This paper will focus on the additions made to the GUI over the last year.

Originally, one of the ideas for future work on the GUI in [1] was that the polynomial output of the impact model could be used as an input for the CoR in LMS Virtual.Lab Motion (LMS VLM). It was soon realized that LMS VLM was not setup to accommodate that type of input for a contact model, and thus this would not be possible, so other options were explored. A means of implementation was realized that would allow for a dynamic coefficient of restitution. This realization led to the addition of the Multi-Material Impact Modeler, which allows for multiple contacts to be computed simultaneously. It also warranted the addition of the Fortran Script output to the Single-Impact Modeler. These changes allow a user of LMS VLM to implement a dynamic CoR in multi-body dynamics models, yielding a more accurate representation of the impact dynamics. In addition, citations were added to the material database for traceability.

2. THE REBOUND DYNAMICS GUI

The Rebound Dynamics GUI is split into four main programs: the Launch Screen, the Materials Manager, the Coefficient of Restitution Modeler and the Multi-Material Impact Modeler (MMIM). Both the Coefficient of Restitution Modeler and the MMIM are based on the same principle for calculating the coefficient of restitution (CoR) for a pair of materials [2]. The difference between these two programs is that the Coefficient of Restitution Modeler is designed to analyze a single pair of materials, whereas the MMIM is devoted to computing multiple impacts simultaneously for use in LMS Virtual.Lab Motion (LMS VLM). Each program is able to be launched individually or from the Launch Screen. The Rebound Dynamics GUI will be stored at https://sharepoint.sandia.gov/sites/simulate.

2.1 Launch Screen

The Launch Screen (Fig. 1) is the first screen that most users will see when opening the program. It looks similar to the old version; however it has been expanded to include a field for the Multi-Material Modeler. To access the Launch Screen, the user may double click "Impact_Launcher.fig" from their directory or they may also choose to run it from the command line. The Launch Screen is also accessible from the three programs within the GUI.

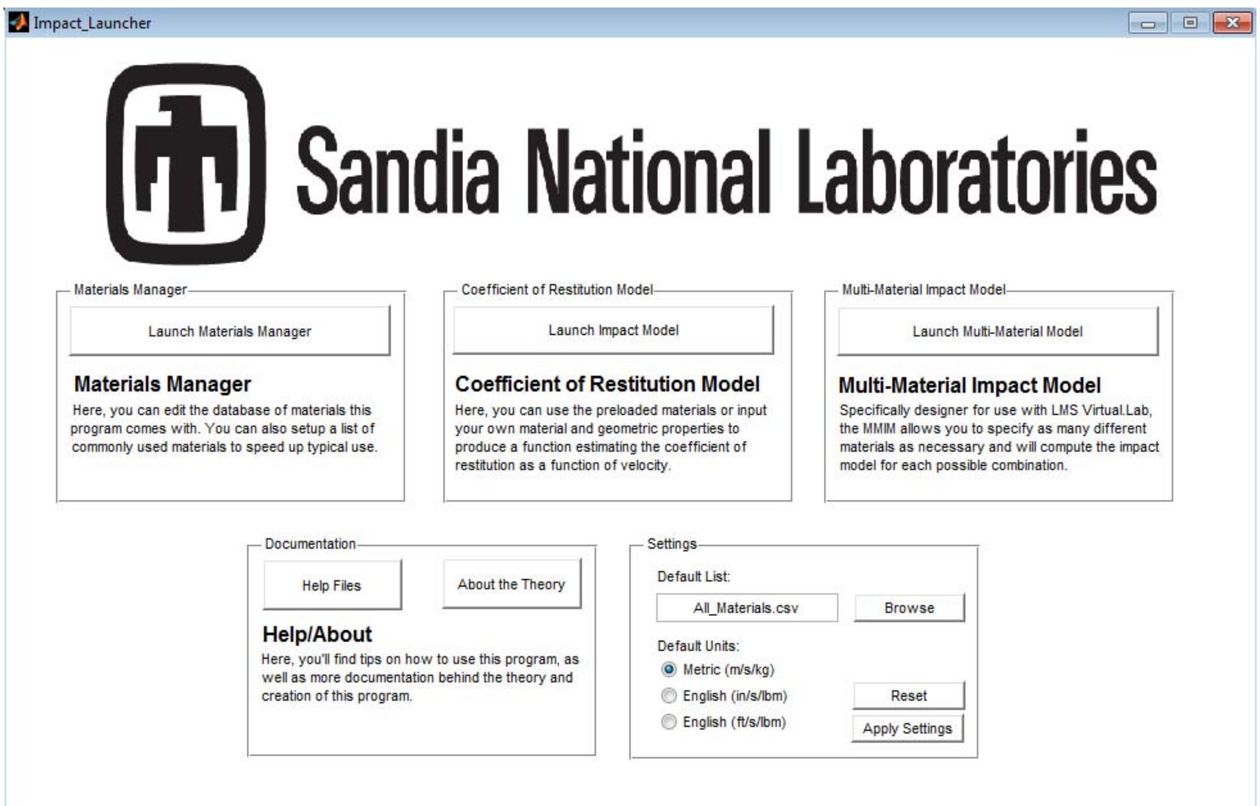

Figure 1. Launch Screen

2.2 Materials Manager

The Materials Manager program is where the user can view, edit, and sort all the materials in the database as well as add additional materials. When the program is first opened all the materials and their properties are loaded from the "All_Materials.csv" spreadsheet. Underneath the material database is the "Material Editor". This is where the changes to the material properties can be made. The right portion of the screen contains the List Manager where the user can create custom material lists to be used in the two impact modelers.

Three additions have been made to the Materials Manager (Fig. 2). A new "Source" column has been added to the material database, and two buttons have been added to the Material Editor: "Open Citation" and "Edit Citation". Additionally, several minor bugs have been corrected to improve the functionality of the program. All previous functions of the Material Manager are still present. The layout has been changed to accommodate the addition of the citation buttons as well as decrease some of the ambiguity with adding and removing materials from the List Manager. The following subsections will describe each section of the Materials Manager in more detail.

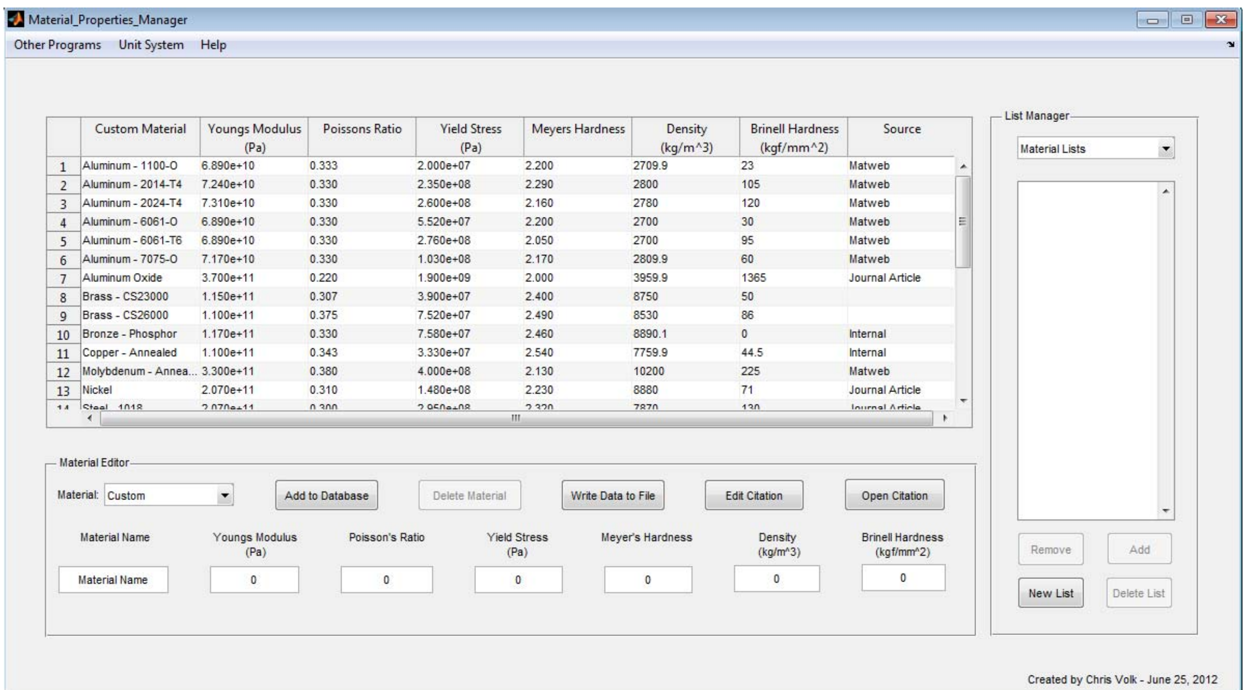

Figure 2. The Materials Manager

2.2.1 Material Database

When the program is first opened all the materials and their properties are loaded alphabetically from the "All_Materials.csv" spreadsheet. The added source column provides the user a quick glance as to where the properties originated from. Examples of existing sources include Matweb, journal articles, and internal testing done at Sandia National Laboratories.

