Structural Engineering and Mechanics, Vol. 30, No. 4 (2008) 467-480
DOI: http://dx.doi.org/10.12989/sem.2008.30.4.467

Aggregation multigrid method for schur complement system in FE analysis of continuum elements

Jin Hwan Ko†

Department of Aerospace information Engineering, Konkuk University, Seoul, 143-701, Korea

Byung Chai Lee‡

Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, Taejeon, 373-1, Korea

(Received October 15, 2007, Accepted August 21, 2008)

Abstract. An aggregation multigrid method (AMM) is a leading iterative solver in solid mechanics. Recently, AMM is applied for solving Schur Complement system in the FE analysis of shell structures. In this work, an extended application of AMM for solving Schur Complement system in the FE analysis of continuum elements is presented. Further, the performance of the proposed AMM in multiple load cases, which is a challenging problem for an iterative solver, is studied. The proposed method is developed by combining the substructuring and the multigrid methods. The substructuring method avoids factorizing the full-size matrix of an original system and the multigrid method gives near-optimal convergence. This method is demonstrated for the FE analysis of several elastostatic problems. The numerical results show better performance by the proposed method as compared to the preconditioned conjugate gradient method. The smaller computational cost for the iterative procedure of the proposed method gives a good alternative to a direct solver in large systems with multiple load cases.

Keywords: Schur complement system; aggregation multigrid method; FE analysis; continuum elements; multiple load cases.

1. Introduction

This paper investigates the extended application of an aggregation multigrid method (AMM) for solving the linear elasticity system

$$AX=F, \tag{1}$$

where A is a large symmetric positive definite matrix, which is constructed in a FE discretized elliptic PDE, and F is the load matrix in multiple load cases (or multiple right hand sides). The

† Ph.D., Corresponding author, E-mail: jhko@aero.konkuk.ac.kr, jhko@gmail.com
‡ E-mail: bchlee@kaist.ac.kr

dimension of the system is n and the number of the load cases is $nrhs$. The multiple load cases are demanded in order to improve the efficiency of the numerical tests in a stress analysis and mandatory in an eigenproblem as well.

When computing the solution matrix X, both direct and iterative solver can be utilized. A direct solver needs only one factorization even in the multiple load cases, but the computational cost of the factorization dramatically increases as n becomes large. When A is sparse, which is generally true in case of the FE discretization, the state-of-the-art direct solvers such as a mutifrontal solver need the float operation of $O(n^2)$. However, as the matrix becomes denser, the computational cost goes to $O(n^3)$. Hence, a direct solver is extremely expensive for large and dense systems. Meanwhile, an iterative solver uses less memory and does not need the factorization, but it requires an iterative procedure per a load case. Therefore, the multiple load cases have been a challenging problem of an iterative solver due to the cost increase of the iterative procedure.

In order to avoid factorizing a full-size matrix, the substructuring method splits the original system into numerous subsystems of smaller size, and then constructs Schur Complement (SC) system after a static condensation. The subsystems are solved quickly by the direct solver because of the smaller size, but SC system becomes so dense that a direct solver still becomes expensive. Hence, SC system has been generally solved by an iterative solver. One of famous iterative solvers is the Krylov subspace method of which performance is known to be mainly dependent on a preconditioner. Saad and Sosonkina (1999) employed and compared various preconditioners based on inverse matrix approximation, incomplete LU factorization, or approximate block LU factorization for each subdomain, and demonstrated their robust performances. However, a better efficient preconditioner for the iterative solver is still demanded in order to obtain its competitiveness in the multiple load cases compared to a direct solver.

AMM which is a high performance iterative solver was introduced in the mid-1990s. AMM became popular due to a near-optimal convergence of a multigrid method, which means little effects on the convergence when varying the element size of the FE discretization. Bulgakov (1995), Fish and Belsky (1997) have developed the iterative solver based on AMM, which provides wide applications in the solid mechanics Bulgakov (1997). Recently, the use of AMM for Schur complement system is introduced in the FE analysis of shell structures Ko and Lee (2006). In this paper, we propose the extended application of AMM for the FE analysis of continuum elements and also investigate the performance of the proposed method in the multiple load cases.

