2018 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
Modeling & Simulation, Testing and Validation (MSTV) Technical Session
August 7-9, 2018 – Novi, Michigan

HIGH PERFORMANCE COMPUTING FRAMEWORK FOR CO-SIMULATION OF VEHICLE–TERRAIN INTERACTION

Radu Serban
Dept. of Mechanical Engineering
University of Wisconsin - Madison
Madison, WI

Nicholas Olsen
Dept. of Mechanical Engineering
University of Wisconsin - Madison
Madison, WI

Dan Negrut
Dept. of Mechanical Engineering
University of Wisconsin - Madison
Madison, WI

ABSTRACT

Current modeling and simulation capabilities permit tackling complex multi-physics problems, such as those encountered in ground vehicle mobility studies, using high-fidelity physics-based models for all involved subsystems, including the vehicle, tires, and deformable terrain. However, these come at significant computational burden; research and development on new software architecture and parallelization techniques is crucial in enabling such predictive simulation capabilities to be useful in design of new vehicles or in operational settings.

In this paper, we describe the architecture, philosophy, and implementation of a distributed message-passing-based granular terrain simulation capability and its incorporation into an explicit force–displacement co-simulation framework to enable effective simulation of multi-physics mobility problems. We demonstrate that the proposed infrastructure has good parallel scaling characteristics and can thus effectively leverage available computing resources. Furthermore, we show that the outer communication layer, also implemented with a message passing approach, is effective and adds negligible overhead.

1 INTRODUCTION

Multibody system dynamics models and tools for vehicle simulation, both commercial and open-source, are nowadays at very high levels of maturity and sophistication and are routinely used, both in research and industry, for design and analysis of ground vehicles, particularly for on-road scenarios. However, simulations of off-road vehicles continue to rely heavily on empirical or semi-empirical methods that, while computationally expeditious, lack the level of fidelity necessary to capture all effects in complex vehicle – terrain interactions. Relatively recent advances in terramechanics modeling and associated simulation software are motivating a push towards physics-based, high-fidelity modeling and simulation of the overall off-road mobility problem, including the vehicle, tires, and soil. An example of efforts in this direction is the ongoing work on producing a next generation of the NATO Reference Mobility Model [1] using a completely physics-based solution, including full 3-D vehicle multibody models and high-fidelity soil and terrain representations [2].

While solutions coupling high-fidelity vehicle models with fully-resolved granular dynamics terrain models have been demonstrated to capture the effects important for predictive mobility studies [3, 4, 5], they are computationally very intensive; this is especially the case for simulations coupling Finite Element Analysis (FEA) for tire modeling with Discrete Element Method (DEM) for granular soil representation. Monolithic simulations are prohibitively expensive and undesirably limit freedom in selecting appropriate numerical methods and/or parallelization technique due to the inherent multi-physics nature of the problem. Furthermore, solutions such as using a moving patch approach to decrease the computational cost of deformable soil simulations are not always appropriate (for example, when interested in multi-pass effects or simulations of vehicle convoys).

Limitations imposed by monolithic solutions can be alleviated by adopting a co-simulation approach to the multi-physics mobility problem, allowing independent and concurrent simulation of subsystems naturally separated by their physical characteristics (vehicle – tire – terrain). Such techniques have been demonstrated to be suitable for effectively calculate mobility metrics such as drawbar pull and motion resistance, investigate deformable tire behavior (strain and stress distributions), and quantify multi-pass effects on terrain compaction and vehicle performance [3]. As part of previous work, we introduced a software framework which implements an explicit force–displacement co-simulation technique to decouple the global problem involving a vehicle multibody system, flexible FEA-based tires, and deformable terrain modeled using DEM techniques [4]. That solution relies on multi-core parallelization of the granular terrain simulation, implemented through OpenMP [6]. While it was demonstrated that this permits effectively advancing the state of all involved subsystems independently and in parallel, thus reducing the overall time to solution to the cost of the most expensive subsystem, it was also noted that scenarios involving large-scale terrain patches quickly lead to the granular dynamics simulation becoming the computational bottleneck. This is due to the fact that multi-core parallelization does not scale well beyond a relatively small number of threads owing to cache coherence and false-sharing issues [7]. Other single-node alternatives, such as using GPU acceleration for the granular dynamics simulation are still limited in scalability, primarily due to the available device memory, computational cost of out-of-core implementations, or complexity of multi-GPU approaches.

The work presented herein was therefore motivated by the desire to provide a fully scalable solution for granular terrain simulations (limited only by available computing resources) while maintaining the advantages provided by a co-simulation approach to the vehicle–tire–terrain global problem. Both strong and weak scalability of the proposed solution are important and both are considered herein. In the context of a co-simulation framework, the final goal in establishing a scalable granular dynamics capability is ensuring that, in all scenarios of interest, the computational bottleneck is always the simulation of one flexible FEA-based tire. While this calculation can be (and usually is) expensive itself, this is a fixed computational cost. Furthermore, the subsystem separation provided by the co-simulation approach allows work on improving efficiency of FEA calculations to occur as a separate task; this is the object of ongoing related work.

The outcome is a hybrid platform which implements parallelism at multiple levels, including a distributed granular dynamics solver based on a domain-decomposition approach and leveraging multi-core parallel computing on each subdomain. This co-simulation framework relies on Message Passing Interface (MPI) [8] for the data exchange in the outer co-simulation layer and on a hybrid MPI-OpenMP parallelization for highly-scalable terrain simulation.

This paper is organized as follows. In the next section we describe the overall approach and the architecture of the co-simulation framework and provide a brief overview of the vehicle and flexible tire modeling and simulation capabilities available in Chrono. Section 3 discusses the hybrid distributed/multi-core parallel solution technique for large-scale granular dynamics problems, based on distributed domain-decomposition coupled with multi-core parallelization of each local subproblem. We provide results from numerical experiments in Section 4 where the emphasis is on computational and parallel efficiency. We conclude the paper in Section 5 where we also discuss related ongoing and planned research and development efforts.

