Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 302 (2016) 281–292
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Sparse matrix factorization in the implicit finite element method on petascale architecture

Seid Koric[a,b,*], Anshul Gupta[c]

[a] National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, USA
[b] Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, USA
[c] Mathematical Sciences, IBM T.J. Watson Research Center, USA

Received 2 May 2015; received in revised form 12 January 2016; accepted 15 January 2016
Available online 27 January 2016

Abstract

The performance of the massively parallel direct multifrontal solver Watson Sparse Matrix Package (WSMP) for solving large sparse systems of linear equations arising in implicit finite element method on unstructured (free) meshes in solid mechanics was evaluated on one of the most powerful supercomputers currently available to the open science community—the sustained petascale high performance computing system of Blue Waters. We have performed full-scale benchmarking tests up to 65,536 cores using assembled global stiffness matrices and load vectors ranging from 11 to 40 million unknowns extracted from "real-world" commercial implicit finite element analysis (FEA) applications. The results show that a direct multifrontal factorization method with a hybrid parallel implementation in WSMP performs exceedingly well on a petascale high-performance computing (HPC) system, and delivers superior factorization time and parallel scalability, thus opening the door for the high fidelity modeling of complex industrial structures and assemblies in real scale.

Keywords: Sparse linear solvers; Factorization; Petascale high performance computing; Finite element method; Unstructured mesh

1. Introduction

1.1. High performance computing in engineering

Across a range of engineering fields, the use of simulation and computational models is pervasive for designing engineered systems. High Performance Computing (HPC) systems play an essential role in simulations and modeling. Researchers and manufacturing teams depend on HPC to create safe cars and energy-efficient aircraft as well as effective communication systems and efficient supply chain models. Availability of advanced HPC technologies has also fundamentally altered the investigative paradigm in the field of biomechanics. While emerging peta-scale

* Correspondence to: National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, 1205 W. Clark St., Urbana, IL, 61801, USA.
E-mail address: koric@illinois.edu (S. Koric).

http://dx.doi.org/10.1016/j.cma.2016.01.011
0045-7825/

computing is already a strategic enabler of large-scale simulations in many scientific areas such as astronomy, biology and chemistry [1–3], paradoxically for many engineers and researchers, the existing hardware and software often cannot be used to solve their problems. On one hand, current HPC systems in production often lack the computational power, network bandwidth and data storage needed for solving tomorrow's real-world engineering challenges. On the other hand, even the most powerful hardware will fail to deliver on its full potential unless matched with appropriate algorithms designed specifically for such environments. Sparse matrix factorization, a critical algorithm in many science, engineering, and optimization applications, has traditionally had difficulty tuning to and leveraging the ever increasing computational power of HPC [4].

The main objective of this work is to demonstrate that the multifrontal sparse factorization algorithm with hybrid parallelization, such as the one in the WSMP solver code, can scale efficiently in today's large-scale supercomputers, opening a new horizon of high fidelity and robust finite element simulations in the engineering academic and industrial realms.

1.2. Sparse linear solvers in implicit finite element methods, background and previous work

Solving linear system of equations:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{1}$$

is responsible for 70%–80% of the total computational time in many problems in computational science and engineering. When discretizing continuous solid mechanics problems with implicit finite element method, the associated matrix $\boldsymbol{A}$ is sparse, symmetric and often positive definite. A single solution of Eq. (1) suffices for linear problems. For nonlinear problems, however, within each quasi-static time step, a system of nonlinear equations is linearized and solved with a Newton–Raphson (NR) iteration scheme [5,6], which requires several linear solver solutions of global equilibrium iterations (subscript i) as follows:

$$\left[K_{i-1}^{t+\Delta t}\right]\left\{\Delta u_{i-1}^{t+\Delta t}\right\} = \left\{R_{i-1}^{t+\Delta t}\right\}. \tag{2}$$

Here $\left\{\Delta u_{i-1}^{t+\Delta t}\right\}$ is the incremental change to the solution vector (displacements in mechanical problems), and $\left\{R_{i-1}^{t+\Delta t}\right\}$ is the residual error vector. A linear solver is used to solve Eq. (2) for $\left\{\Delta u_{i-1}^{t+\Delta t}\right\}$, which is used to update the solution vector in Eq. (3), until convergence is achieved everywhere at time $t + \Delta t$ (i.e., when the update vector is sufficiently small).

$$\left\{u_{i}^{t+\Delta t}\right\} = \left\{u_{i-1}^{t+\Delta t}\right\} + \left\{\Delta u_{i-1}^{t+\Delta t}\right\}. \tag{3}$$

The tangent stiffness matrix $\left[K^{t+\Delta t}\right]$ is defined in Eq. (5) from the consistent tangent operator, also known as the material Jacobian, $[J]$, which is defined in Eq. (4) for mechanical problems, taking $\Delta\hat{\boldsymbol{\varepsilon}}^{t+\Delta t}$ as a guessed mechanical strain increment, based on the current best displacement increment.

$$\underline{\underline{\mathbf{J}}} = \frac{\partial \Delta \boldsymbol{\sigma}^{t+\Delta t}}{\partial \Delta \hat{\boldsymbol{\varepsilon}}^{t+\Delta t}} \tag{4}$$

$$\left[K^{t+\Delta t}\right] = \int_V [B]^{T} [J] [B] \, dV. \tag{5}$$

Here $[B] = \partial [N] / \partial \mathbf{x}$ contains the spatial derivatives of the element shape functions $[N]$.

There has been considerable interest in the development of numerical algorithms for solving large sparse linear systems of equations and their efficient parallel implementation on HPC systems for more than three decades. The algorithms may be grouped into two broad categories: direct methods and iterative methods.

Iterative method algorithms repeatedly apply a sequence of operations at each step attempting to improve upon its current approximation to a solution. Krylov subspace methods are an important class of iterative methods. This class includes the Conjugate Gradient (CG) method [7,8] and its variants, which are robust for Symmetric Positive Definite (SPD) matrices. In solving the large systems in finite element method, combining a Krylov subspace method