ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

CO-SIMULATION OF LARGE MBD, FEA AND DEM SYSTEM

José L. Ortiz[1], **and Neil MacDonald**[2]

[1] MSC Software Corporation
201 Depot St. Suite 100, Ann Arbor, MI 48104, USA
jose.ortiz@mscsoftware.com

[2] Neil MacDonald
49 Queen St. Edinburgh EH2 3NH, UK
neil.macdonald@dem-solutions.com

Keywords: Co-simulation, multibody, finite elements, discrete elements.

Abstract. *An extension to an existing co-simulation algorithm [1] is presented. The original algorithm allows co-simulation of a large multibody dynamics (MBD) system with a large finite elements analysis (FEA) model; the present extension consists of a force-displacement co-simulation with a discrete element method (DEM) system.*

The proposed algorithm implements a force-displacement approach where the MBD system steps first using approximations for the forces provided by the DEM model. The DEM model later simulates its model using prescribed displacements provided by the MBD system.

As implemented in [1], there is no fixed communication interval. Each code (MBS, FEA and DEM) are free to take a simulation steps that fits well the current status of the numerical solution. The glue (master) code holds data at every step the co-simulationg codes take.

Implementation details and discussion follow.

1 INTRODUCTION

Co-simulation of large Multibody Systems (MBD) with both Finite Element models (FEA) and Discrete Element models (DEM) is nowadays a requirement from diverse industrial sectors which relay on simulation to speed up the design and development of vehicle, machinery and other devices. The present work is an extension of a recent development [1] where one large MBD system co-simulates with one or more FEA models. The extension consists of allowing the co-simulation with a DEM model.

The present work implements a basic force-displacement algorithm with the additional feature of computing a tangent stiffness matrix from the FEA models. The MBD model runs first with instantaneous forces provided by the FEA model and corresponding stiffness matrix. After the MBD system has advanced in time, the FEA model takes turn and advances in time with prescribed displacements at the points of interaction between the codes. Similarly, the co-simulation with the DEM model is implemented by requesting current forces at the interaction points. The MBD system simulates first using extrapolated values for the forces. Later, the DEM model simulates using prescribed motion at the interaction points. Currently, the DEM model does not support the computation of a tangent stiffness matrix at the interaction points.

2 ALGORITHM SPECIFICATIONS

As described in reference [1], the algorithm is based on a force-displacement interaction between the codes. Technical details of the case of co-simulation between an MBD model and multiple FEA models can be found in [1]. Here only main findings are presented as well as some details of the implementation to support a DEM model.

2.1 Interaction forces acting on the MBD system

The case of co-simulating with a FEA model is handled by applying a force on the MBD system given by the following equation:

$$\begin{Bmatrix} F_i \\ F_j \end{Bmatrix} = \begin{Bmatrix} F_{i0} \\ F_{j0} \end{Bmatrix} - \left[K_{ij} \right] \begin{Bmatrix} U_i - U_{i0} \\ U_j - U_{j0} \end{Bmatrix}. \tag{1}$$

Equation (1) assumes there are two interaction points between the FEA model and the MBD model. The case of arbitrary number of interaction is similar. Forces *Fi* and *Fj* given by Equation (1) are applied on the MBD model to simulate the MBD system first. Here *Fio* and *Fjo* stand for the current value of the interaction forces (provided by the FEA model), *Uio* and *Ujo* stand for the displacements at the starting time step, *Ui* and *Uj* stand for the displacements at the interaction points *i*, and *j* respectively. Finally, *Kij* stands for the tangent stiffness matrix provided by the FEA code.

For the case of co-simulating with a DEM model, the interaction forces have the following expression (assuming two interaction points with the DEM model):

$$\begin{Bmatrix} F_i \\ F_j \end{Bmatrix} = \begin{Bmatrix} F_{ie}(t) \\ F_{je}(t) \end{Bmatrix}. \tag{2}$$

Functions *Fie(t)* and *Fje(t)* are continuous functions built with extrapolated forces obtained using the history of applied forces computed by the DEM model. The DEM model does not compute a stiffness matrix. The implementation allows selecting different extrapolation algorithms.

2.2 Variable communication algorithm

The original algorithm implements a variable communication algorithm (see reference [1]) between the co-simulating codes. The same implementation applies for the case of co-simulating with a DEM model. The basic idea is to allow all codes simulate using their best simulation setting (error settings, particular simulation options, etc.) without imposing a fixed communication interval. The master controller code keeps a history of the forces and displacements of all codes and communicates interpolated/extrapolated values to the co-simulating codes as needed.

