ACCELERATING VIRTUAL PROTOTYPING AND CERTIFICATION IN THE AEROSPACE INDUSTRY WITH SCALABLE FINITE-ELEMENT ANALYSIS

Allocation: Director Discretionary/150 Knh
PI: Seid Koric[1]
Co-PIs: Robert F. Lucas[2], Erman Guleryuz[3]

[1]University of Illinois at Urbana–Champaign
[2]Livermore Software Technology Corp.
[3]National Center for Supercomputing Applications

EXECUTIVE SUMMARY

The aerospace industry increasingly relies on physics-based modeling and simulation for the design and analysis of systems in complex engineering products. Advanced modeling and simulation techniques such as finite element analysis (FEA) are being used to replace physical testing with virtual testing and mitigate the costs and risks of certification. Improved fidelity in simulation models generally requires explicit modeling of smaller physical details, which leads to larger models with finer grids. To further the state of the art in computationally intensive implicit FEA and to reduce time-to-solution, the research group studied real-life jet engine models with LS–DYNA software and removed performance scaling barriers. The team solved an implicit finite-element model with over 200 million equations in-core on Blue Waters. This work demonstrated the feasibility of efficiently solving extreme-size real-world multiphysics problems at peta- and potentially exascale levels, thus adding novel value to engineering research by enabling the creation of high-fidelity models that yield detailed insight into the performance and safety of proposed engineering designs.

RESEARCH CHALLENGE

The imperative to design higher-performing products pushes technological advances in advanced computational analysis techniques such as FEA. FEA allows gas-turbine engine designers to develop systems that continue to increase thrust and fuel efficiency while meeting a complex set of requirements, including minimizing weight, maintaining part life, and reducing manufacturing costs. In practice, large finite element method (FEM) models today use hundreds of cores, and large original equipment manufacturers are evaluating the use of thousands of cores. More realistic models with higher fidelity will require tens of thousands of cores and even, in the near future, hundreds of thousands. The growth in high-performance computing (HPC) resources required to run larger, more realistic models is complemented by the industry's need to reduce the time for each simulation so they can be included in the design cycle rather than remain a research function.

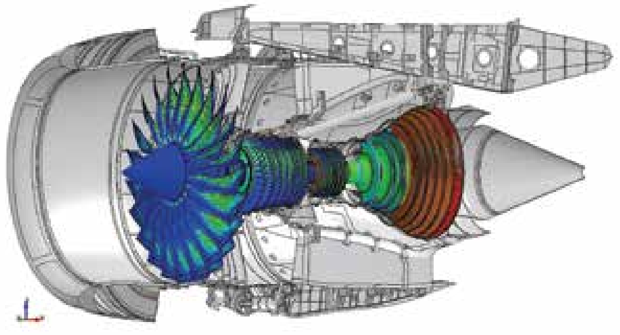

Figure 1: A cross-section of the Rolls-Royce jet engine model.

Several barriers exist today to increasing realism and adding optimization to the design cycle: (1) single simulations are not yet realistic enough, using too few cores; (2) commercial software developers have not had sufficient access to HPC to modernize their codes; and (3) commercially accessible HPC of sufficient scale is not available yet to foster stochastic optimization. This project addresses the technical barriers of scale and access for developers.

METHODS & CODES

The aerospace industry relies on advanced commercial FEA codes that are used for nonlinear, quasistatic, and dynamic deformations in computational structural models. LS–DYNA is a prominent FEA code used widely in the industry. This research has been performed with LS–DYNA and with modules of the LS–DYNA code, including a graph-partitioning scheme and the factorization kernel used in the direct solver. The aerospace community uses LS–DYNA for extreme-event modeling for gas-turbine engines such as bird-strikes, fan-blade-off, and whole engine thermomechanical analysis. Bird-strike and fan-blade-off testing is mandated by the Federal Aviation Administration and European Aerospace Safety Agency to certify the safety of jet engines. These tests must demonstrate that the engine casing can contain the debris from the rotors and that engine mounts do not fail.

To further the development of FEA to reduce time-to-solution, Rolls-Royce developed representative models of a gas-turbine engine for research (Fig. 1). As part of a multiyear research effort, the multidisciplinary research team formed by collaborators from NCSA, Rolls-Royce, Livermore Software Technology Corp., and Cray has worked to improve the scalability of LS–DYNA using real-world engine models developed specifically for scaling studies on supercomputers.

RESULTS & IMPACT

The computational time of implicit FEA is dominated by the analysis and solution of a sparse linear system of equations. A single solution of the system suffices for linear problems. However, for nonlinear problems, within each quasistatic timestep a system of nonlinear equations is linearized and solved with a Newton–Raphson iteration scheme, often requiring several linear solver solutions. Stiff or ill-conditioned linear systems arise in several application areas. These usually involve fine-grain, unstructured computational meshes on irregular geometries, and the coupling of multiple models (*e.g.*, multiscale, multiphysics). For these types of challenging systems, implicit methods with direct solvers are often the most robust solution methods. LS–DYNA offers a multifrontal direct solver for these numerically challenging linear systems of equations.