2 TECHNICAL APPROACH

The co-simulation framework described here was implemented within the Chrono multi-physics package [9, 10]. Developed through a joint university effort, Chrono is available under a BSD-3 license [11] and is distributed as a suite of middleware libraries organized in separate functional modules. The core module provides support for the modeling, simulation, and visualization of rigid multibody systems; optional modules provide support for additional classes of problems (e.g., finite element analysis and fluid-solid interaction), for modeling and simulation of specialized systems (such as ground vehicles and granular dynamics problems), provide specialized parallel computing algorithms for large-scale simulations (multi-core, GPU, and distributed), or offer additional utilities for post-processing and run-time visualization.

As an open-source, freely available software package, Chrono fosters collaborative development and offers a convenient framework for exploring and developing new simulation and architecture paradigms, such as the one discussed in this paper. Furthermore, its permissive licensing model enables cost-effective design of experiment analyses allowing parallel high throughput computing simulations, relevant in applications such as off-road vehicle mobility assessment.

We provide next an overview of the hybrid MPI-MPI-OpenMP co-simulation framework proposed as a solution to simulating complex vehicle–terrain interaction in which the terrain is represented through a Discrete Element Method (DEM) on domains of practically arbitrary dimensions and with arbitrary levels of resolution of the granular material (of course, modulated by available computing resources and numerical limitations inherent to simulating particles at very small scale). We then briefly list the Chrono modules that enable simulation of the various subsystems considered here, in particular its ground vehicle modeling capabilities through the Chrono::Vehicle module and the Chrono::FEA finite element analysis (FEA) module used to model and simulate flexible tires. We concentrate most of the discussion on efficient parallelization of granular dynamics simulations, focusing in particular on the design, architecture, and implementation of Chrono::Distributed, a new module targeted at distributed simulation of many-body dynamics with friction and contact.

2.1 High-level framework overview

The proposed co-simulation framework (see Fig. 1) relies on two MPI layers, one for implementing the co-simulation data exchange and an inner layer for the domain-decomposition-based distributed granular terrain simulation, and a multi-core parallel computing layer used to accelerate FEA calculations for the tire subsystems as well as the local granular dynamics problems in each terrain subdomain.

The top level of the co-simulation framework is a three-way communication mechanism between the vehicle subsystem, the four (or more) tire subsystems, and the terrain. This layer implements an explicit force–displacement co-simulation scheme, wherein the vehicle exchanges wheel state and tire forces with each individual tire subsystem and each tire exchanges (deformable or rigid) mesh node states and contact forces with the terrain subsystem. The current implementation relies on MPI (see Fig. 1) but could be implemented with a different mechanism.

Decoupling the simulation of the inherently multi-physics off-road mobility problem into vehicle, tire, and deformable terrain subsystems allows greater flexibility in accelerating the simulation. First, each subsystem can evolve its internal dynamics with a suitable integration time step, which can be different between the various subsystems and dictated by accuracy and stability consideration for that subsystem. Second, each subsystem can employ a different numerical integration scheme, suitable for its specific dynamics problem; for instance, we use an adaptive HHT (Hilber-Hughes-Taylor) scheme in Chrono::FEA while using a semi-implicit Euler scheme for granular dynamics problems. Finally, and most importantly in the context of this work, a co-simulation approach permits leveraging different and independent parallelization techniques for each subsystem, as dictated by the structure, type, and characteristic of each subproblem.

Computation of internal forces and Jacobians in Chrono::FEA, particularly compute-intensive when using the Absolute Nodal Coordinate Formulation (ANCF), leverages shared memory multi-core parallelism through OpenMP. Granular terrain simulation can be performed either in a shared memory, OpenMP-based parallel setting, or else in a distributed fashion using the Chrono::Distributed hybrid parallel model. As described below in Section 3 and demonstrated in Section 4.1, the latter offers greatest scalability and parallel performance.

2.2 Vehicle and tire simulation

Chrono::Vehicle [12] is a specialized Chrono module which provides a collection of templates (parameterized models) for various topologies of both wheeled and tracked vehicle subsystems, as well as support for closed-loop and interactive driver models, and run-time and off-line visualization of simulation results. Chrono::Vehicle leverages and works in tandem with other Chrono modules, such as Chrono::FEA for finite element support; Chrono::Granular for granular dynamics support; Chrono::Irrlicht and Chrono::OpenGL for run-time visualization; and Chrono::Parallel and Chrono::Distributed for parallel computing support.

Chrono::Vehicle provides a comprehensive set of vehicle subsystem templates (for tires, suspensions, steering mechanisms, drivelines, sprockets, track shoes, etc.), templates for external systems (for powertrains, drivers, terrain models), and additional utility classes and functions for vehicle visualization, monitoring, and collection of simulation results. As a C++ middleware library, Chrono::Vehicle requires the user to provide classes for a concrete instantiation of a particular template. An optional Chrono library provides complete sets of such concrete C++ classes for a few ground vehicles, both wheeled and tracked, which can serve as examples for other specific vehicle models. An alternative mechanism for defining concrete instantiation of vehicle system and subsystem templates is based on input specification