2.2.2 Material Editor

The Material Editor is where edits to existing materials and the addition of new materials take place. To edit a material, select the material from the database above or from the dropdown menu in the Material Editor. This will populate the fields in the toolbox with the correct values. When an existing material is selected, the “Save Changes” and “Delete Material” buttons become available in the Material Editor. To add a new material, select “Custom” from the dropdown menu. From here the user can specify a name and material properties for the new material. When “Custom” is selected the “Add to Database” button becomes available and after clicking this button the new material will be saved to the database. Be aware that until “Write Data to File” is clicked the changes will not be saved to the “All_Materials.csv” file.

The two buttons added to the Material Editor (Fig. 3) handle editing and viewing the source information and links. The “Open Citation” button will open the referenced webpage or .pdf file for the selected material. For example, many of the material properties came from Matweb. When the “Open Citation” button is pressed, a webpage will open to the specific material’s webpage. If the source is listed as Journal Article, when “Open Citation” is clicked a .pdf will open with the journal article that the properties came from. If there is no citation listed in the “All_Materials.csv” file, an error message will appear stating the source could not be found.

Figure 3. The Material Editor

The “Edit Citation” button allows the user to edit the “Citation Source” and “Citation Name” information for a specific material. When the button is pressed a popup (Fig. 4) will appear that will have the current citation information automatically populated. From here the user is able to enter a website or .pdf file as the new source information. Note: PDF files should be stored in the same directory as the GUI itself. Once the user has entered the citation information, clicking “Save Data” in the Material Editor updates the material database and “Write Data to File” saves the data to the “All _Materials.csv” spreadsheet. If “Edit Citation” is pressed when “Custom” is listed as the material, a new material will be created named “Custom” and the material properties listed in the Edit Toolbox will be saved to the database. The user will then be able to edit the citation information.

Figure 4. Citation Editor

2.2.4 List Manager

The functionality of the List Manager (Fig. 5) remains the same as in [1]. Here, users can create custom lists of materials to be used in the two impact modelers. The name displayed in the dropdown menu is that of the current list. All of the materials that are stored in the particular list selected are displayed in alphabetical order. The layout of the buttons was the only change made to the List Manager. The “Add” button was moved from the Material Editor to the List Manager section to decrease possible confusion as to the functionality of this button. When creating a new list, select the desired material from the material database and press the “Add” button in the List Manager.

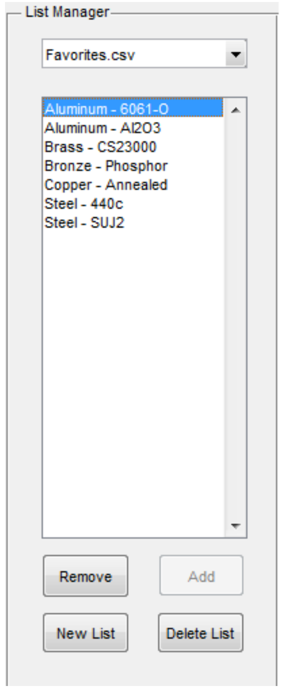

Figure 5. List Manager

2.2.4 Menu Toolbar

The Menu Toolbar, shown in Fig. 6, includes the same functions as before [1] with the addition of the MMIM in the "Other Programs" dropdown. From the "Other Programs" dropdown the user is able to switch between the MMIM, Impact Model and Launch Screen.

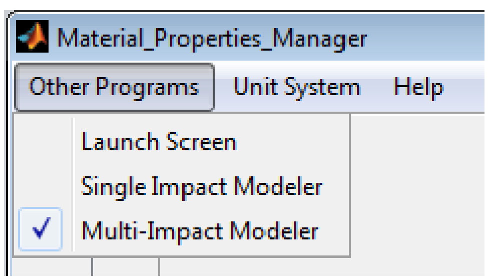

Figure 6. Menu Toolbar for Material Manager

The second option, "Unit System", allows the user to switch between Metric (m/s/kg), English (in/s/lb), and English (ft/s/lb) units. The last option is "Help"; it is here where the user can find documentation regarding the theory behind the model as well as help for the GUI itself.

2.3 Coefficient of Restitution Modeler

The Coefficient of Restitution Modeler appears graphically the same as the previous GUI; however, there are several important changes to its functionality. First, the model used for predicting the coefficients of restitution has been significantly revised to include higher order effects such as strain hardening. This change enables the program to predict more accurate CoR across a large range of velocities within a single simulation for metallic materials. Second, the "Export" button now brings up a window that prompts the user for the output data type. The three original options, "Array", "Array & Function Handle", and "CSV File", have remained, but an additional option has been added, "Fortran Script", which is discussed at length in Section 2.3.1. Note: Coefficient of Restitution Modeler and Single-Impact Modeler are used interchangeably.

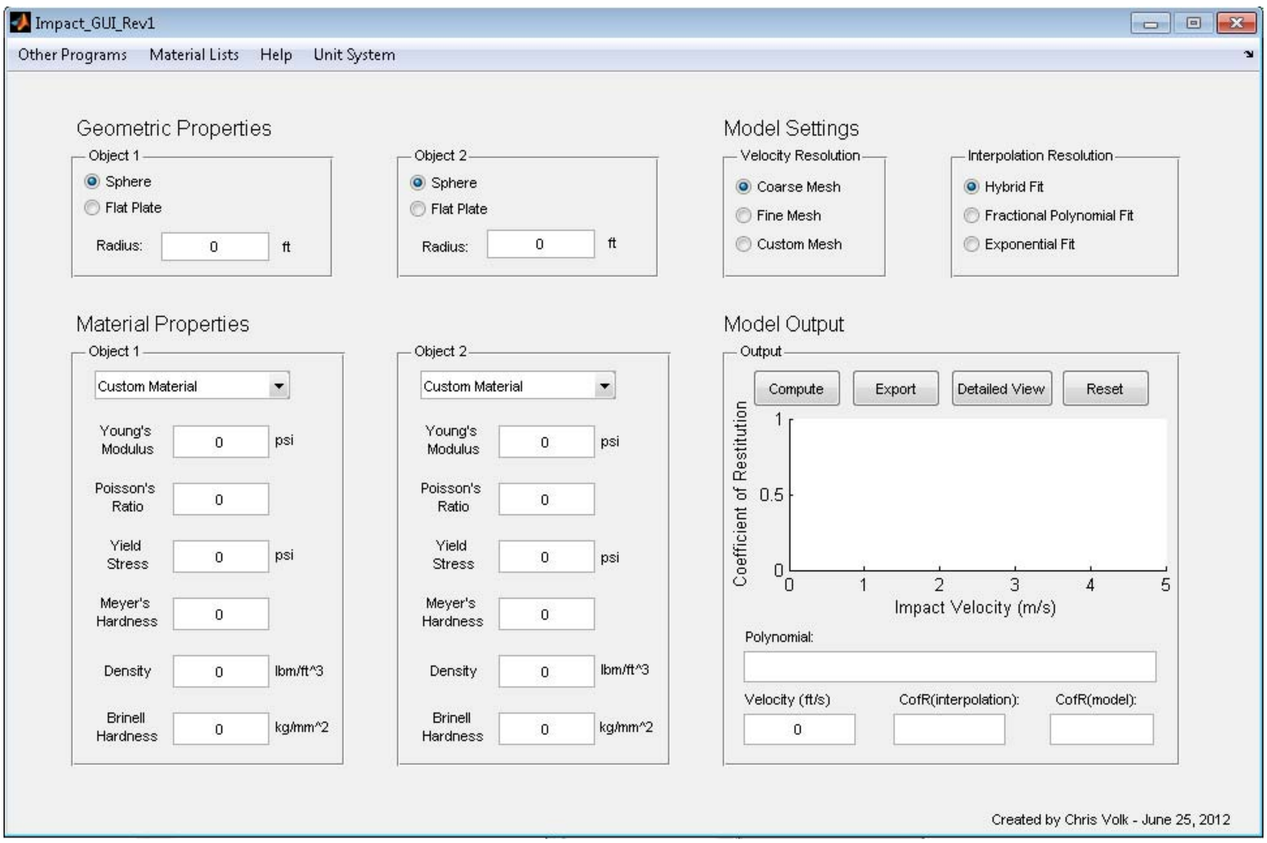

Figure 7. The Coefficient of Restitution Modeler

The Coefficient of Restitution Modeler (Fig. 7) is laid out to clearly distinguish each section of the program. In the top left corner is the geometric section where the user can select the two objects and their radii, if applicable. Directly below the Geometric Properties are the Material Properties. This is where the materials to be used in the simulation are selected from the material database. Upon selecting a material, the properties are automatically populated. In the upper right corner, the settings of the model are specified in terms of number of velocity data points to use in the CoR fit and type of fit to use (hybrid, polynomial, or exponential). Lastly, the bottom right contains all of the data output: a graph, a polynomial interpolation, and the numerical output of velocity and CoR. Like in the Material Manager, there is a toolbar where the user can easily switch between programs, change units and open help files. The "Material Lists" dropdown displays the lists that are created in the Material Manager. Upon selecting one of the lists, the materials stored in that list are displayed in the two dropdown menus under the Material Properties section.