This paper is organized as follows. Section 2 reviews the formulation of the Schur complement system and Section 3 describes the AMM that is applied to the Schur complement system. Finally, in Section 4, numerical experiments and their results are presented for the FE models of continuum elements in solid mechanics.

2. Schur formulation

The substructuring method divides an original domain into numerous subdomains, which are decoupled and connected with others by interfacial variables only. Hence the matrix components representing the coupling terms among the inner variables of the subdomains become zero. By partitioning into N subdomains, the original system of Eq. (1) is rearranged as the following structure

$$\begin{Bmatrix} B_1 & & & & E_1 \\ & B_2 & & & E_2 \\ & & \ddots & & \vdots \\ & & & B_N & E_N \\ E_1^T & E_2^T & \cdots & E_N^T & C \end{Bmatrix} \begin{Bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \\ Y \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ \vdots \\ F_N \\ G \end{Bmatrix} \tag{2}$$

where each B_i represents the relations among the inner variables X_i, E_i denotes the coupling among the interfacial variables and the inner variables of a subdomain and C stands for the relations among the interfacial variables. By so-called static condensation, the Schur Complement (SC) and the right hand side matrices associated with the interfacial variables are given as

$$C_{SC} = \sum_{i=1}^{N} (C_i - E_i^T U_i^{-1} L_i^{-1} E_i), \tag{3}$$

$$G_{SC} = G - \sum_{i=1}^{N} E_i^T U_i^{-1} L_i^{-1} F_i, \tag{4}$$

in which $B_i = L_i U_i$, where L_i, U_i denote the factorized lower and upper triangular matrices for i-th subdomain and C_i is the contribution matrix of i-th subdomain to C. Using Eqs. (3)-(4), the original system is condensed to the SC system

$$C_{SC} Y = G_{SC}. \tag{5}$$

By factorizing C_{SC} into $L_{SC} U_{SC}$, then $Y = U_{SC}^{-1} L_{SC}^{-1} G_{SC}$. Once Y is obtained, the solution X of the original system is recovered by the following procedure

$$X_i = U_i^{-1} L_i^{-1} (-E_i Y + F_i), \qquad \text{where} \quad i = 1,.....N. \tag{6}$$

The size of the SC matrix has reduced to the number of the interfacial variables, but the matrix becomes dense. In addition, the SC matrix preserves the SPD of the original system Saad (1996). When solving the interfacial system, an iterative solver is more preferred than a direct solver if the number of the interfacial variables becomes large. The Krylov subspace method is frequently used for solving the SC system and it requires a preconditioner to accelerate its convergence. One-level preconditioners for the SC matrix have been introduced by many researchers. Among them, we adopt an incomplete factorization for each subdomain Saad and Sosonkina (1999), Saad (1996) as the one-level preconditioner. To construct it, SC is first approximated by the static condensation after incomplete factorizing B_i matrices as

$$\tilde{C}_{SC} = \sum_{i=1}^{N} (C_i - E_i^T \tilde{U}_i^{-1} \tilde{L}_i^{-1} E_i), \tag{7}$$

in which $B_i \approx \tilde{L}_i \tilde{U}_i$, where $\tilde{L}_i, \tilde{U}_i$ are the low and upper triangular matrices after the incomplete factorization (*IC*), and $\tilde{C}_{SC}$ is the approximate SC matrix. Afterwards, the preconditioner M_{SC} is composed of the incomplete factorized matrices as

$$M_{SC} = IC(\tilde{C}_{SC}) = \tilde{L}_{SC}\tilde{U}_{SC}. \tag{8}$$

For second order elasticity problems such as the FE analysis of continuum elements, the condition number grows asymptotically as $O(h^{-2})$ where h is the maximum diameter of the discretized elements. In the case of the SC system, Brenner (1999) defined the upper and lower bounds of the condition number in m-th order Sobolev space by

$$\alpha H^{-1}h^{-2m+1} \le \kappa(C_{SC}) \le \beta H^{-1}h^{-2m+1}, \tag{9}$$

where H is the maximum diameter of the subdomains and α, β are constants independent of H and h. The condition number of the SC system grows as $O(h^{-1}H^{-1})$ because the displacements of the FE models of continuum elements are often approximated on the first order Sobolev space.