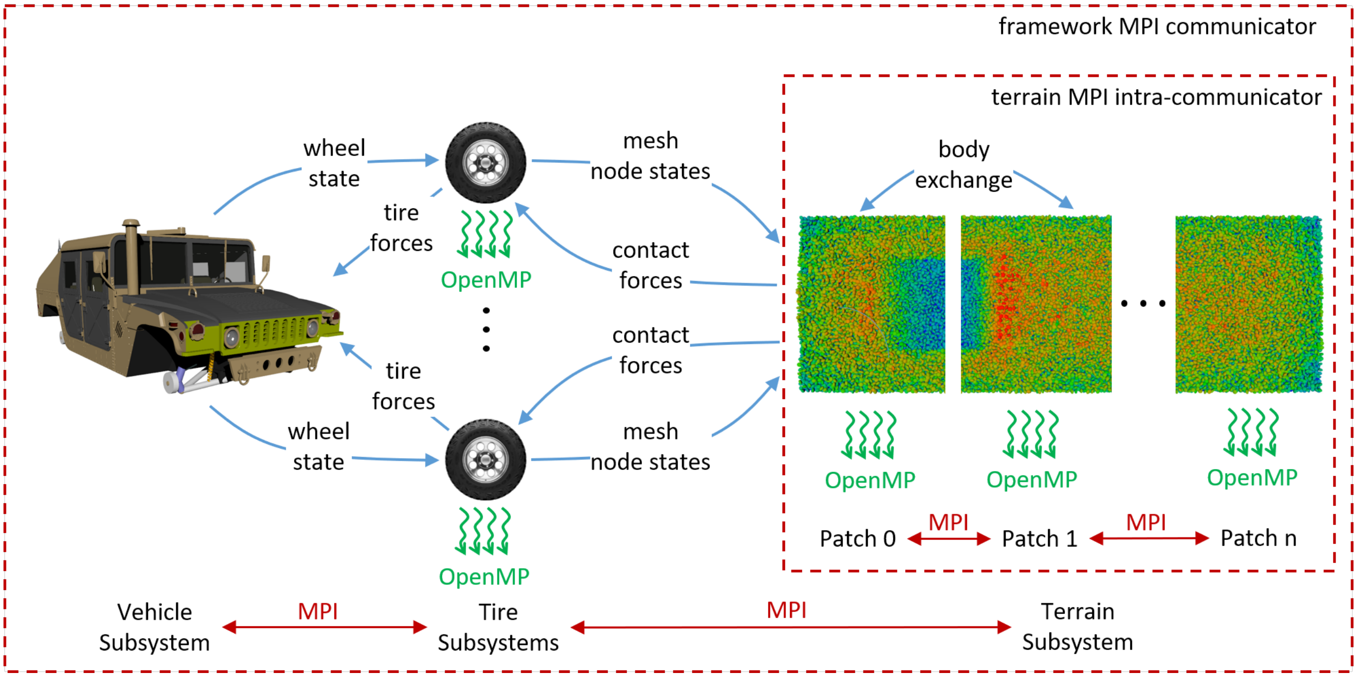


Figure 1: Schematic of the hybrid MPI-MPI-OpenMP co-simulation framework. The outer MPI layer is responsible for orchestrating the inter-system co-simulation communication. The inner MPI-OpenMP layer is responsible for the hybrid parallel granular terrain simulation.

files in the JSON format.

The particular off-road vehicle used for the numerical experiments in Section 4 was a four-wheel drive off-road vehicle with independent double-wishbone suspension and Pitman arm steering (see Fig. 2).

Chrono::Vehicle currently supports three different categories of tire models: rigid, semi-empirical, and finite element. Rigid tires can be modeled as cylindrical shapes or else as non-deformable triangular meshes. From the second class of tire models, Chrono::Vehicle provides template implementations for Pacejka (89 and 2002), Fiala, Lugre, and TMeasy tire models, all suitable for maneuvers on rigid terrain. Finally, the third class of tire models offered are full finite element representations of the tire, using either ANCF or Reissner shell elements. The ANCF-based flexible tire model in Chrono::Vehicle, used in this study, is based on the laminated ANCF shell el-

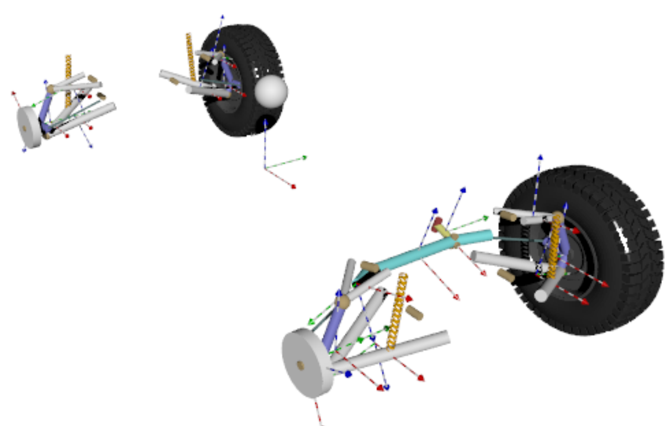


Figure 2: Wheeled vehicle with double wishbone suspensions and Pitman arm steering.

ement [13, 14]. Additional details on the actual Chrono implementation of this element can be found in [15].

2.3 Granular terrain simulation

While providing a full-fledged multibody simulation framework, a unique characteristic of Chrono lies with its mature support for frictional contact. For such problems, Chrono implements both non-smooth (NSC) and smooth (SMC) contact approaches. The former relies on a complementarity formulation which, in conjunction with a Coulomb friction law, results in a Differential Variational Inequality (DVI) form of the equations of motion. The smooth contact method (also known as a penalty method) can be viewed as regularization approach wherein normal and tangential contact forces are based on local body deformation at the contact point. NSC and SMC also differ in terms of their modeling capabilities, parameterizations, as well as in their computational complexity and amenability to parallel computing. The two approaches to frictional contact implemented in Chrono have been compared and validated in [16]. For further details on NSC, the reader is directed to [17, 18]. The SMC formulation is discussed in [19, 20].

