Applied Mathematics and Computation 267 (2015) 698–710

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Massively parallel solution of elastoplasticity problems with tens of millions of unknowns using PermonCube and FLLOP packages

Alexandros Markopoulos [a,b,*,1], Vaclav Hapla [a,*,1], Martin Cermak [a], Martin Fusek [a,b]

[a] IT4Innovations National Supercomputing Center, VSB – Technical University of Ostrava, 17. listopadu 15/2172, 70833 Ostrava, Czech Republic
[b] Faculty of Mechanical Engineering, VSB – Technical University of Ostrava, 17. listopadu 15/2172, 70833 Ostrava, Czech Republic

ARTICLE INFO

Keywords:
TFETI
Elastoplasticity
Parallel mesh
PERMON
PermonCube
FLLOP

ABSTRACT

In this paper we are presenting our PermonCube and FLLOP packages, and their use for massively parallel solution of elastoplasticity problems. PermonCube provides simple cubical meshes, partitioned in a non-overlapping manner. By means of finite element method it assembles all linear algebra objects required for solution of the physical problem. Two chosen nonlinear material models are presented, and a solving strategy based on the Newton's method is briefly discussed. PermonCube uses our FLLOP library as a linear system solver. FLLOP is able to solve problems decomposed in a non-overlapping manner using domain decomposition methods of the FETI type. It extends PETSc (Portable, Extensible Toolkit for Scientific Computation). In the last section, large-scale numerical experiments with problem size up to 60 million of degrees of freedom are presented.

1. Introduction

The goal of this paper is to present our set of tools for solution of large-scale problems of mechanics. We will be interested particularly in elastoplastic problems and their solution using the Newton's method. In each Newton iteration, the linearized problem is solved using the FETI (Finite Element Tearing and Interconnecting) domain decomposition method (DDM).

We consider elastoplasticity with the von Mises plastic criterion and the kinematic hardening law (see e.g. [56,41,61]). The corresponding elastoplastic constitutive model is discretized by the implicit Euler method in time and consequently a nonlinear stress–strain relation is implemented by the return mapping concept (see e.g. [61,41]). Our approach with the balance equation, the small strain assumption and a combination of the Dirichlet and Neumann boundary conditions leads to solution of a nonlinear variational equation with respect to the primal unknown displacement in each time step. This equation can also be equivalently formulated as a minimization problem with a potential energy functional [64].

By a finite element space discretization of the one time step problem, we get a system of nonlinear equations. A suitable choice for solution of the nonlinear system is the semismooth Newton method since the strong semismoothness together with other properties ensure local quadratic convergence. Semismooth functions in finite dimensional spaces and the

* Corresponding authors at: IT4Innovations National Supercomputing Center, VSB – Technical University of Ostrava, 17. listopadu 15/2172, 70833 Ostrava, Czech Republic (A. Markopoulos).
E-mail addresses: alexandros.markopoulos@vsb.cz (A. Markopoulos), vaclav.hapla@vsb.cz (V. Hapla).
[1] Correspond with Alexandros Markopoulos and Vaclav Hapla for packages PermonCube and FLLOP, respectively.

http://dx.doi.org/10.1016/j.amc.2014.12.097
0096-3003/

semismooth Newton method were introduced in [63]. Semismoothness in elastoplasticity was investigated for example in [64,65].

For the linearized system in each Newton iteration, we can use the FETI DDM, originally introduced by Farhat and Roux [53] and theoretically analyzed by Mandel and Tezaur [60]. In our approach we use modification of FETI method, called TFETI (Total Finite Element Tearing and Interconnecting) [51]. Hence all subdomain stiffness matrices are singular with a priori known kernels. With known kernel basis we can regularize effectively the stiffness matrix without extra fill in and use any standard sparse Cholesky type decomposition method for nonsingular matrices [59,42]. Application of TFETI to problems with classical boundary conditions can be found e.g. in [48,46,47], with contact conditions in [49], and with limit loading in [45].