3. Multigrid method

A multigrid method is motivated by the fact inexpensive iterative solvers, such as Gauss-Seidel, are effective at reducing the high frequency error. These solvers are called *smoothers* because they render the error geometrically smooth by reducing the high frequency content. However, smoothers are ineffectual at reducing the low frequency error. The low frequency error can be reduced effectively with solving the coarse version of the problem, which a main procedure of the multigrid method and called *coarse grid correction*. Here, a two-level multigrid method is described as follows

Algorithm 1 *Basic two-level multigrid*
1: *pre-smoother*: $X \leftarrow X - D^{-1}(AX - F)$
2: *coarse grid correction : find V from the coarse problem*
$$P^T A P V = P^T(AX - F)$$
3: *prolongate and correct the solution by* $X \leftarrow X - PV$
4: *post-smoother* : $X \leftarrow X - D^{-1}(AX - F)$

where the prolongator P transfers from R^m to R^n ($m < n$). The space R^m is called the coarse grid space, R^n is the fine grid space, and $P^T A P$ is the coarse grid operator. D is a preconditioner for the smoother, which can be a diagonal matrix or incomplete factorized matrices. The multigrid methods are used as stand-alone solvers or preconditioners for an iterative solver such as the Krylov subspace method.

Algebraic multigrid methods are methods that construct the coarse grid operator algebraically. More specifically, they refer to the multigrid method that constructs the coarse grid operators from a stiffness matrix alone, with little or no extra data required from the application. Such method is the aggregation multigrid (AMM) which utilizes the nodal coordinate as the extra data and we employ AMM because AMM shows best performance in solid mechanics Bulgakov (1995; 1997).

3.1 Aggregation multigrid method (AMM)

In solid mechanics, the AMM utilized the rigid body modes, which are easily constructed from the nodal coordinates. Several rigid body mode methods have been developed Bulgakov (1995),

Fish and Belsky (1997). They have shown near-optimal convergence of a multigrid method, which means that the convergence is barely affected by the variation of the element size.

The aggregation is a kind of fine/coarse grid splitting, but has a simple relationship between the fine and coarse grids in order to reduce the computations in constructing the coarse grid operator. Aggregates refer to the sets of the nodes connected to each other and play the part of the nodes in the coarse grid space.

3.1.1 Construction of aggregates

When applying a typical aggregation to the SC system, we need to aggregate each subdomain separately in order that the coarse grid spaces preserve the relationship among the subdomains as same as the fine grid space. This aggregation is referred to as *SC-aggregation*. Coarse/fine grids are selected by *strong connectivity*, which is denoted in Vanek *et al.* (1996), based on the stiffness matrix of l-th level, A^l. Hence a neighborhood set $\mathbb{N}_s^l(\varepsilon)$ with the strong connectivity of l-th grid is defined as

$$\mathbb{N}_s^l(\varepsilon) = \{s : |a_{st}| \geq \varepsilon \sqrt{a_{ss} a_{tt}}\}, \tag{10}$$

which is subjected to the condition that t must belong to the same inner part of the subdomain or the interface including s. In Eq. (10), ε is a dropping limit to exclude weakly connected components and is called *aggregation tolerance* Ko and Lee (2006). When the aggregation tolerance is zero, each aggregate has the maximum number of the nodes and the number of the aggregates then becomes miminum.