Large-scale simulations of granular material (i.e., problems with millions of bodies interacting through contact and friction), where each particle is considered individually and all inter-particle interactions accounted for, are collectively known under the name Discrete Element Method (DEM). Because NSC formulations for frictional contact are relatively more recent, the term DEM is usually associated with SMC-type formulations. To distinguish between the two, here we use the acronyms DEM-C (for complementarity) and DEM-P (for penalty).

As mentioned above, a distinguishing characteristic of the two formulations (NSC and SMC) and therefore of the two corresponding DEM flavors is the type of resulting equations of motion. While DEM-C requires the solution of a global optimization problem (obtained after manipulations of the DVI equations of motion), inclusion of frictional contact forces in DEM-P is a local process, on a per-contact basis. As such, DEM-P is much easier to parallelize, particularly in a distributed computing environment.

3 PARALLEL GRANULAR DYNAMICS

In order to enable large-scale, high-resolution granular dynamics simulations, we developed the Chrono::Distributed module of Chrono. Chrono::Distributed wraps existing OpenMP support in Chrono::Parallel with MPI support for a *hybrid parallel* solution to large-scale computational granular dynamics. The intent of Chrono::Distributed is to break a large granular simulation into many smaller, mostly-disjoint Chrono::Parallel simulations which run on powerful many-core nodes with fast interconnect. On this type of architecture, Chrono uses parallelism at three levels to utilize a large number of CPU cores and memory and accelerate simulation: MPI for domain-decomposition, OpenMP for shared memory parallelization of the local problem on each processor, and SIMD instruction-level parallelism within each local thread.

3.1 Multi-core granular dynamics

Chrono::Parallel, an optional module of Chrono, provides multi-core shared-memory support for granular dynamics using OpenMP directives and the Thrust library with the OpenMP back-end.

Chrono::Parallel is constructed so that it relies on the core Chrono functionality for its modeling capabilities, while using different data structure and algorithms more suited to OpenMP shared-memory parallelism. Most importantly, Chrono::Parallel uses a single data container which implements a *Structure of Arrays* (SoA) scheme. This contrasts with Chrono's main *Array of Structures* (AoS) representation of data. The SoA form is much more amenable to loop parallelism and to SIMD capabilities on modern CPUs. Dependencies of Chrono::Parallel include OpenMP [6] (for parallel for loops), Thrust [21] (for parallel sort, scan, and reduction algorithms), and Blaze [22] (for high-performance sparse matrix representation).

Chrono::Parallel gives the user access to a large set of the modeling elements available in Chrono; in particular, these include rigid bodies with frictional contact (both penalty- and complementarity-based), arbitrary collision shapes, kinematic joints, force elements (springs, actuators, etc.), as well as 1-D shaft elements required for simulation of complete Chrono::Vehicle models of both wheeled and tracked vehicle systems. Chrono::Parallel also provides implementations of most DVI solvers in Chrono (including Barzilai-Borwein and Nesterov-type solvers), but the module is currently limited to a semi-implicit Euler time stepping scheme. Because of this restriction, finite-element models cannot be currently tackled with Chrono::Parallel directly.

Another module, Chrono::Granular, provides a number of utility functions for specification of granular materials with either preset or user-defined distributions of material attributes. These include specifications of particle size and shape, physical properties, and contact material characteristics. Chrono::Granular also provides efficient spatial sampling for initializing granular material in prismatic, cylindrical, and spherical domains according to regular grids, hexagonal close-packing, and Poisson disk sampling (which provides uniformly distributed initial positions with guaranteed minimum separation).

Parallel collision detection. Both DEM-P and DEM-C approaches require a mechanism for collision detection, a process responsible for producing a geometric characterization of interactions of colliding shape pairs at any given system configuration. This characterization includes the pair of closest points, normal, radii of curvature at the contact point, etc. The most basic approach to collision detection naively tests all pairs of contact shapes in the system, which gives it $\mathcal{O}(n^2)$ complexity and therefore renders it impractical for large-scale problems. In order to improve upon the complexity of the naive algorithm, most collision detection systems use (at least) a two-phase process. The first phase, called *broad-phase*, quickly dismisses pairs of shapes that are clearly not interacting. In order to do this efficiently, broad-phase uses bounding shapes (spheres, axis-aligned boxes, or object-oriented boxes) and specialized data structures like dynamic trees or hierarchical grids to spatially organize collision shapes; this reduces the set of possible interactions to only include pairs of shapes near to each other. In a second *narrow-phase*, the pairs of shapes identified during broad-phase are closely examined geometrically, either using analytical methods for shapes with simple geometry, or else with algorithms like Gilbert-Johnson-Keerthi (GJK) or Minkovski-Portal-Refinement (MPR) which are able to determine the contact characterization for pairs of general convex shapes. Depending on their representation, concave collision shapes may require a convex decomposition pre-processing step.

The broad-phase in Chrono::Parallel implements a binning algorithm (see [23] for implementation details), suitable for both multi-core and GPU shared-memory parallel environments, and analytical algorithms with a fall-back on MPR for the narrow-phase collision detection.

3.2 Distributed granular dynamics

Chrono::Distributed is a relatively minimal module which provides synchronization functionality in order to coordinate Chrono::Parallel simulations, which each run on their own MPI rank. In the distributed framework, the predefined spatial simulation domain is broken into subdomains along an axis and each subdomain is assigned to a rank. Each rank is then only responsible for computations on the bodies in its designated subdomain and in a thin border layer with its neighbor subdomains. This boundary layer extends slightly into both subdomains and is populated with bodies that are seen on both MPI ranks. In order to maintain consistency between neighbor subdomains Chrono::Distributed stores a status for each body on each rank to denote whether information for that body must be shared with neighboring ranks. These statuses are induced by a spatial breakdown of each subdomain as show in Fig. 3. Each subdomain consists of three types of regions: owned, shared, and ghost. Bodies in the owned region are seen only by that rank and no synchronization is necessary for them. Shared bodies are those bodies still in the rank's subdomain, but that are close enough to affect bodies in the neighboring subdomain. They are advanced in time by this rank, but are represented as a proxy body on the neighboring rank. Finally, bodies in the ghost region lie outside of the subdomain, but are close enough to affect bodies in the subdomain. They are advanced in time by the neighboring rank, but exist as proxies in order to couple the neighboring subsystems. Note that shared bodies on one rank are represented by proxy bodies on another, and vice versa.

