Available online at www.sciencedirect.com

ScienceDirect

Computers and Structures 85 (2007) 489–498

Computers & Structures

www.elsevier.com/locate/compstruc

Static load balancing applied to Schur complement method

Ondřej Medek [a,*], Jaroslav Kruis [b], Zdeněk Bittnar [b], Pavel Tvrdík [a]

[a] *Department of Computer Science and Engineering, Faculty of Electrical Engineering, 16100 Prague, Czech Republic*
[b] *Department of Structural Mechanics, Faculty of Civil Engineering, Czech Technical University, Prague, Czech Republic*

Received 12 September 2005; accepted 2 August 2006
Available online 25 October 2006

Abstract

A finite element method often leads to large sparse symmetric and positive definite systems of linear equations. We consider parallel solvers based on the Schur complement method on homogeneous parallel machines with distributed memory. A finite element mesh is partitioned by graph partitioning. Such partitioning results in submeshes with similar numbers of elements and, consequently, submatrices of similar sizes. The submatrices are partially factorised. The time spent on the partial factorisation can be different, i.e., disbalanced, because methods exploiting the sparsity of submatrices are used. This paper proposes a Quality Balancing heuristic that modifies classic mesh partitioning so that the partial factorisation times are balanced, which saves overall computation time, especially for time dependent mechanical and nonstationary transport problems.

Keywords: Domain decomposition; Finite element methods; Mesh partitioning; Multilevel graph partitioning; Parallel solvers; Static load balancing; Schur complement method; Time-dependent problems

1. Introduction

Parallel computers have become a popular and widespread tool for solving large scientific and engineering problems. Parallelisation of sequential algorithms may involve considerable changes. Algorithms for solving systems of linear equations are an important example. In this paper, only the finite element (FE) method [1,2], and large sparse symmetric and positive definite linear systems are considered. Parallelisation of classic direct methods for solving linear systems, such as the $\boldsymbol{LDL}^{\mathrm{T}}$ factorisation, is not easy and only partial success has been achieved [3]. The parallelisation of iterative methods, such as the conjugate gradient method, is easier. However, convergence properties are not always optimal. Methods based on at least two level approaches have significantly better properties. Domain decomposition methods are an example of such methods [4–7].

The parallel solvers should be faster than sequential ones. Ideal time reductions cannot be obtained if the processor loads are not balanced. Load balancing describes the fact that all used processors execute an identical or nearly identical number of operations. A slightly disbalanced load of processors is acceptable in problems where the linear system is solved only once. A typical example of such problems is the static linear problem. On the other hand, there are problems such as creep analysis or nonstationary heat transfer where numerous time steps are needed. Each time step involves the solving of a linear system in the case of the implicit method. Every slightly disbalanced load in such cases is significantly amplified and the best possible load balancing is desirable.

In this paper, only the Schur complement method is considered for domain decomposition. Typically, the FE mesh is represented by a graph that is partitioned by graph partitioning. It produces submeshes with similar numbers of elements and nodes. Consequently, submatrices of similar sizes are assembled on each submesh. The Schur complements are computed by partial factorisation from the submatrices

[*] Corresponding author.
E-mail addresses: xmedeko@fel.cvut.cz (O. Medek), jk@cml.fsv.cvut.cz (J. Kruis), bittnar@fsv.cvut.cz (Z. Bittnar), tvrdik@fel.cvut.cz (P. Tvrdík).

0045-7949/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compstruc.2006.08.025

and the reduced problem is usually solved with a suitable iterative method. Common methods for partial factorisation exploit the structure of submatrices, i.e., the number and positions of nonzero matrix entries [8,9]. Therefore, the computational complexity of the partial factorisation depends more on the structure than on the size of a submatrix and classic mesh partitioning may not result in good load balancing. This problem has already been observed [10,11], but has not yet been resolved satisfactorily.

This paper deals with a static load balancing technique for the Schur complement method and homogeneous parallel computers with distributed memory. This technique is called a Quality Balancing (QB) heuristic and its preliminary version appeared in [12]. As the QB heuristic prolongs the time spent on mesh partitioning, its advantages appear in problems where several linear systems with the same structure need to be solved.

This paper is organised as follows: Sections 2 and 3 describe time dependent mechanical and nonstationary transport problems. Sections 4 and 5 explain Schur complement methods for a parallel solution of a linear system. Classic mesh partitioning is described in Section 6 and the QB heuristic is proposed in Section 7. Some illustrative results of solutions to practical problems are presented in Section 8. Finally, Section 9 concludes the paper.

2. A time dependent mechanical problem

For the purposes of this paper, a *time dependent mechanical problem* denotes a problem that depends on time, but the inertial forces are negligible. A typical example of such a problem is creep analysis [1].