3.1.2 Intergrid operator

After the nodes belonging to each aggregate are determined, the intergrid operator, that is, the prolongators and restrictors are constructed based on the rigid body modes Fish and Belsky (1997). The rigid body modes Φ_{st} are defined for t-th aggregate of the coarse grid and s-th node of the fine grid and expressed by X_T corresponding to three translations and X_R corresponding to three rotations. Note that each aggregate in the coarse grid spaces will have six degrees of freedom even if each node in the finest grid space does not have six degrees of freedom. The FE model by the continuum elements has three translation degrees of freedom per node therefore the intergrid relation by the rigid body mode can be expressed as

$$X_s = \Phi_{st} X_t \Leftrightarrow (X_T)_s = [I_3 \quad \Delta] \begin{Bmatrix} (X_T)_t \\ (X_R)_t \end{Bmatrix}, \tag{11}$$

where I_n is an identity matrix with n size, and the block Δ represents the contribution of the rotational variables of s-th fine grid to the translations of t-th coarse grid. The block is defined as

$$\Delta = \begin{bmatrix} 0 & \Delta_3 & -\Delta_2 \\ -\Delta_3 & 0 & \Delta_1 \\ \Delta_2 & -\Delta_1 & 0 \end{bmatrix}, \tag{12}$$

in which $\Delta_k = q_k^s - q_k^t$, where q_k is the k-th coordinate of the corresponding node; the superscript denotes the index of the nodes (or the aggregates) and the subscript stands for the index of degrees of freedom of the node. If $(X_R)_s$ is existed, the intergrid relation of the rotational degrees of freedoms by the rigid body modes can be expressed by $(X_R)_s = I_3 (X_R)_t$.

When constructing the intergrid operator for the model of the continuum elements, difficulties are on the different number of d.o.f. per a grid between the coarse and fine spaces. The coarse grid operators constructed by Φ_{st} of Eq. (11) may be singular. The difficulty is resolved when Q matrices after QR factorization of the rigid body modes are adopted instead. By the modified intergrid operator, the coarse grid operators become nonsingular and positive definite Adams (2002). The smoothed aggregation Vanek, Mandel and Brezina (1996) is not included for a simple implementation. The prolongators in the *nl*-level case are constructed via the following algorithm.

Algorithm 2 *Construction of prolongate operators and coarse grid matrices*
1: *For each aggregate t,* $\Phi_t^l \rightarrow \Upsilon_t^l$ at $l = nl-1$ where, $\Phi_t^l = \sum \Phi_{st}^l, s \in \mathbb{C}_t^l$
2: *if* $l \neq nl-1$, *construct* $\overline{P}_t^l \Phi_t^l \rightarrow \Upsilon_t^l$
3: *QR factorize* $\Upsilon_t^l \rightarrow P_t^l \overline{P}_t^{l-1}$
4: P^l *is a block diagonal matrix with* P_t^l *s in the t-th diagonal block*
5: Construct the coarse grid operator of *l*-th grid, $(P^l)^T A^l P^l \rightarrow A^{l-1}$
6: $l \rightarrow l-1$, *if* $l \neq 0$, *go to procedure* 2

where $\overline{P}_t$ is the initial prolongator of *t-th* aggregate, and Υ_t is an intermediate prolongation operator. In procedure 3, P_t corresponds to the Q matrix after QR factorization is used as the actual prolongator at the present grid level, and $\overline{P}_t$ corresponds to the R matrix after QR factorization is used as the initial prolongator at the next grid level.

For the system of Eq. (2), the prolongator is composed of the local prolongators correspond to the inner parts of the subdomains and the interfaces, which are separately constructed according to the SC-aggregation. Thus the prolongator becomes the following block rectangle matrix

$$P = \begin{bmatrix} P_1 & & & 0 \\ & \ddots & & \\ & & P_N & \\ 0 & & & P_{N+1} \end{bmatrix}. \tag{13}$$

This is also separated into the subdomain form of

$$P_i = \begin{bmatrix} P_i & 0 \\ 0 & (P_{N+1})_i \end{bmatrix}, \quad P_{N+1} = (P_{N+1})_i + \cdots + (P_{N+1})_N, \tag{14}$$

where P_i is the prolongator of the inner part of *i*-th subdomain, $(P_{N+1})_i$ s are the prolongators of the interfacial part of *i*-th subdomain generated by P_{N+1}, which becomes the prolongator of the interfacial variables.