Communication between ranks is done at the end of each time-step, at which point all bodies on each rank are classified into the aforementioned statuses based on their location. A series of MPI messages between ranks then (*i*) creates new proxy bodies for bodies which have moved into a border region, (*ii*) deletes proxy bodies for bodies which have moved out of a border region, and (*iii*) updates proxy bodies from the appropriate rank in order to maintain continuity between all ranks. MPI communication consists of packing buffers of MPI custom data types and sending them point-to-point between neighboring pairs of ranks. In the framework, it is important to note that communication is never global, and that each rank need only communicate with a maximum of two neighbors. This prevents major scaling penalties associated with collective MPI transactions in large communicators.

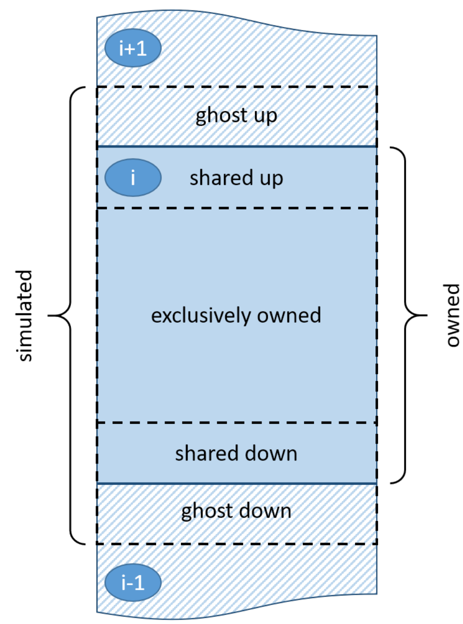


Figure 3: Regions of an internal subdomain i in Chrono::Distributed. In addition to bodies in its *owned* region, subdomain i also integrates the states of bodies in *ghost layers* of adjacent subdomains.

Chrono::Distributed does not require a specialized distributed collision detection algorithm. Instead, in line with its design philosophy, it provides enough state information about the system through appropriate replication of shared bodies with proxies on neighboring MPI ranks to simply use the existing Chrono::Parallel shared memory parallel collision detection described above, in parallel and independently on each processor.

4 NUMERICAL EXPERIMENTS

The primary focus of the simulations and numerical experiments presented herein was on scalability and parallel efficiency of overall solution and distributed terrain simulation within the co-simulation framework. We do not address the issue of performance of the Chrono::FEA module (which, as shown below can and does become the computational bottleneck); while a very important aspect in itself and critical to the effectiveness of the proposed approach, this falls outside the scope of this paper but is the object of ongoing work. Similarly, we do not concentrate on using the proposed framework for conducting actual mobility studies for off-road vehicles on deformable terrain. The ability of capturing relevant mobility metrics with a simulation capable of complex, high-fidelity representation of a vehicle model, flexible tires, and deformable granular soil has already been demonstrated in [3, 4].

All numerical experiments reported here were conducted on a Cray XC30 system. The compute nodes are equipped with 12-core Intel® Xeon® E5-2697 v2 processors and are interconnected with a dedicated Cray Aries high-speed network.

4.1 Scalability properties

We assessed the scalability characteristics of Chrono::Distributed, first as a stand-alone granular dynamics solver and second within the co-simulation framework. We present results and analyze both strong and weak scalability. The former is concerned with the change in solution time as the number of computing resources is increased on a *problem of fixed total size*; in the context of off-road vehicle mobility, good strong scaling properties are relevant in accelerating solution of a given maneuver on a fixed terrain domain with fixed soil characteristics when resource allocation is increased. Weak scaling is defined as the variation in solution time with number of processors for a *fixed problem size per processor*; good weak scaling properties translate in the ability to increase resolution of terrain representation proportional to compute resource allocation, with minimal effect on time to solution.

Denoting with $T(n)$ the time to solution (wall-clock time) for a simulation conducted on n processors, appropriate performance metrics for *parallel efficiency* can be defined as:

$$E_s(n) = \frac{T(1)}{nT(n)} \qquad \text{for strong scaling,} \tag{1}$$

$$E_w(n) = \frac{T(1)}{T(n)} \qquad \text{for weak scaling.} \tag{2}$$

Perfect scaling corresponds to $E(n) = 1$ for all n.

Taking into account the current limitations of Chrono::Distributed, most notably the lack of dynamic load balancing, a scalability analysis was performed on a settling simulation of layered granular material. This particular test was also selected because it is representative of the use of Chrono::distributed for simulation of granular terrain for mobility studies. Strong and weak scaling performance of Chrono::Distributed is summarized in Tables 1 and 2, respectively. These simulations correspond to 2 s of settling for the corresponding number of particles in a domain of width 0.5 m and a base length of 4 m. For the weak scaling experiments, the computational domain length was increased proportionally with the number of processors (MPI ranks), up to 128 m when using 32 processors. In all cases, the granular material consisted of identical spheres of radius 1.25 mm, initialized in layers with uniformly distributed positions in each layer in a rectangular grid.