2.3.1 Export Options

The Export Options menu, shown in Fig. 8, displays four methods for recording the simulation results: "Array", "Array & Function Handle", "CSV File", and the new option, "Fortran Script". This new option is designed specifically to integrate the CoR modeler into LMS Virtual.Lab Motion (LMS VLM) in order to implement a velocity-dependent CoR in impact modeling.

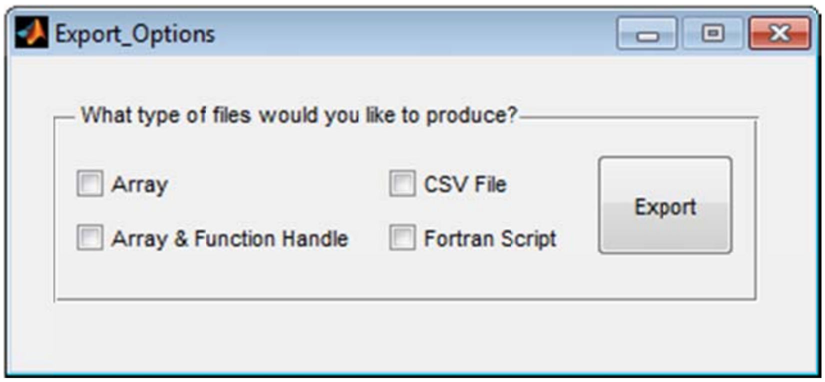

Figure 8. Export Options

To produce the desired output, select the corresponding checkbox next to each option, and click "Export". A different dialog window will appear for each selection made, prompting the user to select a folder in which to save the files. The Fortran Script option will save both an excel sheet of the data and a Fortran file with fixed-format F77 code meant for implementation into LMS VLM. However, due to certain limitations pertaining to this implementation, a second way to implement a dynamic CoR was developed that allows for more flexibility within the model, as discussed in the MMIM Section.

2.4 Addition of the Multi-Material Impact Modeler

The MMIM (Fig. 9) is designed for use with LMS VLM. This GUI is able to export Excel files containing data points for the CoR at a set list of velocities that can be used to implement a dynamic CoR in a rigid body dynamics model. It is also automated to calculate the dissipation curves for different pairs of impacting surfaces, allowing the user to save time while also specifying all of the different materials and geometries used within a model.

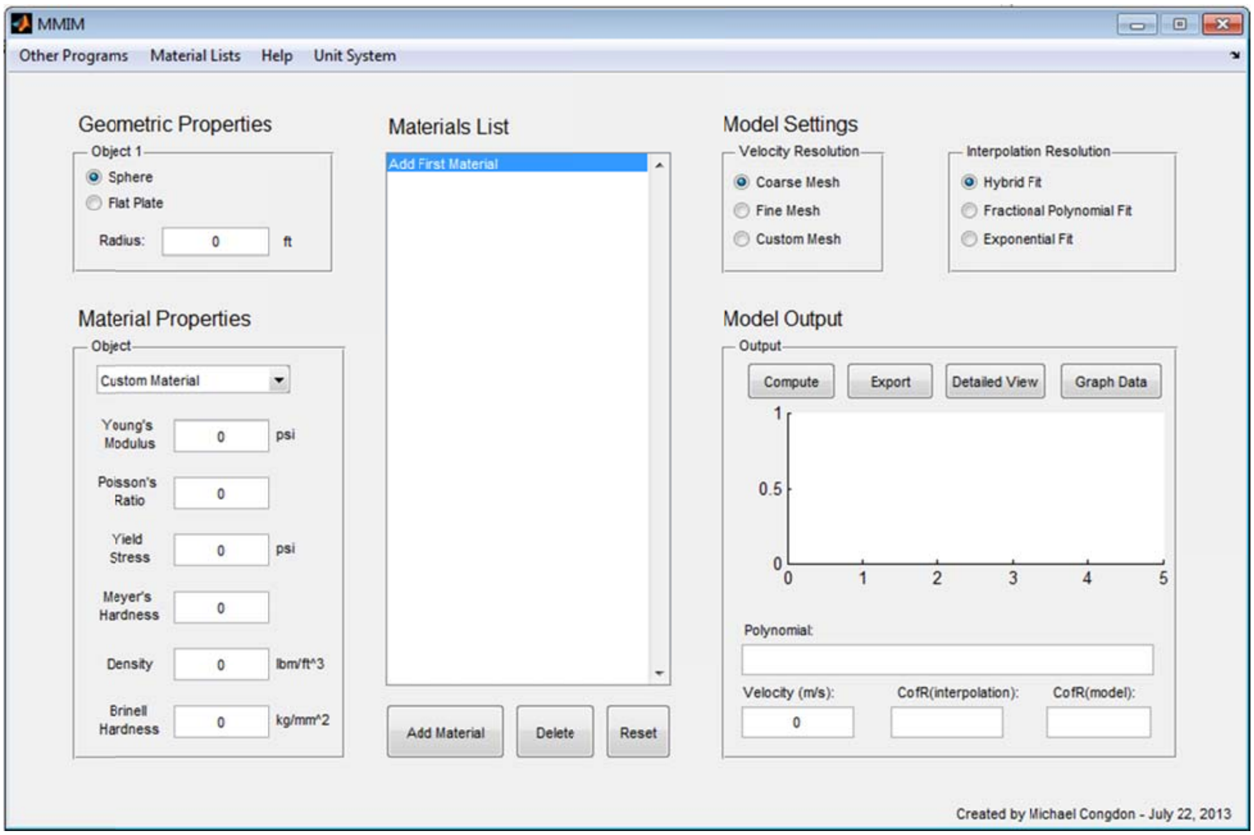

Figure 9. Multi-Material Impact Modeler

Although the MMIM is similar to the Single-Impact Modeler, there are significant differences in its functionality. The first column of the MMIM allows the user to specify a specific geometry and material combination. Once a material and geometry is specified, the "Add Material" button can be used to add this combination to the Materials List, which is the 2nd column of the MMIM. Once all material and geometry pairs have been added to the Materials List, the "Compute" button calculates the dissipation curve for every possible combination of two materials, including a material against itself for valid cases, and skipping iterations of flat plates versus flat plates. When the "Export" button is pressed, the user is prompted for a destination folder, and Excel files are generated with the data for each model and then saved within the folder. The toolbar at the top is the same as the Coefficient of Restitution Modeler. The user can easily switch between programs, access previously created material lists, open help documents and change the units.

2.4.1 Adding Materials to the Materials List

The first step in creating the Excel spreadsheets of the CoR data is to add materials to the Materials List. The Materials List (Fig. 10) in the MMIM is what displays the current selection of materials before computation, and displays the computed combinations afterwards. The “Add Material”, “Delete”, and “Reset” buttons handle the passing of selections into the list as well as deleting them from it. It should be noted, however, that the “Reset” button has a much more global scope in the MMIM, because it can also erase the contents of the box after computation, though it will not clear the graph to the right, similar to the Single-Impact Modeler. Pressing “Reset” will also clear any computations that have been made, allowing for the designation of a new set of materials.

Figure 10. Materials List

2.4.2 Viewing the Data

After clicking the "Compute" button and letting all of the calculations finish, the graph and the Materials List are populated with the data. The Materials List displays every combination that was successfully computed, shown with the title:

Material A, r = X.xxxx <u> ---vs--- Material B, r = X.xxxx <u>

The "r = …" part of each material name gives the radius so that it can be differentiated from the same material with different geometries, and "u" denotes the units. Please note that it will be replaced with "flat plate" if applicable. Initially, the graph on the right will be populated with the first contact that was computed, though any of the impacts may be viewed simply by selecting them within the list and clicking "Graph Data".

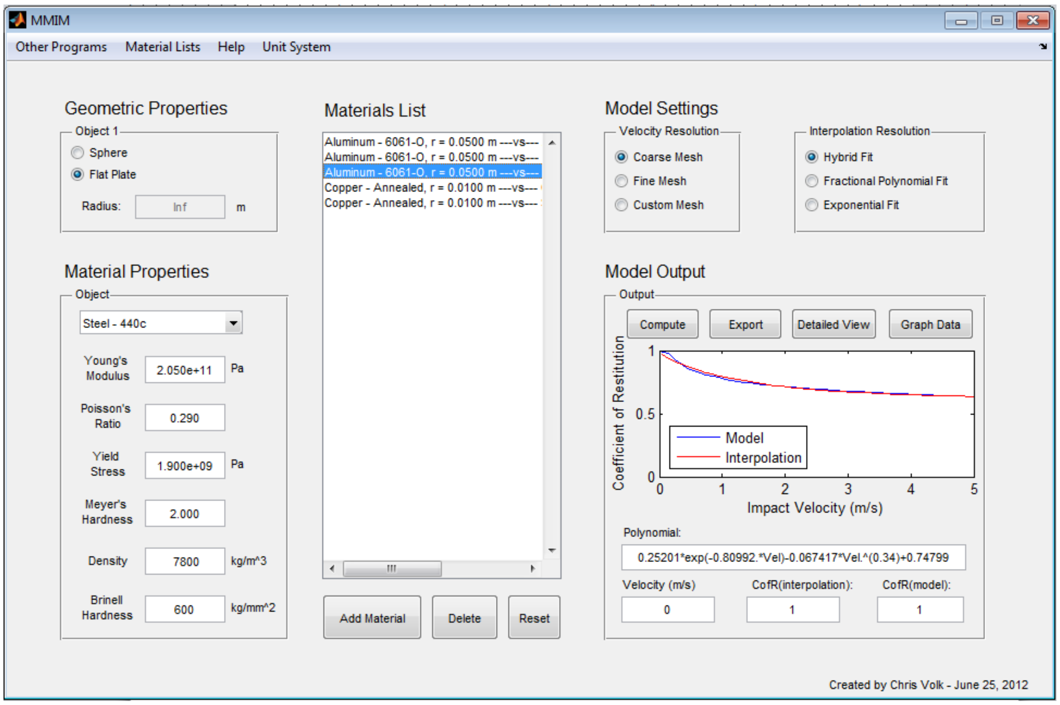

Figure 11. Example Calculation

The example shown above (Fig. 11) was calculated using aluminum and copper balls as well as a steel plate. The data shown in the graph is from the 3^{rd} data set.

