Chrono: A Parallel Multi-Physics Library for Rigid-Body, Flexible-Body, and Fluid Dynamics

Toby Heyn[1], Hammad Mazhar[1], Arman Pazouki[1], Dan Melanz[1], Andrew Seidl[1], Aaron Bartholomew[1], Luning Fang[1], Alessandro Tasora[2], Eftychios Sifakis[3], Dan Negrut[1]

[1] Simulation Based Engineering Lab, Department of Mechanical Engineering, University of Wisconsin, Madison, WI, 53706: {heyn,hmazhar,pazouki,melanz,aaseidl,abartholome2,lfang9,negrut}@wisc.edu

[2] Department of Industrial Engineering, University of Parma, V.G.Usberti 181/A, 43100, Parma, Italy: tasora@ied.unipr.it

[3] Department of Computer Science, University of Wisconsin, Madison, WI, 53706: sifakis@cs.wisc.edu

Abstract

The last decade witnessed a manifest shift in the microprocessor industry towards chip designs that promote parallel computing. Until recently the privilege of a select group of large research centers, Teraflop computing is becoming a commodity owing to inexpensive GPU cards and multi to many-core x86 processors. This paradigm shift towards large scale parallel computing has been leveraged in Chrono, a freely available C++ multi-physics simulation package. Chrono is made up of a collection of loosely coupled components that facilitate different aspects of multi-physics modeling, simulation, and visualization. This contribution provides an overview of Chrono::Engine, Chrono::Flex, Chrono::Fluid, and Chrono::Render, which are modules that can capitalize on the processing power of hundreds of parallel processors. Problems that can be tackled in Chrono include but are not limited to granular material dynamics, tangled large flexible structures with self contact, particulate flows, and tracked vehicle mobility.

The Chrono::Engine software is a general-purpose simulator for three dimensional multi-body problems. Target applications include tracked vehicles operating on granular terrain or the Mars Rover operating on discrete granular soil. In these applications, it is desirable to model the granular terrain as a collection of many thousands or millions of discrete bodies interacting through contact, impact, and friction. Note that such systems also include mechanisms composed of rigid bodies and mechanical joints. These challenges require an efficient and robust simulation tool. Chrono::Engine was initially developed leveraging the Differential Variational Inequality (DVI) formulation as an efficient method to deal with problems that encompass many frictional contacts - a typical bottleneck for other types of formulations [1]. This approach enforces non-penetration between rigid bodies through constraints, leading to a cone-constrained quadratic optimization problem which must be solved at each time step. Chrono::Engine has since been extended to also support the Discrete Element Method (DEM) formulation for handling the frictional contacts present in granular dynamics problems. This formulation computes contact forces by penalizing small interpenetrations of colliding rigid bodies. Various contact force models can be used depending on the application. Chrono::Engine is capable of leveraging a Graphics Processing Unit (GPU) to accelerate the solution of the collision detection problem and the dynamics problem. Chrono::Engine has been further extended to allow the use of CPU parallelism for certain problems, leveraging a domain-decomposition approach to allow the use of multicore compute clusters.

The Chrono::Flex software is a general-purpose simulator for three dimensional flexible multi-body problems and provides a suite of flexible body support. The implementation is based on the Absolute Nodal Coordinate Formulation (ANCF) and provides gradient-deficient beam elements and gradient-deficient plate elements [2, 3]. Elements may interact with each other through bilateral constraints, mechanical joints, or friction and contact. The equations of motion are solved by a fully-implicit integration scheme. Additionally, Chrono::Flex leverages the GPU to accelerate solution of large problems.