The main reason for departures from a perfect parallel efficiency value $E(n) = 1$ in domain-decomposition architectures such as the one implemented in Chrono::Parallel is the increased cost of inter-process communication (more interface boundary in the strong scaling experiments or more data per message in the case of weak scal-

MPI Ranks	Particles	Ratio	Time (s)	$E(n)$
1	1,236,372	1	23,489.5	-
2	1,236,372	1	11,826.7	0.993
4	1,236,372	1	5,954.3	0.986
8	1,236,372	1	2,772.5	1.059
16	1,236,372	1	1,439.5	1.020
32	1,236,372	1	712.3	1.031

Table 1: Strong scaling results on settling problem. The parallel efficiency is defined as: $E_s(n) = \frac{T(1)}{nT(n)}$. Wall-clock times represent the time to simulate the process for 1 s.

MPI Ranks	Particles	Ratio	Time (s)	$E(n)$
1	1,236,372	1	23,489.5	-
2	2,472,744	2	23,901.0	0.983
4	4,944,700	4	23,997.6	0.979
8	9,889,400	8	24,099.4	0.975
16	19,778,012	16	24,406.5	0.962
32	39,555,236	32	24,480.8	0.960

Table 2: Weak scaling results on settling problem. Here, parallel efficiency is defined as: $E_w(n) = \frac{T(1)}{T(n)}$. Wall-clock times represent the time to simulate the process for 1 s.

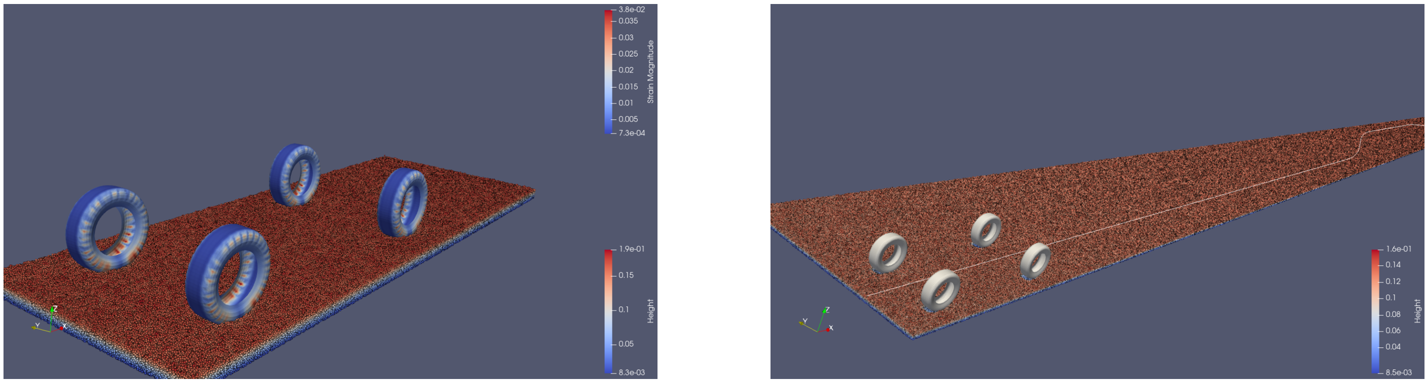
ing). As the results in Tables 1 and 2 indicate, the implementation of the communication layer in Chrono::Distributed, most notably the judicious use of asynchronous (immediate) point-to-point MPI communication, results in minimal communication overhead and thus translates in parallel efficiencies close to 1. It is worth noting that, in some instances, strong scaling yields efficiencies above 1.0. The reasons for this are (i) the slightly nonlinear scaling of the broad-phase collision detection phase in Chrono::Parallel, which performs better on small problem sizes, and (ii) likely better cache utilization on smaller local problem sizes.

4.2 Full vehicle on deformable terrain

To demonstrate the capabilities of the proposed framework and the effectiveness of a distributed simulation of the underlying granular terrain subsystem, we chose to simulate the following two scenarios:

- Acceleration test of a vehicle with flexible tires on granular terrain (Fig. 4a). The goal of this example is to demonstrate that, beyond a certain point, the resolution of the granular terrain representation does not affect overall performance (given enough computing resources). This is due to the fact that the time evolution of the granular terrain subsystem occurs at a fraction of the cost of advancing the dynamics of a single flexible FEA-based tire subsystem.

- Double lane change (DLC) maneuver of a vehicle with rigid-mesh tires on granular terrain (Fig. 4b). The purpose of this example is to eliminate the limiting effect of FEA-based tires on overall time to solution and demonstrate the strong scalability characteristics of the distributed terrain simulation within the context of the proposed co-simulation framework.



(a) Acceleration test; ANCF-based flexible tires.

(b) Double-lane change; rigid-mesh tires.

Figure 4: Simulation snapshots for off-road vehicle on granular terrain.

Parameters of the two types of simulations above are summarized in Table 3, including number of particles and particle radius. In all cases, the soil cohesion is set at 100 kPa. The vehicle has an overall mass of 2550 kg, independent front and rear suspension and a Pitman arm steering mechanism. In simulations **A** and **B**, each FEA-based tire is slick and is modeled with a 90×24 mesh of multi-layered, orthotropic ANCF shell elements [24], with the section geometry created by cubic splines. The default tire internal pressure is assumed to be 200 kPa. Rigid-mesh tires in experiments **C** and **D** are modeled with a $2 \times 90 \times 24$ fixed triangular mesh. In the acceleration tests, the vehicle throttle input is ramped to 80% over a duration of 0.2 s and kept constant afterwards; steering and braking inputs are maintained at 0%. The DLC simulations rely on a driver controller which adjusts steering input to follow the path depicted in Fig. 4b while throttle and braking inputs are adjusted to maintain a constant speed of 10 m/s.