2.4.3 Exporting the Data

When the "Export" button is pressed, the GUI checks to make sure that the impacts have been calculated, ensuring that the data exists and is available to be exported. Once verified, a window (Fig. 12) is brought up prompting the user to select a folder in which to save the files.

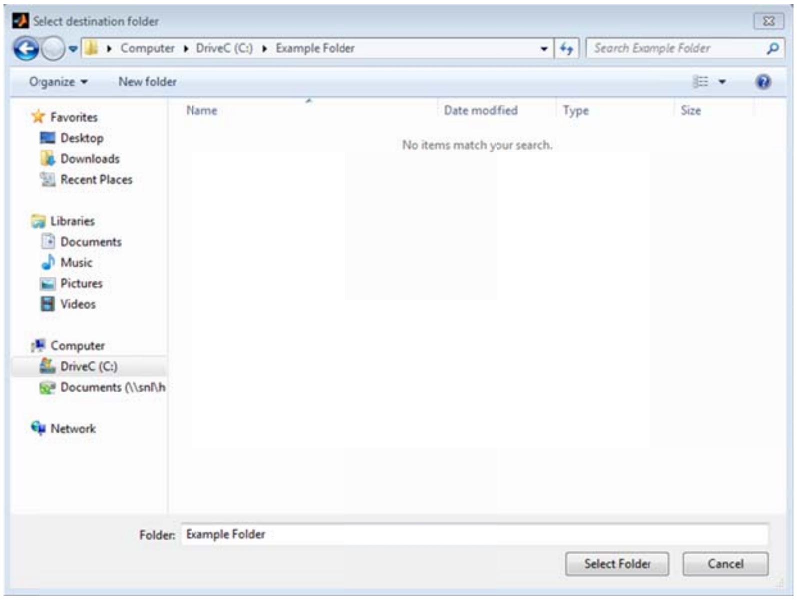

Figure 12. Example Export Folder

If no folder is selected, an error dialog is displayed stating "No folder selected, export aborted." However, the data is not deleted, and may still be viewed in the graph or exported again. If a folder is selected, then excel files are generated, one for each impact computed, and saved to the designated location. Again, the information is not deleted until the "Reset" button is pressed underneath the Materials List, meaning it can be exported multiple times to a different folder each time.

3. LMS VIRTUAL.LAB MOTION

The MMIM is designed for use with LMS Virtual.Lab Motion (LMS VLM). After generating the required data, it can be exported to a folder that can then be accessed by LMS VLM for use with its models. The modifications to LMS VLM involve User Defined Forces (UDFs), Fortran-77 scripts, and Fortran common blocks that work together to interpret, pass, and manipulate data and ultimately modify model computations. The following sections discuss the steps necessary to implement a velocity- dependent CoR within LMS VLM. A second UDF, which applies an initial force to every individual body in the model given a direction and either a spline curve or the coefficients to a time-based exponential equation, is also detailed. For the purposes of these explanations, two different models are used as examples, one for implementing the dynamic CoR, and another for the Inertial Force Applicator. It is assumed that the model is nearly complete and is ready for the UDFs to be applied.

3.1 Implementing a Dynamic CoR

An example model is used to demonstrate the implementation of a dynamic CoR. In the example, a 0.02 m radius ball is dropped onto a flat plate. For a CoR of 1, the ball will never dissipate energy on impact and thus will always bounce to the original drop height. For a CoR of 0, the ball will never bounce. For a static CoR, the ball dissipates energy at a rate independent of velocity. As a result, these models either predict settling times much faster than in reality if they are calibrated for high impact velocities, or settling times much longer than in reality if they are calibrated for low impact velocities. In what follows, five steps are presented for including a velocity dependent CoR model in LMS VLM. The order in which these five steps are presented does not have to be followed exactly; however, several steps must come in order as highlighted in the text where it applies.

3.1.1 Defining Input Data in LMS Virtual.Lab Motion

In LMS VLM, function approximations can be defined by what are known as "spline curves". These curves can either be created with user defined points or based on external data. The data exported by the GUI is constructed in such a way that they may be used as spline curve definitions without any modification. There are other ways to define functions in LMS VLM, but for the present implementation the velocity-dependent CoR must be defined via a spline curve.

> *To define a spline curve, press the "Spline Curve" button from within the "Data Definition" box on the right. If it is not visible, double click "Analysis Model" in the model tree and it should appear. A second way of defining a spline curve is via "Insert -> Data -> Spline Curve" from within the Motion Workbench.*

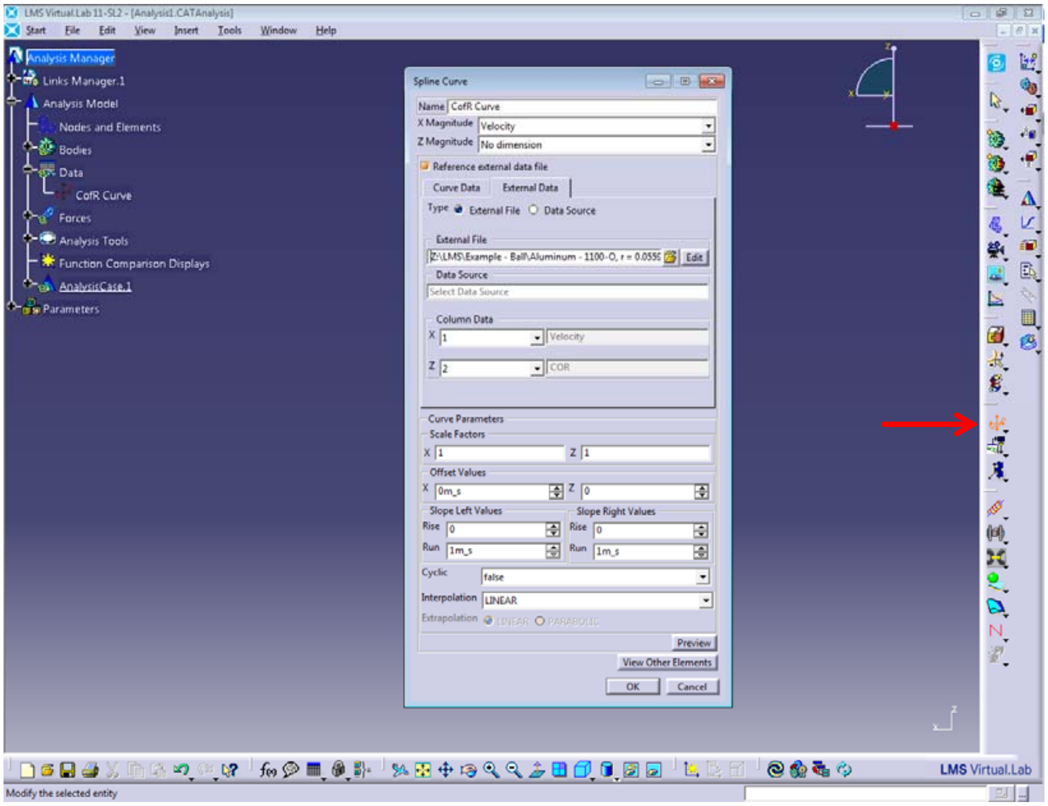

Figure 13. Spline Curve Button and Definition Box

Fig. 13 shows the button needed to bring up the spline curve definition box, and the definition box itself. *Note: The button is only orange because it is currently selected; normally it appears as a black graph with a superimposed red curve.* For the X Magnitude, select “Velocity”, and for the Z Magnitude, select “No Dimension”. Once that is done check the boxes labeled “Reference External Data,” and click the folder icon to select the desired Excel spreadsheet. In the section titled “Column Data”, the X and Z dimensions will populate with the dimensions at the top of the excel spreadsheet, “Velocity”, and “CoR”. At the bottom, make sure that the left and right slope values are set to 0 (since the CoR has asymptotic behavior), and the cyclic option is set to “false”.