In the following sections, we present our newly emerging PERMON toolbox, its core solver module FLLOP [34], and the PermonCube module providing massively parallel benchmark generation and Newton iteration. Remaining sections describe the model problem of elastoplasticity, TFETI DDM and its specialization for elastoplasticity, nonlinear material models implemented in PermonCube, and large-scale numerical experiments.

2. Related work

2.1. Open source toolkits for numerical computations

This section discusses well-known general toolkits for numerical computations which provide linear algebra, linear system solvers, preconditioners and so on. They provide infrastructure for partial differential equation (PDE) solution on all major hardware platforms and operating systems of personal computers and supercomputers.

PETSc (Portable, Extensible Toolkit for Scientific Computation) [1–3] is a suite of data structures and routines for the scalable (parallel) solution of scientific applications modeled by partial differential equations. It supports MPI, shared memory pthreads, and GPUs through CUDA or OpenCL, as well as hybrid MPI-shared memory pthreads or MPI-GPU parallelism. It provides parallel matrix data structures and algorithms, preconditioners, linear and nonlinear system solvers, time integration, data structures and operations for unstructured and structured meshes, debugging and profiling tools, and others. PETSc also provides extensive set of interfaces to many external libraries (sparse and dense linear system solvers, preconditioners, input/output, partitioning, and others). This is a popular feature as it greatly reduces effort needed for integration of all those libraries into the user's code. PETSc is written in ANSI C, callable also from C++, FORTRAN, Python and Java. It is distributed under a permissive open source license (2-clause BSD).

There are two important libraries extending PETSc that use the same "look & feel": **SLEPc** (Scalable Library for Eigenvalue Problem Computations) [4,5] for eigenvalue problem solution (licensed LGPL v3), and **TAO** (Toolkit for Advance Optimization) for nonlinear optimization. TAO became a part of PETSc from its 3.5 version.

The **Trilinos** Project [6] is an effort to develop algorithms and enabling technologies using modern object-oriented software design for the solution of large-scale, complex multi-physics engineering and scientific problems, while still leveraging the value of established libraries such as PETSc, METIS, SuperLU, Aztec, the BLAS and LAPACK. It consists of relatively autonomous, loosely coupled packages with common web page, build system, and basic infrastructure. Particularly, its Epetra package provides the parallel sparse linear algebra foundation layer. Its more modern, template-based successor is the Tpetra package. Trilinos emphasizes abstract interfaces for maximum flexibility of component interchanging, and provides a full-featured set of concrete classes that implement all abstract interfaces. The code is written mainly in C++ with FORTRAN and Python bindings. Most Trilinos source code, including the code developed at Sandia National Laboratories, is licensed either LGPL or BSD.

PARALUTION [7] is a library that enables performing various sparse iterative solvers and preconditioners on multi/many-core CPU, GPU, and Intel Xeon Phi/MIC devices. Based on C++, it provides a generic and flexible design that allows seamless integration with other scientific software packages. PARALUTION contains Krylov subspace solvers, Multigrid, Deflated PCG, Fixed-point iteration schemes, Mixed-precision schemes and fine-grained parallel preconditioners based on splitting, ILU factorization with levels, multi-elimination ILU factorization, additive Schwarz and approximate inverse. The library also provides iterative eigenvalue solvers. The PARALUTION library is released under dual license model – GPL v3 and commercial.

The Vienna Computing Library (**ViennaCL**) [8] is a scientific computing library written in C++ and provides CUDA, OpenCL and OpenMP computing backends. It enables simple, high-level access to the vast computing resources available on parallel architectures such as GPUs and is primarily focused on common linear algebra operations (BLAS levels 1, 2 and 3) and the solution of large systems of equations by means of iterative methods with optional preconditioners. ViennaCL wrappers are available in PETSc. ViennaCL is distributed under the MIT/X11 open source license.

2.2. Open source FEM libraries and frameworks

We introduce here an alphabetically sorted selection of open source libraries implementing discretization of PDEs using Finite Element Method (FEM) aka Finite Elements (FEs). It is perhaps impossible to provide here a fully exhaustive list but we have done our best to point out products that are in our humble opinion well-known, widely used, maintained, and support high-performance computing (HPC). A more complete list of FEM software can be found e.g. in [9].