The good parallel scaling characteristics of Chrono::Distributed (see Section 4.1) are maintained when integrating this module within the co-simulation framework for the simulation of the terrain subsystem. This is illustrated by the timing results presented in Table 3. These simulations presented in this section were set up as to permit a scaling assessment of the underlying distributed terrain simulation and an evaluation of the overall co-simulation framework performance when increasing computing resources. Indeed, the acceleration tests using flexible tires (columns **A** and **B**) can be used to estimate the weak scaling efficiency, with an adjusted metric defined as:

$$E'_w = \frac{T(n_A)}{T(n_B)} \cdot \frac{P(n_B)/n_B}{P(n_A)/n_A}\,, \tag{3}$$

where $P(n)$ is the total problem size (number of particles) solved with n processors. Considering only the terrain subsystem, we obtain $E'_w = 0.852$; the decreased efficiency can be explained by the load imbalance introduced by the additional proxy bodies representing the tire mesh within the terrain subsystem. With the setup considered here, in both cases **A** and **B** advancing the state of the terrain subsystem between two co-simulation communication points is done completely in "the shadow" of the flexible tire simulations; in fact, such simulations could have been performed with significantly higher granular resolution (or alternatively on many fewer MPI ranks) without affecting the time to solution. Furthermore, comparing the overall wall-clock time to the cost of the slowest tire node, it can be seen that the communication overhead in the outer MPI layer – dedicated to implementing the force–displacement co-simulation

Problem setup and simulation parameters

	A	**B**	**C**	**D**
Maneuver	Acceleration	Acceleration	DLC	DLC
Tire model	ANCF	ANCF	Rigid	Rigid
Domain size [m × m]	8×3	8×3	110×6	110×6
Particle radius[mm]	12.5	10.0	12.5	12.5
Number particles	283,162	591,090	6,513,518	6,513,518
Step-size [ms]	0.04	0.04	0.04	0.04
Number MPI ranks	$5 + 8$	$5 + 16$	$5 + 16$	$5 + 32$

Average timing information, per-step [ms]

	A	**B**	**C**	**D**
Vehicle subsystem	0.93	0.93	0.92	0.92
Terrain subsystem	348.03	391.57	3953.67	1992.25
Tire 0 (front-left)	3467.53	3072.81	1.37	1.39
Tire 1 (front-right)	3435.08	3444.10	1.42	1.41
Tire 2 (rear-left)	3483.08	3488.41	1.46	1.41
Tire 3 (rear-right)	3472.58	2977.90	1.44	1.39
Overall	3547.53	3545.19	3966.21	2002.35

Table 3: Problem setup and average per-step timing for the 4 vehicle co-simulation experiments. All simulations used a step-size of $4 \cdot 10^{-5}$ s. Simulations **A** and **B** involved ANCF-based flexible tires, while **C** and **D** used rigid-mesh tires. All timing information is reported in milliseconds and represent cost per co-simulation step.

data exchange – is minimal and represents only 1.6% – 1.8% of the overall time to solution.

Similarly, the double-lane change simulations presented in Section 4.2 can be used to assess the strong scaling characteristics of the distributed terrain simulation when incorporated in the co-simulation framework. This is because these runs were performed using rigid-mesh tires, thus effectively taking out of the equation the integration cost on the tire nodes. To accommodate the fact that timing results are available only on 16 and 32 processors in the terrain MPI intra-communicator, we use the modified efficiency metric:

$$E'_s = \frac{T(n_C)}{T(n_D)} \cdot \frac{n_D}{n_C} \tag{4}$$

which, using the timing information in Table 3, results in a value of $E'_s = 0.99$ indicating close to perfect strong scaling of the terrain calculation. Here again, it can be seen that the overhead of communication overhead in the outer MPI layer is negligible (0.3% – 0.5%).

5 CONCLUSIONS

We have discussed the architecture and design of a hybrid MPI-MPI-OpenMP co-simulation framework and demonstrated its applicability to large-scale off-road vehicle mobility simulations. Taking advantage of the hardware configuration of current supercomputer clusters, the proposed methodology leverages parallelism at multiple levels, from multiple node distributed MPI, to multi-core shared-memory within a node, and SIMD instruction-level parallelism within a thread. Scaling analysis results, for both stand-alone granular dynamics problems and within the co-simulation framework itself, demonstrate that the proposed solution can effectively take advantage of available computing resources and enable simulations of large terrain patches, highly-resolved granular material, or both. This is shown to be possible at the computational cost dictated by the simulation of a single flexible FEA tire subsystem.

Future work. Chrono::Distributed was designed and architected as a stand-alone Chrono module dedicated to providing an MPI-based distributed parallel simulation capability for generic granular problems using domain-decomposition. However, in its first incarnation it is tailored to problems that are typical of simulating deformable terrain. Most notably, it is currently assumed that there is little particle migration between subdomains. For the scaling characteristics demonstrated here for settling simulations to carry over to problems such as granular flow, additional work is required to implement dynamic load-balancing. Along the same lines, future development plans include the ability to perform domain decomposition along one, two, or three axes, not necessarily aligned with the global coordinate frame. Furthermore, Chrono::Distributed will be extended to optionally use the DEM-C approach (complementarity-based frictional contact formulation) by providing support for the distributed solution of the resulting differential-algebraic-variational problem and associated conic-constrained quadratic optimization problem.

As demonstrated by the numerical experiments provided in this paper, the adoption of a distributed parallel solution for the granular terrain simulation achieves the stated goal of ensuring that the computational bottleneck is transferred to the FEA-based tire simulation. Ongoing work is dedicated to accelerating and increasing performance of the Chrono::FEA module by improving the element formulation and implementation, enhancing the numerical integration and implementing a time-adaptive scheme, and exploring alternative parallelization techniques (multi-core or GPU).

