International Design Engineering Technical Conferences and Computers and Information in Engineering Conference
IDETC-CIE 2025
August 17 – 21, 2025, Anaheim, CA

ON THE USE OF GPU-BASED LINEAR SOLVERS IN MULTIBODY DYNAMICS

Ganesh Arivoli[1], Huzaifa Unjhawala[1], Michael Taylor[1], Radu Serban[1], Dan Negrut[1,*],

[1]Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI

1. INTRODUCTION

The simulation of Multibody Dynamics systems with flexible bodies and smooth or non-smooth frictional contact requires repeatedly the solution of linear systems. Systems with flexible bodies can have very large degree of freedom (DoF) counts, making the linear system large and computationally expensive to solve. In many applications, such as deformable tires and tracks, the use of continuum-based Absolute Nodal Coordinate Formulation (ANCF) [1] draws on an implicit integrator (e.g., HHT [2]) along with a Newton-Raphson procedure, where at each iteration a sparse linear system needs to be solved. This step is computationally expensive and represents a dominant portion of the overall cost in ANCF-based simulations [3, 4].

Recent work has underscored the limitations of GPU-accelerated sparse direct solvers, particularly their lack of scalability and robustness when applied to ill-conditioned symmetric systems. While GPUs often perform well on dense or well-conditioned problems, these challenges have limited their broader adoption [5]. In this abstract, we investigate the feasibility of using GPUs and recent GPU-based linear solvers to address linear systems commonly encountered in multibody dynamics simulations. Specifically, we use Chrono [6] to extract system matrices from two flexible body simulations of an airless tire modeled with the Absolute Nodal Coordinate Formulation (ANCF). The number of "spokes" in the tire model is varied from 16 to 80 (see Fig. 1) to generate linear systems of different sizes. We evaluate both the performance and accuracy of Nvidia CuDSS, a recently introduced GPU-based direct solver, against Pardiso, a state-of-the-art CPU-based direct solver currently integrated with Chrono. This study also lays the foundation for a broader evaluation of GPU-based direct and iterative solvers across a diverse set of linear systems arising in multibody dynamics applications.

2. METHODS

Here we introduce the deformable ANCF tire simulation from which the linear systems are extracted, the GPU and CPU direct solvers applied to them, and the metrics used to compare solver speed and accuracy.

*Corresponding author: negrut@wisc.edu

2.1 Problem Description

The multibody dynamics problem of interest involves a deformable tire modeled using ANCF with smooth contact. We employ a full Newton-Raphson approach coupled with an implicit HHT integrator, requiring the solution of a linear system at each iteration.

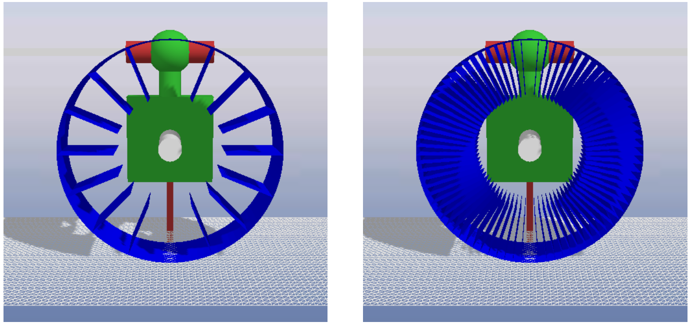
Two variants of the ANCF tire model are evaluated — a 16-"spoke" configuration which includes 384 nodes and 288 elements and a 80 "spokes" configuration which comprises 1,920 nodes and 1,440 elements. A tire test rig is modeled to constrain the tire and apply a constant angular velocity to the wheel. This setup introduces some rigid bodies and joint constraints, leading to system matrices of size 4973×4973 and 83485×83485 for the two tire configurations, respectively. A deformable terrain is also included in the simulation, modeled using the Soil Contact Method (SCM) [7, 8]. As SCM is a force-based model, it does not add additional DOF's to the system. Table 1 summarizes the associated system matrix sizes and sparsity characteristics for the two configurations.

TABLE 1: LINEAR SYSTEM SIZE AND CHARACTERISTICS FOR THE TWO VARYING TIRE CONFIGURATIONS

Spokes	Matrix Size $A \in \mathbb{R}^{n \times n}$	Nonzero Elements
16	4973×4973	419,248
80	83485×83485	7,917,936

2.2 Solvers Compared

The system matrices are saved from Chrono half-way through the simulation. Two solvers are then used to solve the linear system externally and are compared for accuracy and solve time: (1) **NVIDIA CuDSS** is a GPU-accelerated direct sparse solver library by NVIDIA for sparse linear systems, leveraging cuSPARSE and GPU memory optimizations for efficient factorization [9]; (2) **oneMKL PARDISO** is a high-performance CPU-based direct solver for sparse linear systems, integrated into the Chrono framework and commonly used in flexible body simulations with smooth contact [10].