3.1.2 Setting up the Customized Solver

After defining the necessary spline curves, the next step is to generate a custom solver (Fig. 14). This is used in place of the standard LMS VLM solver. To insert a Custom Solver, navigate the drop down menus to:

Insert -> Customized Solver -> Customized Solver…

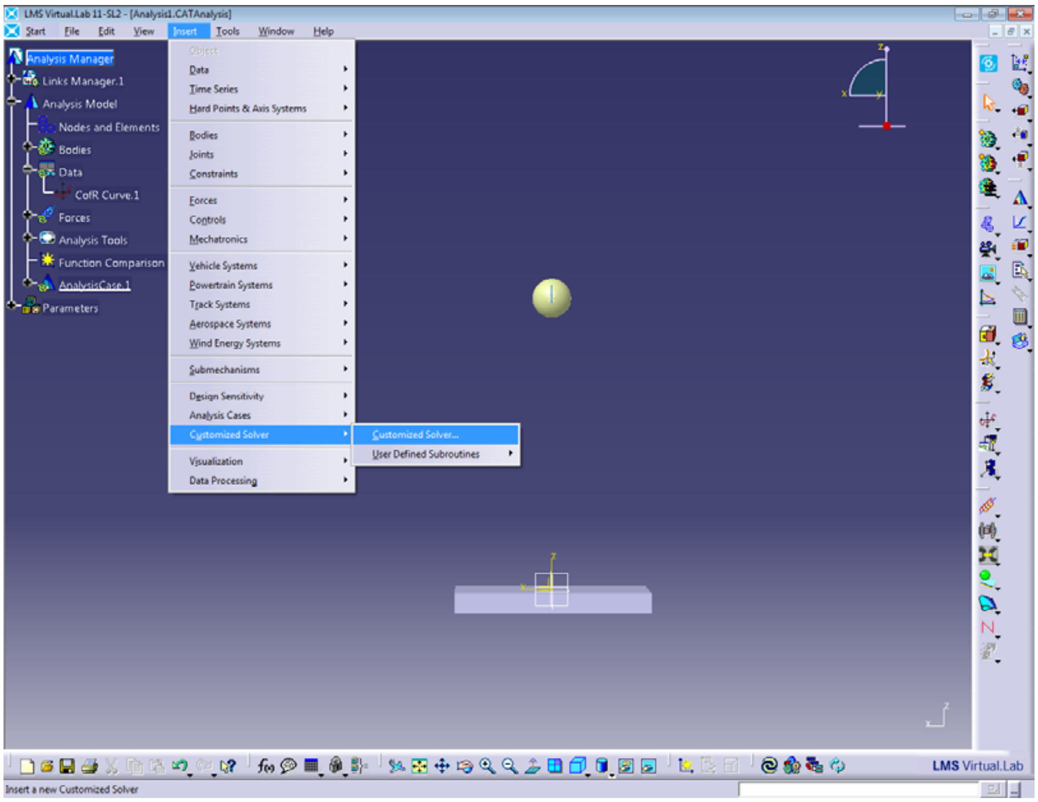

Figure 14. Inserting a Customized Solver

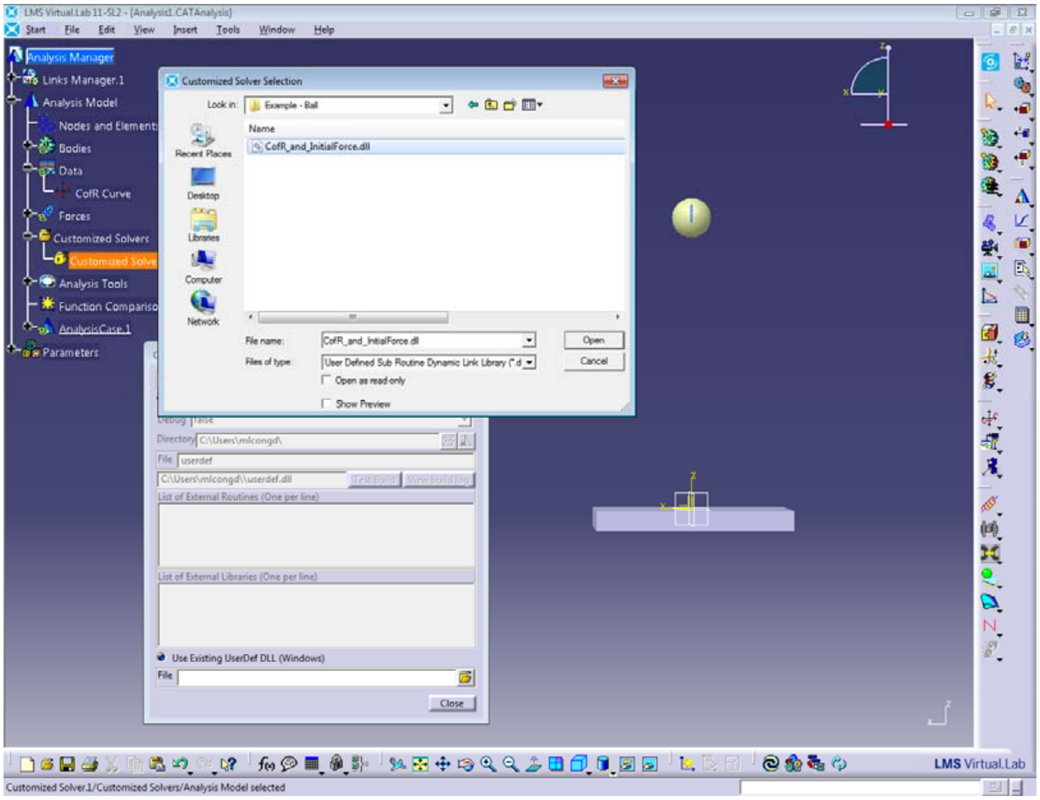

Figure 15. Customized Solver Definition Box

The DLL for the CoR and Inertial Force Applicator (IFA) (Section 3.2), aptly named “CofR_and_InitialForce.dll” is generated by the Rebound Dynamics GUI in Section 2.3 or 2.4 depending on which modeler is used. To specify the DLL generated by the Rebound Dynamics GUI, select “Use Existing UserDef DLL (Windows)” and then click on the folder icon to select the file.

The next step is to change a setting in the Solution Set that will let LMS VLM know to use the customized solver (Fig. 15) instead of the standard solver, since this is not done automatically. In the model tree, expand the “Analysis Case” branch, and double click on “Solution Set” (Fig. 16).

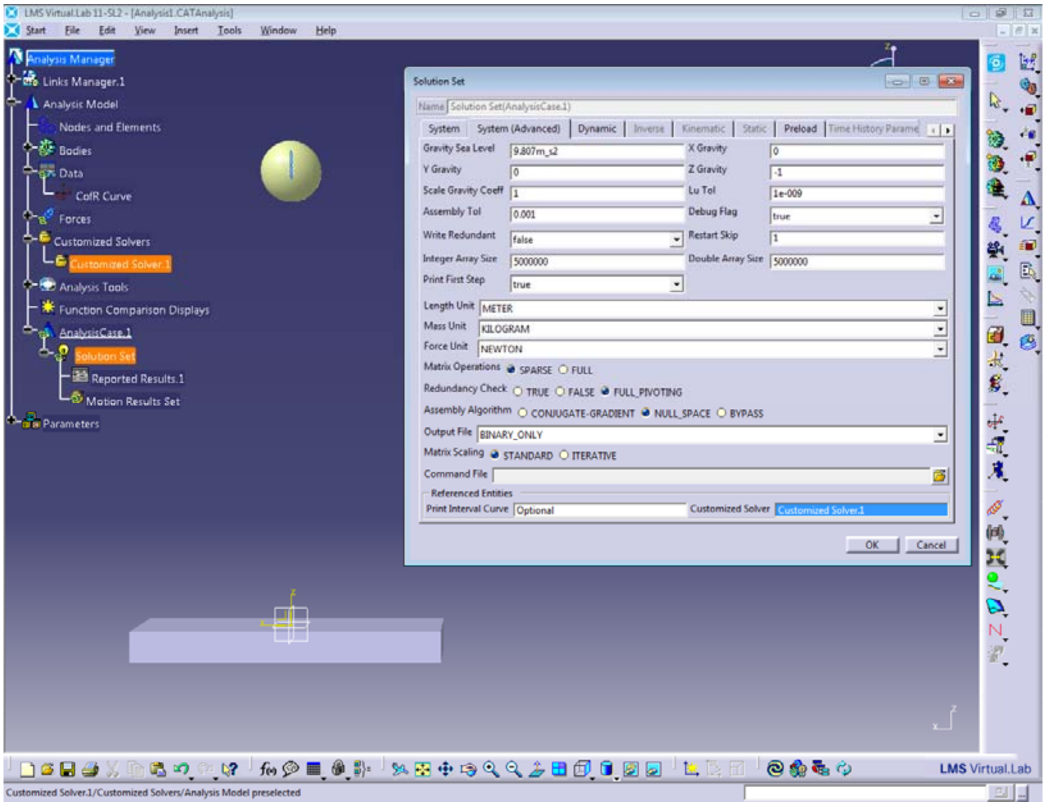

Figure 16. Solution Set

3.1.3 Inserting the User Defined Force Descriptor

The next step in setting up a dynamic CoR is to define the dialog box using the "LMSMotionUDF" file. This pairs a spline curve and a specific contact in the model such that the contact can use the spline curve to define its CoR. To set up the UDF dialog box (Fig. 17), first double click on "Analysis Model" in the model tree on the left, and then navigate to:

> *Insert -> Customized Solver -> User Define Subroutines -> User Defined Force Descriptors -> User Defined Force Type 1…*

Note: UDF Type 1 MUST be selected as the descriptor for the CoR. Due to coding conventions allowing it to be used alongside the Inertial Force Applicator, it will only work if it is of type 1.

