Mar, 2018

PARMES - technical report - 3 - revision - 1

On the non-smooth implicit and smooth explicit multi-body co-simulation coupling in Solfec's hybrid solver

Summary

This report details the design of the hybrid solver in Solfec, coupling its non-smooth implicit multi-body capability with smooth explicit capability of modeling rigid body an spring systems in Parmec. Online examples are avilable.

Contributors

Tomasz Koziara, t.koziara@gmail.com

1 Introduction

This reports describes an implementation of an implicit-explicit co-simulation approach in Solfec [1], where a multi-body domain modeled by means of non-smooth implicit equations is coupled with an explicitly integrated rigid body and spring model. This capability allows to mix geometrically resolved non-smooth multi-body models with simplified models, which may result in saving of computational time, without sacrificing the relevance of the obtained results. The presented functionality has been developed in response to a need of speeding up calculations in a practical application context - modeling of nuclear graphite cores - where two types of models have already been developed:

1. Simplified multi-body models comprising analytical rigid bodies bodies and predefined nonlinear springs (LS-DYNA based).
2. Geometrically resolved multi-body models comprising finite element bodies and (detected during simulation) non-smooth contact constraints (Solfec based).

Algorithm 1 Hybrid co-simulation approach available vie the hybrid solver in Solfec.

In this context, hybrid models - comprised of a small Solfec subdomain, incorporated into a larger simplified model - can be advantageous, delivering relevant results at a fraction of computational cost. Such functionality has been added to Solfec, as the hybrid solver, by combining Solfec with a new code called Parmec, replicating the minimum of LS-DYNA's capability necessary to run unmodified existing input files. In what follows such co-simulation approach is presented in detail.

2 Design

In the hybrid co-simulation approach, also detailed as Algorithms 1, a model in Solfec is coupled with a model in Parmec as follows:

- A set of "boundary" rigid bodies is used, which are present in both models; the "non-boundary" bodies in both models are called "bulk-bodies".
- Parmec time step is equal to the Solfec step divided by an even number $n \geq 2$; this allows to split Solfec step in two sets of $n/2$ Parmec steps, mimicking Solfec's half-step based time stepping approach.
- Boundary bodies computationally belong to and are integrated in Solfec.
- Motion of boundary bodies is gradually applied in Parmec: latest constant velocities from Solfec are used, while their motion is integrated at the Parmec time step.
- Latest resultant body forces and torques from Parmec are applied to boundary bodies in Solfec.
- Contact detection between boundary bodies themselves is disabled by default (it can be enabled via an optional solver argument, and further controlled via the CONTACT_EXCLUDE_{BODIES, SURFACES} commands).
- Contact detection between boundary and bulk bodies is enabled by default (it can be controlled via the CONTACT_EXCLUDE_{BODIES, SURFACES} commands).
- Parmec half-step-run is performed first (comprising $n/2$ Parmec steps), which is followed by a full Solfec step (made of two half-step), which is followed by Parmec full-step-run (comprising n Parmec steps), and so on.
- Parmec runs in a multithreaded manner exploiting all available compute cores (in parallel, it runs on the cluster node hosting the rank 0 MPI process). Solfec can be run both in serial and MPI parallel modes.

3 Implementation

Parmec compiles into parmec4 (single precision) and parmec8 (double precision executables). Along with these, in the Parmec source directory, static library libparmec{4,8}.a and header files parmec{4,8}.h are generated. The double precision version of these is used by Solfec in order to interface with Parmec via the HYBRID_SOLVER interface. The hybrid solver Python call itself is implemented in the lng.c file. The imlpementation is included in hys.{cpp, hpp} and hys.{c, h} files. The C++ hys.cpp and hys.hpp files are the glue layer between the actual C implementation and the libparmec8.{a, h} library interface. The C implementation of the hybrid solver is contained in the hys.c and hys.h files.