3.1.3 Coarse grid operator

The SC coarse grid operator can be constructed by the two approaches: (i) the condensation after the coarsening each subdomain and (ii) the coarsening after the condensation. The latter approach based on the algebraic multigrid method for the SC system was developed by Carvalho, Giraud, Le Tallec (2001). However, the latter approach is not suitable for the AMM, because the aggregation utilizes the nodal coordinates to represent the intergrid rigid body modes; thus the former approach is adopted.

For the simplicity of exposition, we consider the case of the two grid case where superscript *0* means the coarse grid space and no superscript means the fine grid space. Based on the former approach, the SC coarse grid operator is constructed by the following steps. First, the original

system matrix of each subdomain is assembled and transformed into the coarsened system by using Eq. (14)

$$A_i^0 = \begin{bmatrix} B_i^0 & E_i^0 \\ (E^T)_i^0 & C_i^0 \end{bmatrix} = \begin{bmatrix} (P_i)^T & 0 \\ 0 & (P_{N+1})_i^T \end{bmatrix} \begin{bmatrix} B_i & E_i \\ E_i^T & C_i \end{bmatrix} \begin{bmatrix} P_i & 0 \\ 0 & (P_{N+1})_i \end{bmatrix}$$

$$= \begin{bmatrix} (P_i)^T B_i P_i & (P_i)^T E_i (P_{N+1})_i \\ (P_{N+1})_i^T E_i^T P_i & (P_{N+1})_i^T C_i (P_{N+1})_i \end{bmatrix}. \tag{15}$$

Then, the SC coarse grid operator is constructed by the condensation in the same manner as the finest grid space because the SC-aggregation prevents undesirable coupling in the coarse grid space:

$$C_{SC}^0 = \sum_{i=1}^{N} C_i^0 - (E_i^0)^T (U_i^0)^{-1} (L_i^0)^{-1} E_i^0, \tag{16}$$

where $B_i^0 = L_i^0 U_i^0$. In *nl*-grid case, the processes similar as Eq. (15) are executed in sequence from the fine grid to the coarsest grid and the coarsest grid operator can be constructed in the similar form of Eq. (16).

3.1.4 Preconditioner for SC matrix

As we mentioned, the multigrid method is used as the preconditioner of the Krylov subspace method and many researchers introduces their own techniques. Among them, we adopt the technique which Bulgakov proposed and is called *multilevel aggregation method* (MLA) Bulgakov (1995, 1997). MLA employs only the post-smoother of Algorithm 1. Here, all procedures are described in the two-grid case for the simplicity of exposition. In order to construct the preconditioner for the Krylov subspace method, an auxiliary equation is defined as

$$M_{SC} V = R, \tag{17}$$

where R is the residual of the SC system and V is the approximate error of the fine grid space.

First, the residual of the SC system is calculated by

$$R = C_{SC} Y - G_{SC}. \tag{18}$$

Then, the right-hand side vector R is restricted to the coarse grid space as

$$R^0 = (P_{N+1})^T R. \tag{19}$$

Next, the correction vector V^0 is obtained by solving the system of the coarse grid space

$$V^0 = (C_{SC}^0)^{-1} R^0 = (U_{SC}^0)^{-1} (L_{SC}^0)^{-1} R^0. \tag{20}$$

Then, the post-smoother is applied to approximate the fine grid residual as

$$V_{j+1} = V_j - \tilde{U}_{SC}^{-1} \tilde{L}_{SC}^{-1} (C_{SC} V_j - R) \tag{21}$$

where $V_0 = P_{N+1}V^0$ and j is the number of the smoothing iterations. When the incomplete factorization is utilized as the preconditioner of the smoother, quicker convergence has been observed Bulgakov (1995), Fish and Belsky (1997). Hence the incomplete factored matrices of the approximate SC in Eq. (8) are used in Eq. (21). The number of the smoothing iterations has been recommended to be between 1 and 3.