The Chrono::Fluid component aims at leveraging GPU computing to efficiently simulate fluid dynamics and fluid-solid interaction problems. Fluid-Solid Interaction (FSI) covers a wide range of applications, from blood and polymer flow to tanker trucks and ships. Simulation of the FSI problem requires two components: fluid and solid simulations. Simulation of the solid phase, either rigid or flexible, is described in previous paragraphs. To leverage the existing solid phase simulation, the fluid flow simulation should satisfy some conditions, introduced by the aforementioned target problems. First, the fluid flow may experience large domain deformation due to the motion of the solid phase. Second, the two phases should be coupled via an accurate algorithm. Third, target problems may experience free surface as well as internal flow. Finally, the whole simulation should be capable of an HPC implementation to maintain the scalability

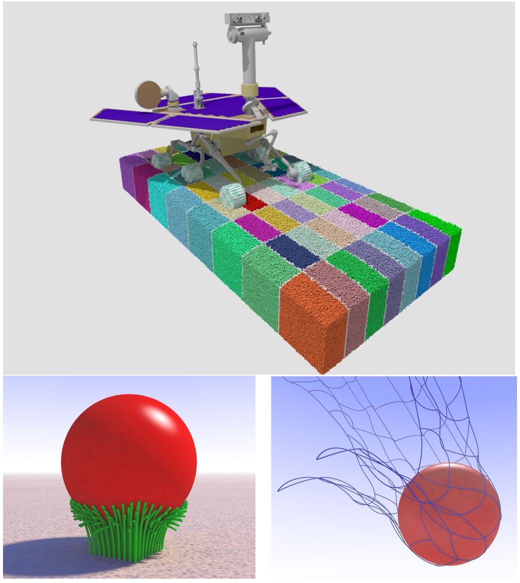

Figure 1. (Top) Chrono::Engine simulation of Mars Rover with 2,016,000 terrain particles using 64 compute cores in domain-decomposition approach. (Bottom) Two models with friction and contact using Chrono::Flex beam elements: a ball sitting on grass-like beams and a ball hitting a net.

of the code. To satisfy these conditions, this module leverages a GPU-parallel implementation of the Smoothed Particle Hydrodynamics (SPH) formulation for the fluid component [4].

Chrono::Render is a software package that enables simple, streamlined, and fast visualization of arbitrary data using Pixar's RenderMan. Specifically, Chrono::Render contains a hybrid of processing binaries and Python scripting modules that seek to abstract away the complexities of rendering with RenderMan. Additionally, Chrono::Render is targeted for providing rendering as an automated post-processing step in a remote simulation pipeline, hence it is controlled via a succinct XML specification for "gluing" together rendering with arbitrary processes. Chrono::Render is capable of combining simulation data, XML describing how to use the data, and optional user-defined Python scripts into a complex, visually-rich scene to be rendered by RenderMan. The guiding principle of Chrono::Render is to make high-quality rendering available to researchers, most of whom don't have the background or bandwidth to spend time learning how to use complex graphics applications. The user must only instance the correct XML components to achieve high-quality renders. Currently, up to 320 frames of simulation data can be rendered in parallel leveraging RenderMan and a compute cluster with sufficient CPU cores.

Chrono is currently free for use both for research and commercial purposes. For users without access to the advanced hardware necessary to take capitalize on the parallel computing features of Chrono, this modeling/simulation/visualization infrastructure will become available in the future in the form of Dynamics-as-a-Service, where the bulk of the computational load is offloaded to remote specialized servers. This mechanism will enable remote users to use Chrono without the need to worry about build issues or deployment on hardware assets that might exceed budget constraints experienced by potential users.

References

[1] A. Tasora and M. Anitescu. A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics. *Computer Methods in Applied Mechanics and Engineering*, 200(5-8):439–453, 2011.

[2] J. Gerstmayr and A.A. Shabana. Analysis of thin beams and cables using the absolute nodal co-ordinate formulation. *Nonlinear Dynamics*, 45(1):109–130, 2006.

[3] K. Dufva and AA Shabana. Analysis of thin plate structures using the absolute nodal coordinate formulation. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, 219(4):345–355, 2005.

[4] A. Pazouki and D. Negrut. Numerical investigation of particle distribution in Poiseuille flow of suspension. *Journal of Computational Physics*, 2013. submitted.