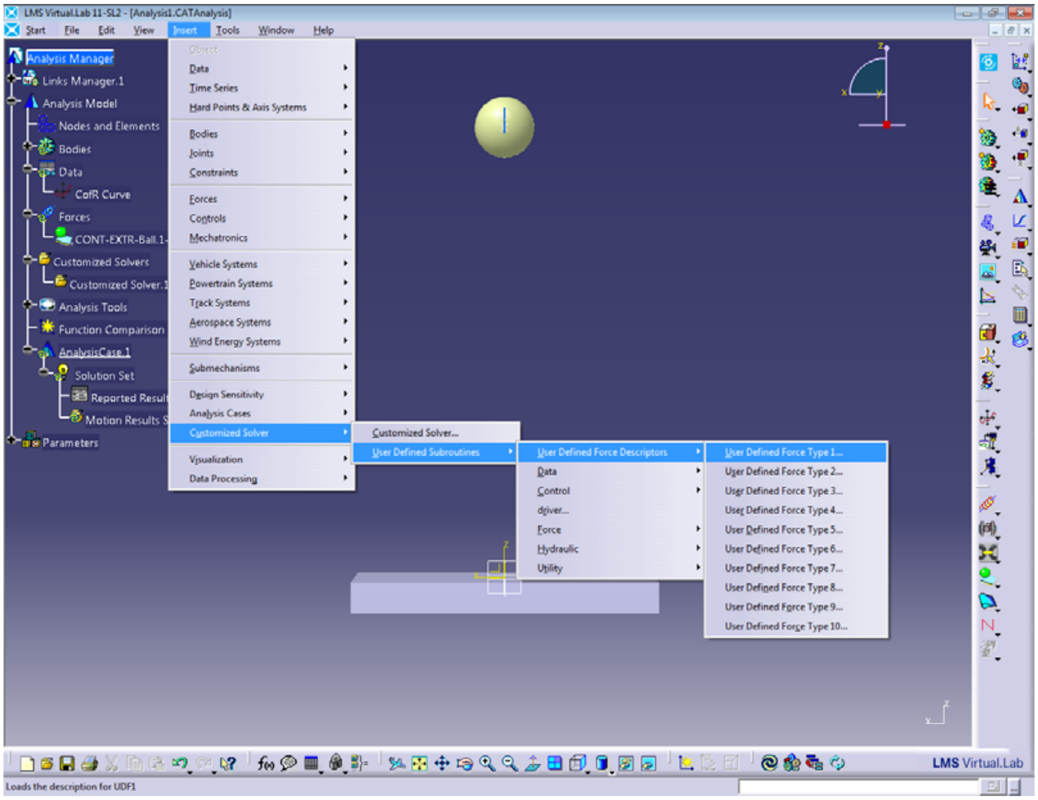

Figure 17. Defining the UDF dialog box

When “User Defined Force Type 1…” is selected, a Windows dialog box will appear so that the proper file may be selected. The definition file for the CoR is called “CofR.LMSMotionUDF”. Once the descriptor is properly set up, the UDF selection will no longer be grayed out in the forces toolbar on the right hand side. After it is defined, the UDF may be selected and applied to the model. Fig. 18 shows the dialog box that is displayed after clicking on the CoR UDF.

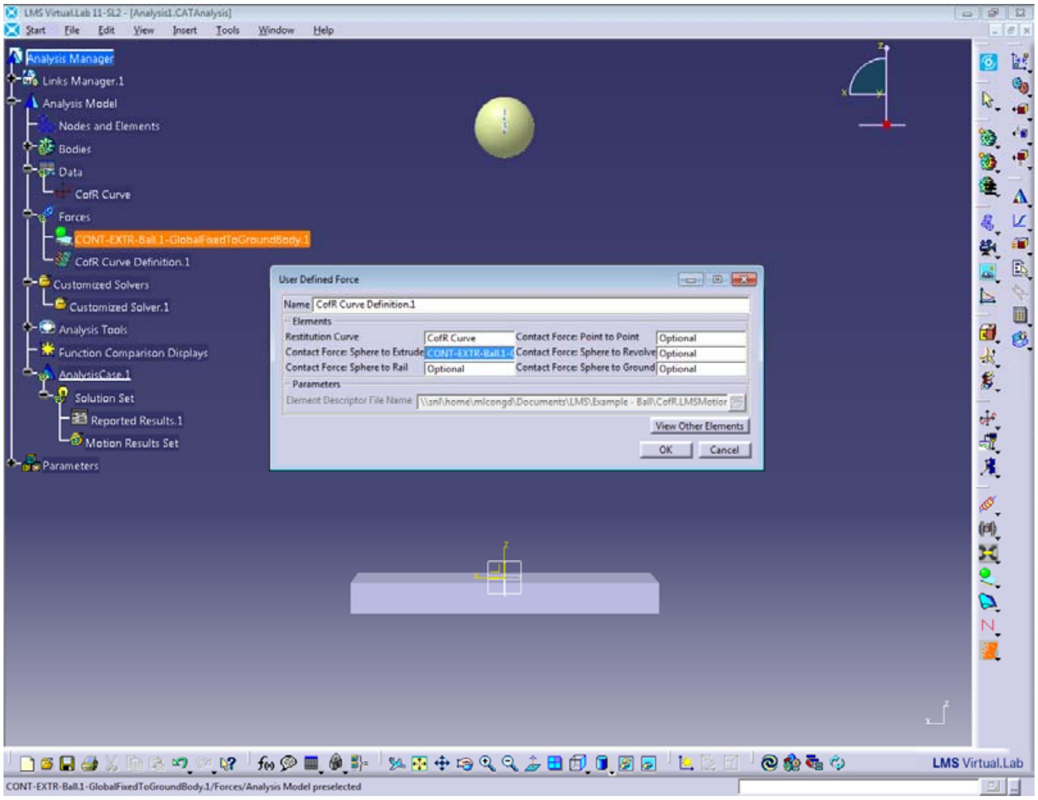

Figure 18. Coefficient of Restitution Dialog Box

The UDF for the CoR takes the input of a spline curve and one of the five types of point-based contacts in the model. For this example, CoR Curve.1 has been selected for the spline curve, and a “Sphere to Extrude” type contact has been selected to modify. The only types of contacts that may be selected are those that involve a point based impact (e.g. Sphere to Sphere, or Sphere to Flat). There is no limit on the number of contacts that can be selected, granted that they are of different types; however, the solver will only use the last contact in the list. Thus, to remove any uncertainty in terms of what model is used, the best practice is to select only one contact type. The selection of a spline curve is mandatory, and the UDF will not be created unless one is selected. This requirement does not hold for the contact; however, if no contact is selected, the UDF will never be called. The function for the dynamic CoR follows the equation

$$CoR = f(VPEN)$$

Where *VPEN* is the velocity of penetration of two bodies, and f(x) is the spline curve. There is a corresponding velocity, *VSLID*, which is unused in this calculation. Because the velocity of penetration of two bodies is dynamic throughout the collision, and the CoR depends only upon the initial impact speed, the first value of VPEN passed during each collision is saved until a new collision is detected for that contact-UDF pair. This approach is an approximation to the true dynamics, which would model the actual contact forces during an impact event; however, from the perspective of rigid body dynamics, this approach is energetically consistent and correct with the largest source of error being due to impact durations being significantly shorter than predicted by high fidelity models. This is the inherent limitation in using a CoR though.

3.1.4 Computing the Modified Solution

After setting the Solution Set to use the customized solver (Section 3.1.2), the solution may be computed as normal by either right clicking on the Solution Set and selecting "Compute", or highlighting the Solution Set and clicking on "Compute" from the toolbar on the right.

3.1.5 Model Analysis and Verification

To verify that the CoR is applied correctly, a second model is shown in Fig. 19 with four balls bouncing side by side. The two balls on either end were customized with data from the Rebound Dynamics GUI, while the center two are considered the controls. Of those, the center-left ball has no modification to its contact algorithm, and uses a CoR of 0.8. The center-right ball has its CoR defined by a spline curve that is also set to be flat at 0.8, therefore it should perfectly mirror the previous ball. This model is also used to verify the IFA, which is discussed at length in the next section. In this demonstration, the balls are dropped from a higher elevation onto the platform shown.