Let us set $j = 0$ from Eq. (21), we have

$$V_1 = (I - \tilde{U}_{SC}^{-1}\tilde{L}_{SC}^{-1}C_{SC})P_{N+1}V^0 + \tilde{U}_{SC}^{-1}\tilde{L}_{SC}^{-1}R\,. \tag{22}$$

By substituting Eqs. (19) and (20) into Eq. (22), the approximate error is computed by

$$V = V_1 = (I - \tilde{U}_{SC}^{-1}\tilde{L}_{SC}^{-1}C_{SC})P_{N+1}(C_{SC}^0)^{-1}(P_{N+1})^T R + \tilde{U}_{SC}^{-1}\tilde{L}_{SC}^{-1}R\,. \tag{23}$$

Finally, M_S^{-1} in the two-grid case is obtained by using Eq. (23) and Eq. (17):

$$M_{SC}^{-1} = (I - \tilde{U}_{SC}^{-1}\tilde{L}_{SC}^{-1}C_{SC})P_{N+1}(C_{SC}^0)^{-1}(P_{N+1})^T + \tilde{U}_{SC}^{-1}\tilde{L}_{SC}^{-1}\,. \tag{24}$$

The inverse matrices in the right hand side are not actually calculated, but instead the factorized matrices are utilized. The preconditioner in Eq. (24) is not symmetric, although all SC matrices are symmetric. Hence the preconditioner cannot be used in the preconditioned conjugate gradient (PCG). Instead, the preconditioned conjugate gradient squared (PCGS) is employed for non-symmetric cases Sonneveld (1989) and it is implemented in the form similar to that described in Ko and Lee (2006).

In nl multi-level case, the Eq. (19) is changed into

$$R^0 = \prod_{l=nl-1}^{1} (P_{N+1}^l)^T R^{nl-1} \tag{25}$$

which presents the multilevel coarsening. The Eq. (22) is also changed into

$$V_1^l = [I - (\tilde{U}_{SC}^l)^{-1}(\tilde{L}_{SC}^l)^{-1}\tilde{C}_{SC}^l]P_{N+1}^l V^{l-1} + (\tilde{U}_{SC}^l)^{-1}(\tilde{L}_{SC}^l)^{-1}R^l, \quad l = 1, ..., nl-1 \tag{26}$$

The inverse of the multi-level preconditioner is not constructed explicitly, but instead the serial matrix-vector multiplications are utilized.

4. Numerical experiments and discussion

4.1 Solution methods

We employ the mutifrontal solver Ashcraft and Grimes (1999) for the SC system (SC-MFS) and PCG for the SC system (SC-PCG) in order to compare to MLA for the SC system (SC-MLA). We also adopt the corrected incomplete Cholesky, which is one of robust incomplete factorizations Saint-Georges *et al.* (1999). Original domain is partitioned by METIS Karypis, which is a public code and has the function minimizing edge-cut.

The proposed method, SC-MLA needs a number of parameters. The influence of some parameters has been investigated in the previous research Ko (2004) hence we determine the values based on the results of the research. First, the fill-in terms of the incomplete factorization are determined by the drop tolerance ϕ, which is given by 10^{-3} for SC-PCG and by 10^{-2} for SC-MLA.