4 Examples

4.1 Interaction between a plate and a u-shaped body

This example is can be found in the solfec/examples/hybrid–solver4 directory. The geometrical setup and boundary conditions are depicted in Figure 1. In the hybrid model, both the u-shaped body and the thin plate are modeled in Solfec. The fixed point constraints along the z-edge of the plate are modeled by (maximally damped) stiff springs in Parmec. The amount of leeway travel of the springs, under static conditions and gravity loading, is controlled by the "leeway factor", proportional to the gap between the plate and the "u-way hole", seen on the right hand side of Figure 1. The far-x z-oriented edge of the u-shaped body is fixed in Solfec and its motion along the y-direction is controlled: sine dwell (at 3 Hz and acc. $5m/s^2$) or sine sweep (2 to 8 Hz at acc. $5m/s^2$) are applied. The whole system oscillates back and forth along the y-direction, while exhibiting various degrees of secondary excitation, depending on the leeway factor (and hence the stiffness of the springs realizing the fixed point constraints) and the excitation frequency. A Solfec-only model is also developed where there is no secondary excitation as all constraints are exactly enforced.

In the hybrid model, there are two scales of time stepping – the Solfec model is integrated with a constant time step of 1E-3s – and the Parmec model is integrated with a time step of order 1E-5s or less, depending on the magnitude of leeway. The gap size is 1mm, and hence for a leeway factor of 1.0 such is the allowed free travel of Parmec fixed point constraints. For the leeway factor of 0.1 the allowed free travel is 0.1mm and the Parmec time step is 10-fold smaller, etc.

In Figures 2 and 3 "node 0" is the point on the u-shaped body with the lowest x, y, z coordinates. Figure 2 illustrates time history of node 0 displacement DY for the Solfec-only model and for three hybrid models with leeway factors of 1.0, 0.1, 0.01, under the sine-dwell excitation condition. We can see that the behavior of the hybrid model approximates the Solfec-only model the better, the higher the stiffness of the fixed point constraints is (the lower leeway factor – "lwf"). In the animated response for this example it is clear that for lwf = 1.0 there is a secondary excitation effect, due to interaction of Parmec springs and contacts along the u-way, superimposed with the primary excitation

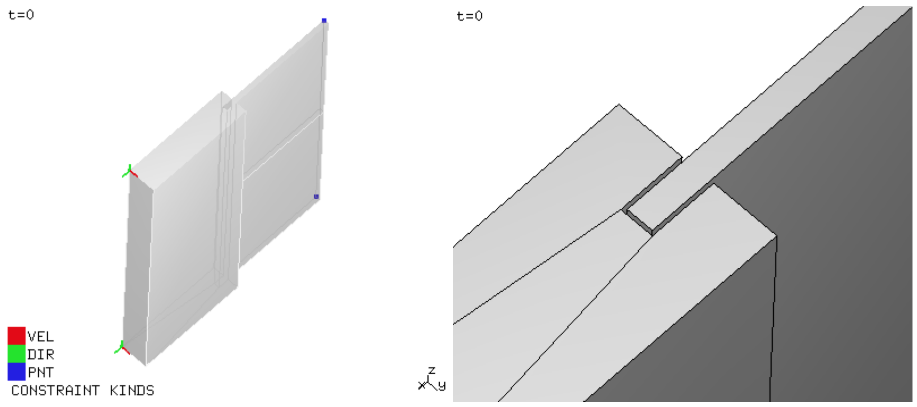

Figure 1: Geometrical setup and boundary conditions of the plate and u-shaped body example.

at the far end of the u-shaped body. Figure 3 illustrates the time history of DY at node 0 for the sine sweep condition. In this case we can see that the Solfec-only and hybrid models do not easily converge. There is a more complex pattern of interactions between the primary and secondary excitation effects, related to the frequency, stiffness and the gap size.

This example illustrates that hybrid models may exhibit behavior in their own right, and are not a straightforward replacement of Solfec-only or Parmec-only models. Care is needed when preparing such models in order to control the possible secondary interaction effects.