The implicit analysis has the following main steps: matrix assembly, constraint processing, matrix reordering, symbolic factorization, numeric factorization, and triangular solution. Because they have different parallel efficiencies, each step consumes a varying fraction of the wall-clock time as the processor count grows. A remarkable increase in efficiency (Fig. 2b) has been achieved since the beginning of this multiyear effort, to the point where LS–DYNA can successfully run the largest engine models on tens of thousands of cores on Blue Waters. The effort included application- and hardware-level profiling, creation of novel scalable algorithms, memory management improvements, computation workflow modifications, and improvements to reduce previously unknown Amdahl (*i.e.*, serial) fractions. The computational complexity of the numerical pipeline is centered on the sparse matrix factorization step.

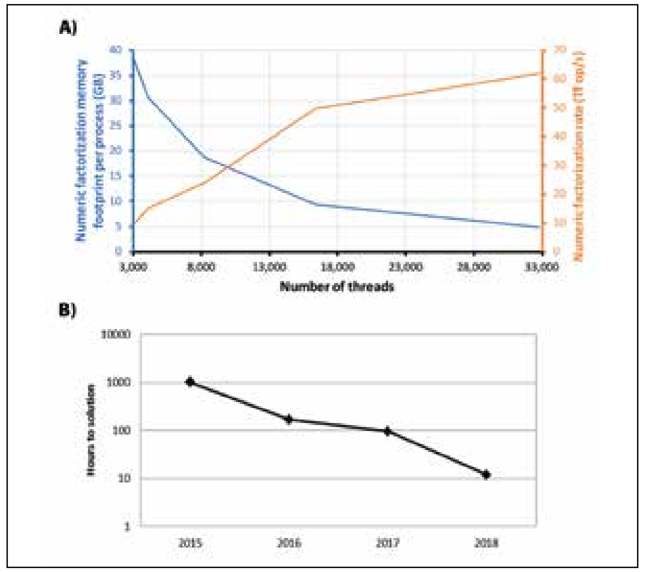

Figure 2: (a) Parallel scaling of arithmetic throughput and memory footprint for numeric factorization. A hybrid (MPI and OpenMP) parallelization model (eight threads per MPI rank) is used to solve the finite-element model with 200 million equations. (b) Aggregate improvement in time-to-solution during the course of the multiyear research effort.

The team's previous research proved for the first time that a sparse matrix factorization algorithm can scale on tens of thousands of computational cores [1]. Floating-point arithmetic throughput rate and memory utilization amount are two major metrics that characterize performance scaling of factorization. Fig. 2a shows strong scaling results for the multifrontal solver's factorization kernel, when factoring symmetric indefinite sparse matrices. An increase in computing operations rate and reduction in memory storage needed per Message-Passing Interface rank has been observed for up to 33,000 OpenMP threads. This sustained performance increase indicates potential for more performance growth. This work constitutes, to the best of the researchers' knowledge, the largest implicit LS–DYNA calculations to date, and the results have been presented in technical conferences [2–4].

WHY BLUE WATERS

Blue Waters—with large amounts of distributed memory, thousands of multicore processors, a low-latency file system, and increased bandwidth of advanced interconnect technologies—enabled the research team to address parallel processing challenges and to demonstrate that implicit FEA of large-scale models could be performed in a timely manner using large-scale computing systems. This research on Blue Waters has allowed computer-aided engineering to have a greater impact on the design cycle for new engines, and is a step toward the long-term vision of digital twins, *i.e.*, computer-generated representations of complex engineering systems.

PUBLICATIONS & DATA SETS

D. Liu, S. Koric, and A. Kontsos, "Parallelized finite element analysis of knitted textile mechanical behavior," *J. Eng. Mater. Technol.*, vol. 141, no. 2, p. 021008, Dec. 2018.

R. Borrell, J. C. Cajas, D. Mira, A. Taha, S. Koric, M. Vázquez, and G. Houzeaux, "Parallel mesh partitioning based on space filling curves," *Comput. Fluids*, vol. 173, p. 264, Sept. 2018.

F. A. Sabet, O. Jin, S. Koric, and I. Jasiuk, "Nonlinear micro-CT based FE modeling of trabecular bone—Sensitivity of apparent response to tissue constitutive law and bone volume fraction," *Int. J. Numer. Meth. Biomed. Engng.*, vol. 34, no. 4, p. 2941, Nov. 2017.

A. Gupta, N. Gimelshein, S. Koric, and S. Rennich, "Effective minimally-invasive GPU acceleration of distributed sparse matrix factorization," in *Euro-Par 2016: Parallel Processing, Lecture Notes in Computer Science*, P.F. Dutot, and D. Trystram, Eds., vol. 9833. Cham, Switzerland: Springer, 2016, p. 672.

M. Vázquez *et al.*, "Alya: Multiphysics engineering simulation toward exascale," *J. Comput. Sci.*, vol. 14, p. 15, May 2016.

V. Puzyrev, S. Koric, and S. Wilkin, "Evaluation of parallel direct sparse linear solvers in electromagnetic geophysical problems," *Comput. Geosci.*, vol. 89, p. 79, Apr. 2016.