(a) 16-spokes tire model. **(b) 80-spokes tire model.**

FIGURE 1: TWO VARIANTS OF THE AIRLESS DEFORMABLE TIRE.

2.3 Evaluation Metric

For accuracy, the **Relative Backward Error (RBE)** is used as a solver-independent metric to assess the accuracy of the computed solution. The RBE is calculated as

$$\text{RBE} = \frac{\|Ax - b\|_2}{\|A\|_2 \|x\|_2 + \|b\|_2} \tag{1}$$

where A is the system matrix, x is the computed solution, and b is the right-hand side vector. Additionally, we compare the "solve time" of the two solvers, which includes time taken for symbolic analysis, numerical factorization, and the solve step. For Pardiso, timing is measured using `std::chrono` whereas for CuDSS we use CUDA events. CuDSS is run on an NVIDIA RTX 4080 Ti GPU and Pardiso on a 13th Gen Intel Core i7-13700K CPU with 16 threads.

3. RESULTS

The solve time and Relative Backward Error (RBE) for the two solvers is summarized in Table 2.

TABLE 2: SOLVER PERFORMANCE AND ACCURACY ACROSS THE TWO TIRE CONFIGURATIONS

Spokes	Solver	Solve Time (ms)	RBE
16	Pardiso	30.04	**1.36E-18**
	CuDSS	**24.55**	3.42E-07
80	Pardiso	482.69	**3.68E-20**
	CuDSS	**130.53**	2.87E-10

On the 16-spoke matrix, CuDSS is about 20% faster than Pardiso; on the 80-spoke matrix, it is 3.6x faster. Accuracy remains acceptable for CuDSS, but Pardiso still achieves relative backward errors several orders of magnitude lower in both cases. The code and data necessary to reproduce these results are available in our public repository at https://github.com/uwsbel/sbel-reproducibility/tree/master/2025/ASME-LinearSolversGPU/.

4. CONCLUSIONS AND FUTURE WORK

Initial tests confirm that GPU-accelerated sparse solvers (e.g., CuDSS) can significantly reduce solve times for high-DOF multibody systems, yet their relatively modest loss in accuracy merits closer study. In future work we will (1) broaden the solver suite to include both direct and iterative GPU methods, (2) exercise them on a larger spectrum of Chrono problems, including rigid and flexible-body dynamics with smooth as well as non-smooth contact, and (3) extend the study to linear systems emerging from implicit Smoothed Particle Hydrodynamics (SPH). Most importantly, we will embed these GPU solvers inside Chrono's time-integration loop, enabling true end-to-end benchmarks that capture factorization cost, communication latency, and data-transfer overhead between host and device—all factors omitted from the present, offline comparison.

REFERENCES

[1] A. A. Shabana. *Dynamics of Multibody Systems*. Cambridge University Press, Cambridge, England, fifth edition, 2020.

[2] Dan Negrut, Rajiv Rampalli, Gisli Ottarsson, and Anthony Sajdak. On the use of the HHT method in the context of index 3 Differential Algebraic Equations of multibody dynamics. *ASME JCND*, 2, 2007.

[3] Michael Taylor, Radu Serban, and Dan Negrut. Implementation implications on the performance of ANCF simulations. *International Journal of Non-Linear Mechanics*, 149:104328, 2023.

[4] Michael Taylor, Radu Serban, and Dan Negrut. An efficiency comparison of different ANCF implementations. *International Journal of Non-Linear Mechanics*, 149:104308, 2023.

[5] Kasia Swirydowicz, Eric Darve, Wesley Jones, Jonathan Maack, Shaked Regev, Michael A. Saunders, Stephen J. Thomas, and Slaven Peleš. Linear solvers for power grid optimization problems: A review of gpu-accelerated linear solvers. *Parallel Computing*, 111:102870, 2022.

[6] 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 International Publishing, 2016.

[7] Alessandro Tasora, Dario Mangoni, Dan Negrut, Radu Serban, and Paramsothy Jayakumar. Deformable soil with adaptive level of detail for tracked and wheeled vehicles. *International Journal of Vehicle Performance*, 5(1):60–76, 2019.

[8] Radu Serban, Jay Taves, and Zhenhao Zhou. Real-time simulation of ground vehicles on deformable terrain. *Journal of Computational and Nonlinear Dynamics*, 18(8):081007, 2023.

[9] NVIDIA Corporation. NVIDIA cuDSS: GPU-Accelerated Direct Sparse Solver Library. https://developer.nvidia.com/cudss, 2025. Version 0.5.0, Preview Release.

[10] Intel Corporation. *Intel oneAPI Math Kernel Library PARDISO*, 2024. Version 2024.2.