4.2 Online examples

Other examples of the application of the hybrid solver can be found online. These include:

- A two–body impact problem
- 1,2,3–dimensional cube array acceleration dwell
- 3–dimensional cube array parallel scaling

5 Conclusions

Hybrid modeling extends the capability of Solfec by allowing coupling of bulk model areas comprising rigid or deformable bodies with non-sooth, implicitly resolved contact constraints, and explicit rigid body and (nonlinear) spring models set up and run as a co-simulation, using Parmec. This approach may help speed up calculations for larger models where only limited areas may need to be resolved in a geometrically and kinematically detailed manner, and include interactions by means of non-smooth contact constraints.

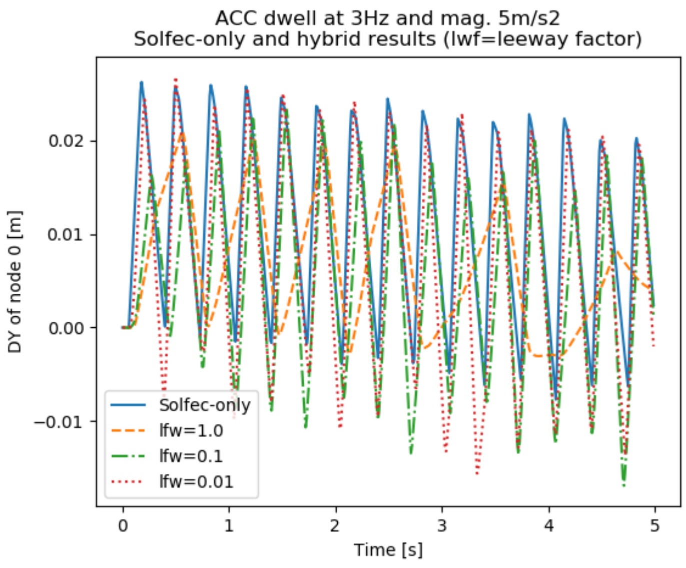

Figure 2: Displacement DY of node 0 (lowest x, y, z for the u-shaped body) for the sine dwell excitation.

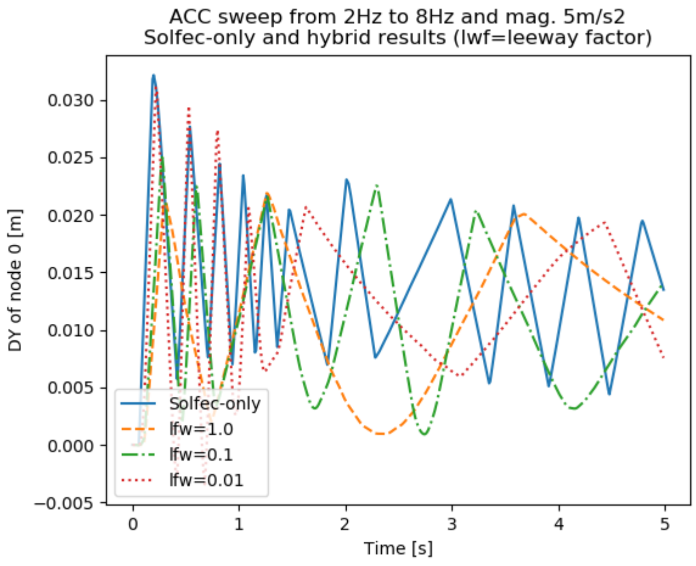

Figure 3: Displacement DY of node 0 (lowest x, y, z for the U-shaped body) for the sine sweep excitation.

Acknowledgements

The support from EDF Energy is gratefully acknowledged.

Revision notes

Rev. 1 Initial revision;

References

[1] Tomasz Koziara and Nenad Bićanić. A distributed memory parallel multibody contact dynamics code. *International Journal for Numerical Methods in Engineering*, 87(1-5):437–456, 2011.