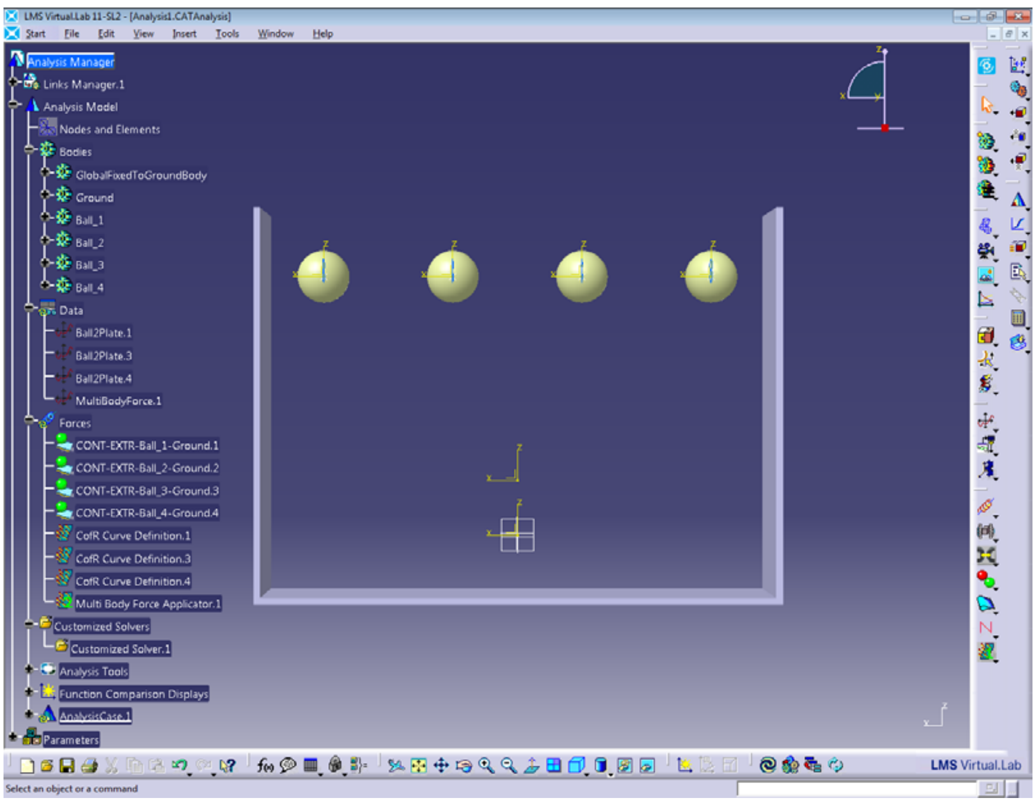

Figure 19. Verification Model

After computing the solution to the model, the velocity and position data for each ball is obtained, and the relative error (Fig. 20) from the expected CoR is calculated for the first four impacts.

Ball 1

Height				
h1	h2	CofR	Expected CofR	Error
1.1	0.823	0.8650	0.8563	1%
0.823	0.679	0.9083	0.8618	5%
0.679	0.522	0.8768	0.8680	1%
0.522	0.404	0.8797	0.8768	0%
Velocity				
v1	v2	CofR	Expected CofR	Error
4.632	4.008	0.8653	0.8563	1%
3.989	3.626	0.9090	0.8618	5%
3.606	3.19	0.8846	0.8680	2%
3.139	2.756	0.8780	0.8768	0%

Ball 2

Height				
h1	h2	CofR	Expected CofR	Error
1.1	0.699	0.7972	0.8000	0%
0.699	0.441	0.7943	0.8000	-1%
0.441	0.293	0.8151	0.8000	2%
0.293	0.186	0.7968	0.8000	0%
Velocity				
v1	v2	CofR	Expected CofR	Error
4.642	3.683	0.7934	0.8000	-1%
3.696	2.937	0.7946	0.8000	-1%
2.937	2.39	0.8138	0.8000	2%
2.396	1.912	0.7980	0.8000	0%

Ball 3

Height				
h1	h2	CofR	Expected CofR	Error
1.1	0.699	0.7972	0.8000	0%
0.699	0.441	0.7943	0.8000	-1%
0.441	0.293	0.8151	0.8000	2%
0.293	0.186	0.7968	0.8000	0%
Velocity				
v1	v2	CofR	Expected CofR	Error
4.642	3.678	0.7923	0.8000	-1%
3.696	2.943	0.7963	0.8000	0%
2.937	2.414	0.8219	0.8000	3%
2.396	1.912	0.7980	0.8000	0%

Ball 4

Height				
h1	h2	CofR	Expected CofR	Error
1.1	0.856	0.8821	0.8683	2%
0.856	0.649	0.8707	0.8786	-1%
0.649	0.526	0.9003	0.8879	1%
0.526	0.424	0.8978	0.8950	0%
Velocity				
v1	v2	CofR	Expected CofR	Error
4.596	4.078	0.8873	0.8683	2%
4.085	3.507	0.8585	0.8786	-2%
3.502	3.205	0.9152	0.8879	3%
3.184	2.843	0.8929	0.8950	0%

Figure 20. CoR Data

From Fig. 20, it can be shown that the dynamic CoR is accurate. In four cases the error was above 2%, and in none of the drops did it exceed 5%. This error is due in part, to temporal discretization. Note that the error is the same for both the unmodified model (with e= 0.8), and the new model with e= 0.8. This is particularly evident in comparing the error in rebound heights to error in rebound velocities.

3.2 Implementing an Inertial Force Applicator

The input to the Inertial Force Applicator (IFA) is an acceleration defined via either a spline curve or the coefficients of a quadratic equation. This acceleration is evaluated with respect to the current simulation time and is the resultant force on bodies within the model is the acceleration multiplied by the body mass. There is also the option of scaling the effect by a constant, such as gravity, and the direction of the applied force can also be designated via global unit vector components. Negative numbers are permitted in the vector coordinates, allowing for an applied force in any desired direction.

To demonstrate the use of the IFA, a two-body model was used. This new model consisted of a hammer attached to a pivot at the end of its handle, so it could be proven that the force was applied at the body center of mass.

3.2.1 Setting up the Customized Solver

The IFA is configured using the same procedure for customized solvers as in Section 3.1.2. Both the CoR and IFA use the same DLL file in their computation.

3.2.2 Inserting the User Defined Force Descriptor

The same method as for the User Defined Force Descriptor of Section 3.1.3 is used. The two methods are very similar; however, there are two key differences between the setup for the IFA and the Dynamic CoR. First, the name of the LMSMotionUDF file for the IFA is titled "InitialForce.LMSMotionUDF". Second, whether or not the dynamic CoR has been implemented, the InitialForce.LMSMotionUDF file **MUST** be inserted as type 2, not type 1. There are checks within the code to differentiate between the two types of UDFs, allowing them to work together. If this convention is not followed, the solution may not deliver the expected results or the simulation could fail to completely. Once the UDF Descriptor has been inserted, the second selection of the UDF section in the Forces Toolbar will be colored in.

3.2.3 Defining the Force

Fig. 21 displays the dialog box for the IFA. There are several different available inputs, though not all of them are used at one time. The option "Force Defined by Spline?" is a logical value that will determine whether to evaluate the acceleration using a specified spline curve or by using the quadratic equation:

$$ACCEL = [A]*TIME\text{^}2 + [B]*TIME + [C].$$

The function is evaluated at each integration step, using the current simulation TIME. The quantity ACCEL is multiplied by the mass of each body in the model to obtain the force. If the force is based on a spline curve, then ACCEL is calculated instead by

$$ACCEL = f(TIME),$$

where f(x) is the spline curve.

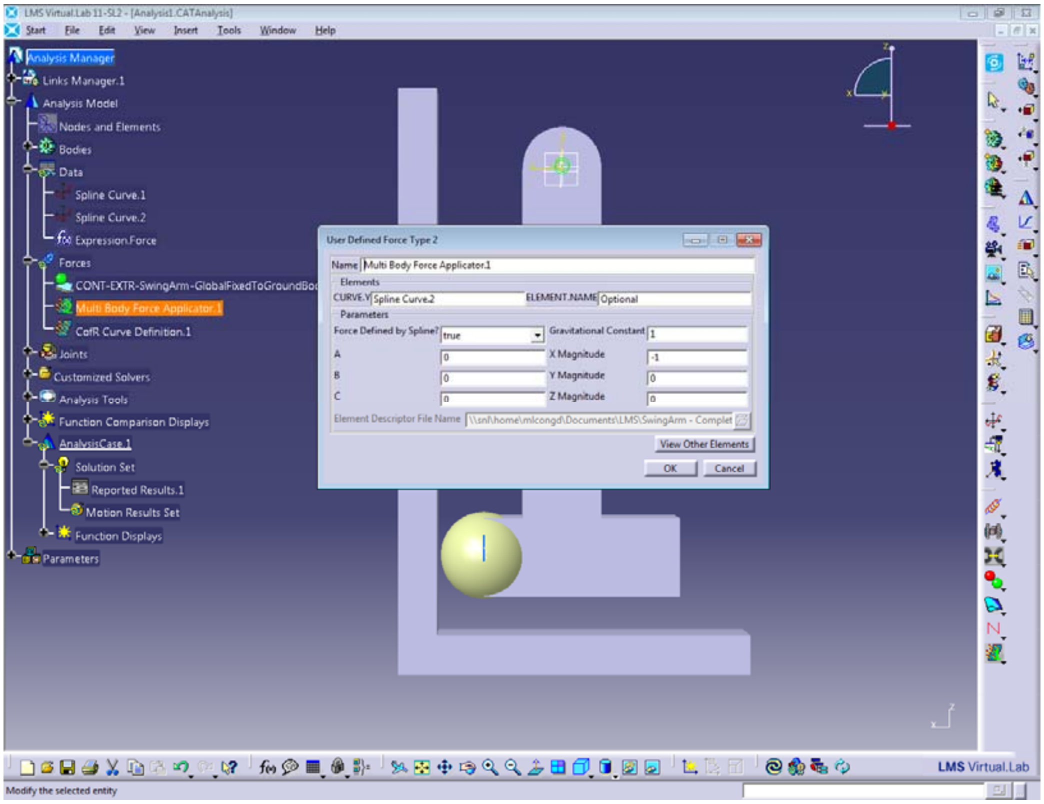

Figure 21. Multi-Body Force Applicator Dialog Box

The X, Y, and Z Magnitudes are the components of a vector that will specify the direction of the force to be applied. Negative numbers are acceptable, and the components are normalized to transform it to a unit vector in the same direction as the original to avoid an additional scaling factor. The "Element.Name" option allows for the specification of a body for which the applied force will be printed to the binary output file if its creation is specified. In Fig. 21, the force is defined by "Spline Curve.2", and the direction lies purely along the negative X-axis (to the right according to the coordinate system in the upper right corner).