Finally, we plan on enhancing the co-simulation framework itself by formalizing it into a dedicated general-purpose Chrono module, suitable to address complex multi-physics problems that incorporate multibody systems, rigid and flexible bodies, granular material, and fluid phase.

Acknowledgments

Chrono development was supported in part by U.S. Army TARDEC Rapid Innovation Fund grant No. W911NF-13-R-0011, Topic No. 6a, "Maneuverability Prediction". Support for the development of Chrono::Vehicle was provided by U.S. Army TARDEC grant W56HZV-08-C-0236.

REFERENCES

[1] R. B. Ahlvin and P. W. Haley. *NATO Reference Mobility Model: Edition II. NRMM User's Guide*. US Army Engineer Waterways Experiment Station, 1992.

[2] M. McCullough, P. Jayakumar, J. Dasch, , and D. Gorsich. The next generation nato reference mobility model development. In *Proc. 8th Americas Regional Conference of the International Society for Terrain-Vehicle Systems*, 2016.

[3] A. M. Recuero, R. Serban, B. Peterson, H. Sugiyama, P. Jayakumar, and D. Negrut. A high-fidelity approach for vehicle mobility simulation: Nonlinear finite element tires operating on granular material. *Journal of Terramechanics*, 72:39 – 54, 2017.

[4] R. Serban, D. Negrut, A. M. Recuero, and P. Jayakumar. A co-simulation framework for high-performance, high-fidelity simulation of ground vehicle-terrain interaction. *Intl. J. Vehicle Performance*, in press, 2018.

[5] Tamer M Wasfy, Hatem M Wasfy, and Jeanne M Peters. Coupled multibody dynamics and discrete element modeling of vehicle mobility on cohesive granular terrains. In

ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pages V006T10A050–V006T10A050. American Society of Mechanical Engineers, 2014.

[6] OpenMP. Specification Standard 4.5. Available online at `http://openmp.org/wp/`, 2017.

[7] David A Patterson and John L Hennessy. *Computer organization and design: the hardware/software interface*. Morgan Kaufmann, Burlington, Massachusets., 5 edition, 2013.

[8] Message P Forum. MPI: A message-passing interface standard. Technical report, Knoxville, TN, USA, 1994.

[9] A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. Fleischmann, M. Taylor, H. Sugiyama, and D. Negrut. Chrono: An open source multi-physics dynamics engine. In T. Kozubek, editor, *High Performance Computing in Science and Engineering – Lecture Notes in Computer Science*, pages 19–49. Springer, 2016.

[10] Project Chrono. Chrono: An Open Source Framework for the Physics-Based Simulation of Dynamic Systems. `http://projectchrono.org`. Accessed: 2016-03-07.

[11] Project Chrono Development Team. Chrono: An Open Source Framework for the Physics-Based Simulation of Dynamic Systems. `https://github.com/projectchrono/chrono`. Accessed: 2017-05-07.

[12] R. Serban, M. Taylor, D. Negrut, and A. Tasora. Chrono::Vehicle Template-Based Ground Vehicle Modeling and Simulation. *Intl. J. Veh. Performance, accepted*, in press, 2018.

[13] H. Yamashita, A. Valkeapaa, P. Jayakumar, and H. Sugiyama. Bi-linear shear deformable ANCF shell element using continuum mechanics approach. In *Proceedings of ASME International Conference on Multibody Systems, Nonlinear Dynamics, and Control, Buffalo, NY*. ASME, 2014.

[14] Hiroki Yamashita, Antti I Valkeapää, Paramsothy Jayakumar, and Hiroyuki Sugiyama. Continuum mechanics based bilinear shear deformable shell element using absolute nodal coordinate formulation. *Journal of Computational and Nonlinear Dynamics*, 10(5):051012, 2015.

[15] B. Peterson and A. M. Recuero. Validation of a bilinear, gradient-deficient ancf shell element in Chrono::FEA – `http://www.projectchrono.org/assets/validations/fea/ancfshellvalidation.pdf`. Technical report, University of Wisconsin–Madison, 2016.

[16] A. Pazouki, M. Kwarta, K. Williams, W. Likos, R. Serban, P. Jayakumar, and D. Negrut. Compliant versus rigid contact: A comparison in the context of granular dynamics. *Phys. Rev. E*, 96, 2017.

[17] M. Anitescu and A. Tasora. An iterative approach for cone complementarity problems for nonsmooth dynamics. *Computational Optimization and Applications*, 47(2):207–235, 2010.

[18] D. Negrut, R. Serban, and A. Tasora. Posing multibody dynamics with friction and contact as a differential complementarity problem. *ASME Journal of Computational and Nonlinear Dynamics*, 13(1):014503, 2017.

[19] P. Cundall and O. Strack. A discrete element model for granular assemblies. *Geotechnique*, 29:47–65, 1979.

[20] C. O'Sullivan. "Particle-Based Discrete Element Modeling: Geomechanics Perspective". *Int. J. Geomech.*, 11(6):449–464, 2011.

[21] Nathan Bell and Jared Hoberock. Thrust: A productivity-oriented library for cuda. *GPU computing gems – Jade edition*, 2:359–371, 2011.

[22] K. Iglberger, G. Hager, J. Treibig, and U. Rude. High performance smart expression template math libraries. In *High Performance Computing and Simulation (HPCS), 2012 International Conference on*, pages 367–373, July 2012.

[23] H. Mazhar. *Physics Based Simulation Using Complementarity and Hybrid Lagrangian-Eulerian Methods*. PhD thesis, University of Wisconsin–Madison, 2016.

[24] Hiroki Yamashita, Paramsothy Jayakumar, and Hiroyuki Sugiyama. Physics-based flexible tire model integrated with lugre tire friction for transient braking and cornering analysis. *Journal of Computational and Nonlinear Dynamics*, 11(3):031017, 2016.