2.3 Other features

The implemented algorithm permits all co-simulating codes run in parallel and execute in hosts across the network. Setup is relatively easy through the use of a configuration script. The configuration script is created by the user to define the topology of the co-simulating models and permits setting diverse options to tune the co-simulation. Options allow users to set different Newtonian reference frames for each model and set conversion factors when units are not consistent.

2.4 Limitations

The main limitation of the implemented algorithm is that it does not implement corrective measures when the relative error between computed forces by the MBD code (using either Equation (1) or Equation(2)) at the end of a simulation step, is greater than the actual forces computed by the FEA and DEM co-simulating models.

A second limitation is related to the topology of the co-simulating codes. Only co-simulation between the MBD model and the other models is supported. No FEA-FEA, nor FEA-DEM co-simulation is supported.

A third limitation is that users need to setup the simulation algorithms for each model independently. For instance, a simulation script is need for the MBD code, simulation data needs to be prepared in the FEA models, and simulation data needs to be included in the DEM model independently. Simulation data includes end time of simulation, error settings, etc. The end simulation must be consistent otherwise the co-simulation may be interrupted when a code finishes at an earlier time.

3 NUMERICAL EXAMPLES

The proposed algorithm has been successfully implemented in MSC Adams to co-simulate with MSC Marc and EDEM (flagship software from DEM Solutions). The following numerical examples show both prototype and industrial strength results obtained using the tool.

3.1 Oil pan damage

Figure 1 shows a co-simulation of full MBD model and a nonlinear FEA code. The box on the base of the car models a simple oil pan while the light-blue sphere simulates a stiff spherical rock hit during driving. The co-simulation is limited to low speed impact. The MBD model has no modelling limitations.

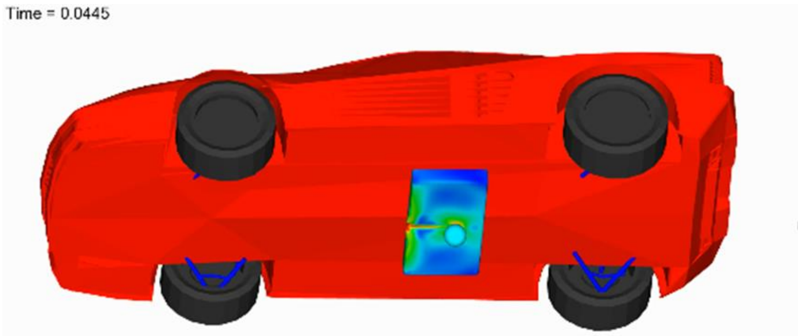

Figure 1: Oil pan damage.

3.2 Moderate misuse scenario

Figure 2 shows a co-simulation of a front suspension and tire MBD model hitting a curb. The control arm in dark blue is modelled in a FEA code and undergoes large nonlinear deformation and plasticity. Forces and displacements have been validated using a different approach showing excellent correlation.

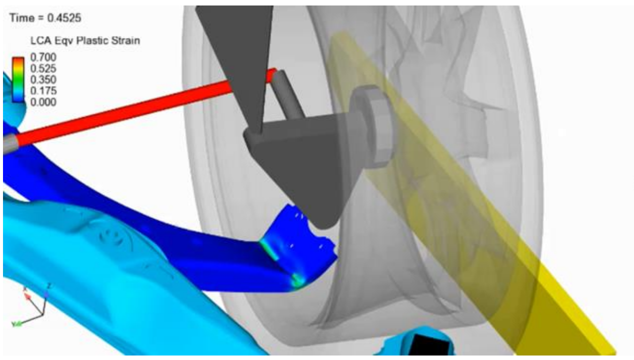

Figure 2: Moderate misuse scenario.

3.3 Torque modulator

Figure 3 shows the results of the simulations using the presented co-simulation algorithm (in red) versus the results using a full FEA model (in light blue). The results shown correspond to a torque modulator forces at a given control point. The full FEA model consists of a set of highly nonlinear coils inside a circular casing. The co-simulation model consists of the just the coils modelled in the FEA code while the casing and gadgets are modelled in the MBD code.