$$\xi_{ij}^2 > \phi d_{ii} d_{jj} \tag{27}$$

where ξ_{ij} is the off-diagonal term located in (i, j) and d_{ii} is i-th diagonal term during the incomplete factorization. Next, the number of the iterations in the smoothers for SC-MLA is fixed to 1. The level of the aggregation is recommended as 3 when the domain size is fixed to be nearly 1000 nodes for each subdomain. The aggregation tolerance is determined in order that the neighborhood set has half entries of $\mathbb{N}_S^l(0)$. Finally, the convergence tolerance in the iterative solvers is set to 10^{-6}, which guarantees the similar level of the accuracy of the solutions. A practical convergence criterion of k-th iteration is

$$\|r_k\| / \|r_0\| < tol \tag{28}$$

For improving the efficiency during the iterative procedure, equivalent multiplication procedures are implemented instead of $C_{SC}v$ Saad, Y. (1996): (i) Compute $v' = Ev$ (ii) Solve $Bz = v'$ (iii) Compute $w = Cv - E^T z$. By the three steps, SC-MLA and SC-PCG do not need to construct the SC matrix in the explicit form of Eq. (3). Hence the matrix-matrix multiplication during the static condensation in Eq. (3) $E_i^T U_i^{-1} L_i^{-1} E_i$ is not required.

4.2 Cost measures

First, the number of non-zero terms (*nnz*) is selected as the memory usage measure. Once the SC system is constructed, *nnz* of the factorized matrices of the SC matrix in the explicit form is used for SC-MFS, which is a direct solver. For the iterative solvers such as SC-MLA and SC-PCG, the total of *nnz* of the SC matrix in the implicit form and the preconditioner is counted:

$$nnz \text{ (SC-MFS)} = nnz(L_{SC}),$$

$$nnz \text{ (SC-PCG)} = \sum_{i=1}^{N} \{nnz(L_i) + nnz(E_i)\} + nnz(C) + nnz(\tilde{L}_{SC}),$$

$$nnz \text{ (SC-MLA)} = \sum_{l=1}^{nl-1} \left[\sum_{l=1}^{N} \{nnz(L_i^l) + nnz(E_i^l)\} + nnz(C^l) + nnz(\tilde{L}_{SC}^l) + nnz(P^l) \right] + nnz(L_{SC}^0).$$

Due to the symmetry of the matrices, each upper triangular matrix U is uncounted.

Next, the number of float operations (*ops*) as the computational cost measure. *ops* of SC-MFS is mainly composed of the construction of the explicit SC matrix (ops_{ec}), the factorization (ops^f), the back and forward substitutions (ops_b) and the recovery procedure (ops_r) as

$$ops_{SC-MFS} = ops_{ec} + ops_f + ops_b + ops_r.$$

Each iterative solver method is made up of the construction of the implicit SC matrix (ops_{ic}), the

construction of the preconditioner (ops_p), the iteration procedure (ops_i), and the recovery procedure (ops_r), that is,

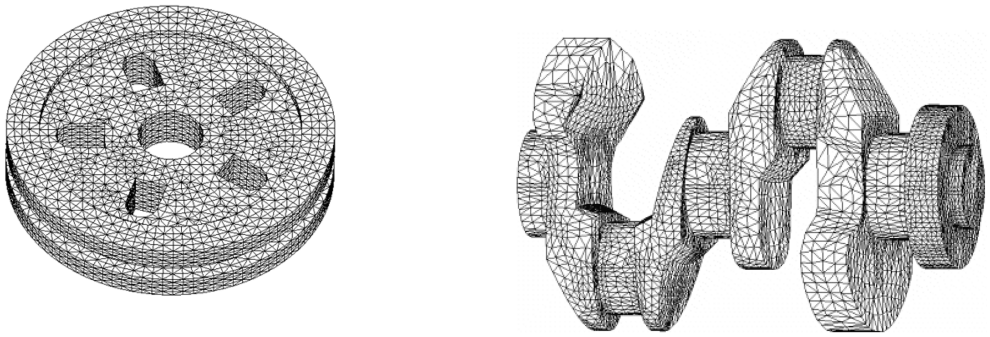
$$ops_{SC-MLA,SC-PCG} = ops_{ic} + ops_p + ops_i + ops_r .$$

Here, ops_r is the same in all solvers and typically occupies the small portion of the total *ops*.