3.2.4 Computing the Modified Solution

After setting the Solution Set to use the customized solver (Section 3.1.2), the solution may be computed as normal by either right-clicking on solution set and selecting "Compute", or highlighting the solution set and clicking "Compute" from the toolbar on the right.

3.2.5 Model Analysis and Verification

The acceleration used in this example is a $10\frac{m}{s^2}$ square wave applied for 0.5 s in the negative X direction. At this time the hammer is allowed to freely swing back down and strike the L-shaped bracket (Fig. 22) under the action of gravity (in the global -z direction). The current simulation time can be seen in the "Player" box in the lower left hand corner of Fig. 22, which shows that over the first 0.5 s of the simulation the IFA has swung the hammer back into an almost lateral position.

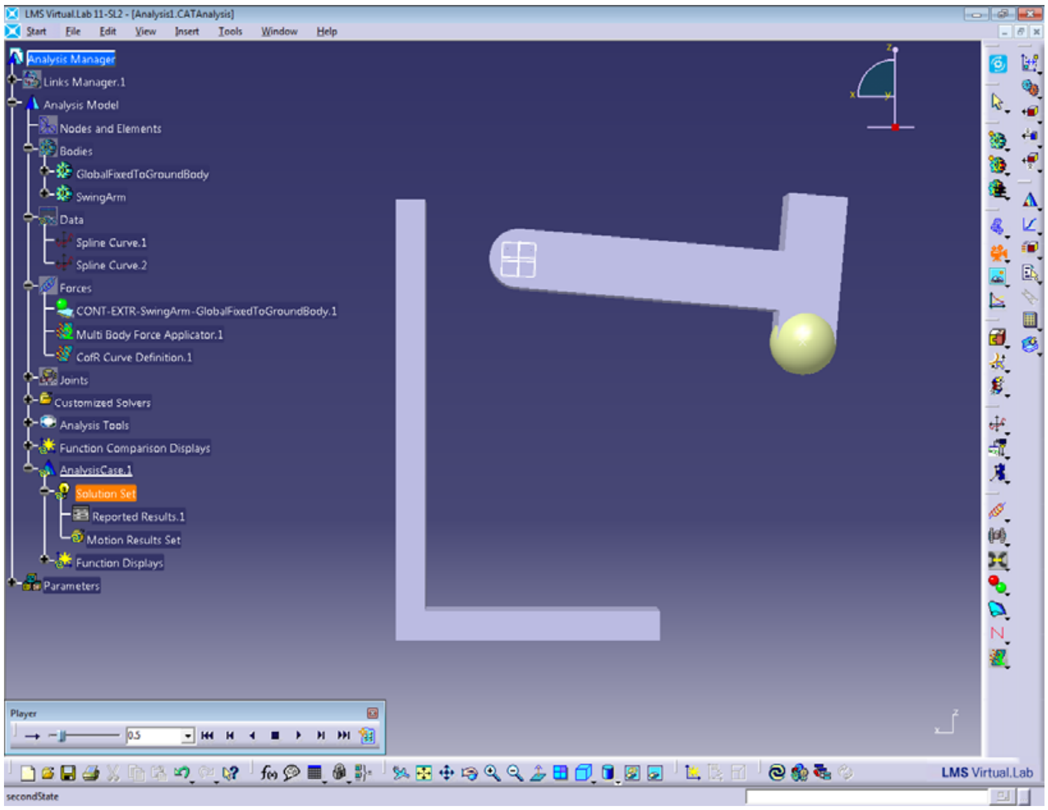

Figure 22. Inertial Force Verification

3.3 Model Independence

It is important to note that the methods discussed above are model-independent. This means that given the DLL file and the LMSMotionUDF files, a dynamic CoR and an initial force can be applied to any model. Also, the CoR may be applied as many times as desired, even reusing the same spline curve multiple times, given that a different contact is selected for each instance of the CoR UDF. The inertial force may also have multiple instances; the applied forces will be summed into the current force acting on each body, allowing for multiple forces acting in multiple directions.

4. CONCLUSION

The improvements to the Rebound Dynamics GUI include an additional impact model, the Multi-Material Impact Modeler. This modeler was designed for use with LMS Virtual.Lab Motion to simulate multiple materials impacting and compute the impact model for each combination. The possible export options have been expanded to now include “Fortran Script” in both the Coefficient of Restitution Modeler and the Multi-Material Impact Modeler. Within the Materials Manager, a “Source” column has been added to allow for traceability of the material properties. Lastly, the model used in predicting the CoR has been updated to more accurately predict CoR across a wide range of velocities.

Future work on this project centers around updating the GUI to accommodate higher order effects, such as strain hardening and oblique, frictional impacts [5].

5. REFERENCES

1. C. P. Volk and M. R. Brake, “A Matlab GUI for the Prediction of Coefficients of Restitution,” 2012, SAND2012-9434, Sandia National Laboratories, Albuquerque, NM.

2. M. R. Brake, "An Analytical Elastic-Perfectly Plastic Contact Model," International Journal of Solids and Structures, 49 (22), pp. 3129-3141, 2012.

3. M. R. Brake, “The Effect of the Contact Model on the Impact-Vibration Response of Continuous and Discrete Systems,” Journal of Sound and Vibration, 332 (15), pp. 3849-3878, 2013.

4. M. R. Brake, “The Role of Epistemic Uncertainty of Contact Models in the Design and Optimization of Mechanical Systems with Aleatoric Uncertainty,” Nonlinear Dynamics, in press.

5. M. R. Brake, “An Analytical Elastic Plastic Contact Model with Strain Hardening and Frictional Effects for Normal and Oblique Impacts,” International Journal of Solids and Structures, submitted.

6. DISTRIBUTION

1 Michael L. Congdon (electronic copy)
University of Illinois at Urbana Champaign
Department of Electrical and Computer Engineering
Champaign, IL. 61820

1	MS0333	Steven A. Fillmore	2616 (electronic copy)
1	MS0333	Matthew D. Williams	2615 (electronic copy)
1	MS0344	Frank J. Dempsey	1526 (electronic copy)
1	MS0346	Jill Blecke	1526 (electronic copy)
1	MS0346	Brenton Elisberg	1526 (electronic copy)
1	MS0346	Kurtis R. Ford	1526 (electronic copy)
1	MS0346	Ryan D. Jamison	1526 (electronic copy)
1	MS0346	Diane E. Peebles	1526 (electronic copy)
1	MS0346	Daniel N. Rogers	1526 (electronic copy)
1	MS0346	Russell D. Teeter	1526 (electronic copy)
1	MS0346	Douglas Vangoethem	1526 (electronic copy)
1	MS0349	Jamey T. Bond	2614 (electronic copy)
1	MS0349	Sergio Gonzalez	2614 (electronic copy)
1	MS0349	Jonathan A. Shelton	2614 (electronic copy)
1	MS0431	Hartono Sumali	0261 (electronic copy)
1	MS0557	Dennis Croessmann	1520 (electronic copy)
1	MS0557	Todd W. Simmermacher	1526 (electronic copy)
1	MS0840	Dannelle S. Aragon	1554 (electronic copy)
1	MS0840	Eliot Fang	1554 (electronic copy)
1	MS0840	John Pott	1555 (electronic copy)
1	MS0840	Daniel J. Rader	1513 (electronic copy)
1	MS0840	James M. Redmond	1550 (electronic copy)
1	MS1064	Georgia Artery	2615 (electronic copy)
1	MS1064	Allen C. Hurst	2615 (electronic copy)
1	MS1064	Laurance L. Lukens	2615 (electronic copy)
1	MS1070	Renee N. Baca	1526 (electronic copy)
1	MS1070	Matthew R. Brake	1526 (electronic copy)
1	MS1070	Jordan Massad	1526 (electronic copy)
1	MS1139	Michael V. Bejarano	1535 (electronic copy)
1	MS1139	Phillip L. Reu	1535 (electronic copy)
1	MS9042	Mike L. Chiesa	8259 (electronic copy)
1	MS0899	Technical Library	9536 (electronic copy)