Time dependent mechanical problems are usually formulated in the rate form

$$\boldsymbol{K}\dot{\boldsymbol{r}} = \dot{\boldsymbol{f}} + \int_V \boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\dot{\boldsymbol{\varepsilon}}^{ir}\,\mathrm{d}V, \tag{1}$$

where $\boldsymbol{K}$ denotes the stiffness matrix of the problem (domain), $\boldsymbol{r}$ denotes the vector of nodal displacements, $\boldsymbol{f}$ denotes the vector of prescribed nodal forces, $\dot{\boldsymbol{\varepsilon}}_{ir}$ denotes the irreversible strains, $\boldsymbol{D}$ denotes the stiffness matrix of the material, $\boldsymbol{B}$ denotes the strain–displacement matrix, and V denotes the volume of the domain considered. The superimposed dot denotes the time derivative. Eq. (1) is solved by a numerical method that discretises time. The number of time steps is denoted by N_s. The basic steps are summarised in Table 1. The method described in Table 1 is explicit and the particular expression for irreversible strain increments depends on the material model used. The algorithm can be applied to visco-plastic problems as well as creep analysis. The increments of irreversible strains are not specified in more detail in Table 1 because they are not the focus of this paper.

The most time consuming part of the algorithm is the computation of displacement increments that consist of solving a system of linear equations

$$\boldsymbol{K}\Delta\boldsymbol{r}_{i+1} = \Delta\boldsymbol{f}_{i+1} + \Delta\boldsymbol{f}^{ir}_{i+1}. \tag{2}$$

Table 1
An algorithm for time-dependent mechanical problems

For $i = 0$ until $i \leqslant N_s$ compute	
Increments of irreversible strains	$\Delta\boldsymbol{\varepsilon}^{ir}_i$
Increments of internal nodal forces	$\Delta\boldsymbol{f}^{\ ir}_{i+1} = \int_V \boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\Delta\boldsymbol{\varepsilon}^{ir}_i\,\mathrm{d}V$
Increments of external (prescribed) nodal forces	$\Delta\boldsymbol{f}_{i+1} = \boldsymbol{f}(t_{i+1}) - \boldsymbol{f}(t_i)$
Increments of displacements	$\Delta\boldsymbol{r}_{i+1} = \boldsymbol{K}^{-1}(\Delta\boldsymbol{f}_{i+1} + \Delta\boldsymbol{f}^{ir}_{i+1})$
New vector of displacements	$\boldsymbol{r}_{i+1} = \boldsymbol{r}_i + \Delta\boldsymbol{r}_{i+1}$
Total strain increments (previous total strain $\boldsymbol{\varepsilon}_i$ is stored)	$\Delta\boldsymbol{\varepsilon}_{i+1} = \boldsymbol{B}\boldsymbol{r}_{i+1} - \boldsymbol{\varepsilon}_i$
Stress increments	$\Delta\boldsymbol{\sigma}_{i+1} = \boldsymbol{D}(\Delta\boldsymbol{\varepsilon}_{i+1} - \Delta\boldsymbol{\varepsilon}^{ir}_i)$
New stresses	$\boldsymbol{\sigma}_{i+1} = \boldsymbol{\sigma}_i + \Delta\boldsymbol{\sigma}_{i+1}$

3. Nonstationary transport problems

Nonstationary transport problems are also considered in this paper. They are similar to time dependent mechanical problems in the sense that several time steps are used to solve them [2].

Basic facts can be shown on an example of a heat transfer with constant coefficients, which is described by the equation

$$k\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}\right) = \rho c\frac{\partial T}{\partial t}, \tag{3}$$

where T denotes the temperature, k denotes the coefficient of conductivity, ρ denotes the density of the material, and c denotes the thermal capacity. After space discretisation, a system of ordinary differential equations is obtained in the form

$$\boldsymbol{C}\frac{\mathrm{d}\boldsymbol{d}}{\mathrm{d}t} + \boldsymbol{K}\boldsymbol{d} = \boldsymbol{f}, \tag{4}$$

where $\boldsymbol{C}$ denotes the capacity matrix, $\boldsymbol{K}$ denotes the conductivity matrix, $\boldsymbol{d}$ denotes the vector of nodal unknowns, and $\boldsymbol{f}$ denotes the vector of prescribed fluxes. This system of equation (4) is then discretised in time using

$$\boldsymbol{d}_{i+1} = \boldsymbol{d}_i + \Delta t\boldsymbol{v}_{i+\theta}, \tag{5}$$

$$\boldsymbol{v}_{i+\theta} = (1-\theta)\boldsymbol{v}_i + \theta\boldsymbol{v}_{i+1}, \tag{6}$$

where the first time derivative of nodal values is denoted by $\boldsymbol{v}$. Substitution of (5) and (6) into (4) results in the system of linear equations

$$(\boldsymbol{C} + \Delta t\theta\boldsymbol{K})\boldsymbol{v}_{i+1} = \boldsymbol{f}_{i+1} - \boldsymbol{K}(\boldsymbol{d}_i + \Delta t(1-\theta)\boldsymbol{v}_i) \tag{7}$$

with unknown vector $\boldsymbol{v}_{i+1}$. Nodal values are obtained from Eqs. (5) and (6).

4. The Schur complement method

This section summarises only the basic facts about the Schur complement method [4,6,7]. It is based on a special form of the linear system of equations

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \tag{8}$$

that has to be written in the form