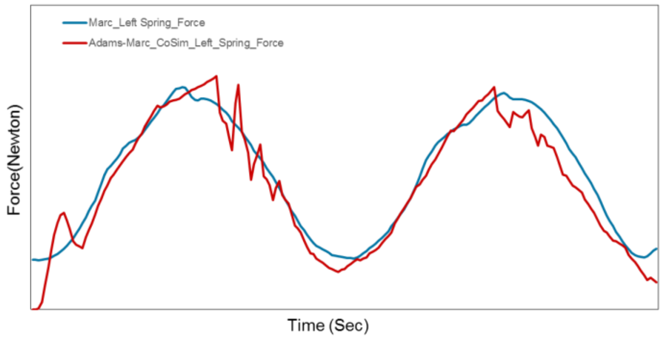

Figure 3: Torque modulator results.

3.4 Vehicle on granular terrain

Figure 4 shows a full ATV MBD model running on a bed of granular material. The MBD model co-simulation consisted of a static analysis to set the model on the ground followed by a dynamic simulation. Figure 5 shows a detail of one tire at time t=1 s.; notice some particles remained attached to the tire threads. The simulation lasted for 10 s and took approximately 20 hours to finish in a standard desktop computer.

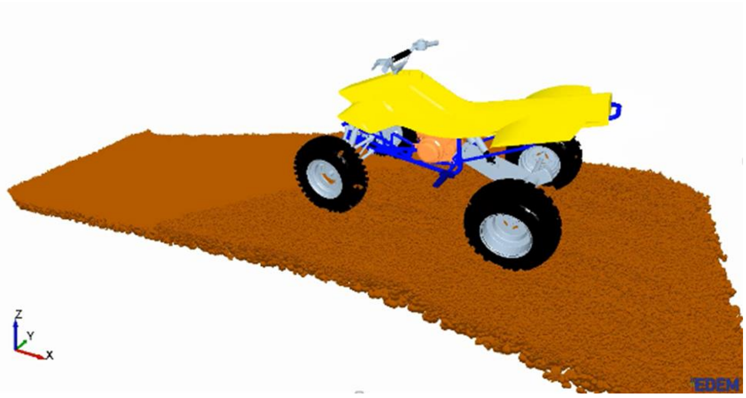

Figure 4: Vehicle on granular terrain.

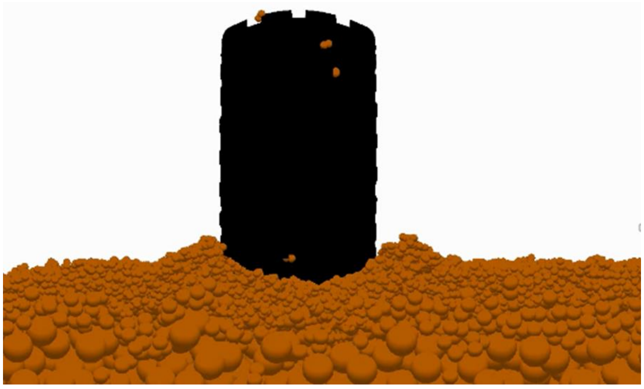

Figure 5: Detail of tire.

4 CONCLUSIONS

- Force-displacement co-simulation algorithms provide useful co-simulation results.
- Variable communication steps among co-simulating codes permits all codes run using their best simulation settings. Co-simulating codes constantly send their interaction data to the controller code.

REFERENCES

[1] I. Ishikawa, M. Tateishi, J. L. Ortiz. Variable communication and stiffness computation in co-simulation of large MBD and FEA systems. Proceedings of ECCOMAS 2015, Barcelona, Spain, July 2015

[2] E. Fahlgren, A. Elliot, M. Carlson. Knee Replacement Simulation Including Large Displacements and Viscoelastic Effects. Proceedings of BioMed 2008. June 18-20, Irvine, CA, USA.

[3] A. Elliot. A Highly Efficient General-Purpose Approach for Co-Simulation with ADAMS. Presented at the 15th European ADAMS Users Conference, November 2000, Rome, Italy.

[4] A. Elliot. Status Update on Advanced General-Purpose Co-Simulation with ADAMS. Presented at the ADAMS 2002 North American Users Conference, May 2002, Scottsdale, Arizona, USA.

[5] The FMI for Model Exchange and Co-simulation Specification v2.0.

[6] S. Sicklinger, V. Belsky, B. Engelmann, H. Elmqvist, H. Olsson, R. Wüchner[1]and, K.-U. Bletzinger. Interface Jacobian-based Co-Simulation. Int. J. Numer. Meth. Engng 2014; 98:418-444.

[7] C. Gear, D. Wells. Multirate linear multistep methods. BIT, 24, 484-502 (1984)