4.3 Numerical examples and their SC system

The original FE models of pulley and crankshaft, which is simply called shaft, are in Fig. 1. For the material properties, Yong's modulus is given by 210 Gpa and Poisson ratio is 0.3. The first model is the pulley which transfers rotational forces from an automobile engine. The clamped boundary conditions are in the the inner circle surface and forces are given at some attachment area by a belt. The second model is the crankshaft which changes the translational motion to the rotational motion in an automobile engine. The symmetry boundary conditions along the horizontal axis are given at the left end and the clamped boundary conditions are given at the right end. Forces on the area where a connecting rod attaches are given in the vertical direction. The finer FE models of the pulley and the crankshaft are also employed and denoted by f-pulley and f-shaft.

After partitioning all examples by 1000 nodes for each subdomain, Table 1 shows the changed properties from the original matrices to the SC matrices. As described in section 2.1, the d.o.f of the SC matrix is reduced to 1/3~1/4 and its density, which is $(100 \times nnz)/(\text{d.o.f})^2$, increases by around 100 times.



(a) (b)

Fig. 1 The FE models of (a) the pulley connected with a belt, and (b) the crankshaft of an engine

Table 1 Information of the original matrices and SC matrices after partitioning all examples (*nnz* represents the entries of the lower triangular matrices)

Examples	Original matrix *A*			Schur Complement matrix (the explicit form)			
	d.o.f.	nnz	Density (%)	N	d.o.f.	nnz	Density (%)
pulley	42,054	7.8e05	0.08	14	10,497	3.0e06	5
shaft	44,892	9.3e05	0.09	15	10,127	3.2e06	6
f-pulley	271,584	5.7e06	0.01	90	91,389	4.3e07	1
f-shaft	321,555	7.1e06	0.01	107	108,350	5.1e07	1

4.4 Performance comparison

4.4.1 SC construction and preconditioner

The SC matrix in the explicit form should be constructed in SC-MFS. Though the iterative solvers can use the implicit forms, they require their preconditioners; thus, we added the cost for constructing the preconditioner to the construction of SC matrix for the iterative solvers. We tabulated the *ops* and *nnz* of each method for those procedures in Table 2.

It is indicated from the table that the *ops* and *nnz* of SC-PCG and SC-MLA, even though they include the cost for constructing the preconditioner, are smaller than those of SC-MFS. Hence, in the preliminary stage, the iterative solvers have advantage in view of the cost than the direct approach. Between the two iterative solvers, SC-MLA in which the larger drop tolerance is used results in smaller *ops* than SC-PCG. However, SC-MLA additionally needs the SC matrices and the approximate SC matrices of the coarse grid spaces; thus, SC-MLA requires larger *nnz* than SC-PCG.

4.4.2 Comparison between two iterative solvers

When we experiments the two finer models, we chose the same H/h ratio (1000 nodes per a subdomain), which means the decrease of both of h and H. Subsequently, the condition number increases according to the order $O(H^{-1}\ h^{-1})$. In order to investigate the performance of the two iterative methods, we represented the number of iterations, *ops* and elapsed times of the iterative procedures in Table 3.

As seen in the table, the number of iterations increases by over 2.6 times in SC-PCG, meanwhile less 1.3 times in SC-MLA when they are applied to each finer example. Due to the better convergence of SC-MLA, *ops* and the elapsed time of it are much smaller than those of SC-PCG, specially, 1/10 for the f-shaft. Subsequently, the ratio of the iteration cost to the total cost for SC-MLA is less than that of SC-PCG.

The maximum MFLOPS of PCGS in SC-MLA is 90 and that of PCG in SC-PCG is 105 by Intel Pentium 4 CPU 3.00 GHz when they are complied by Intel C++ complier Ver. 10. The both

Table 2 *ops* and *nnz* when constructing the SC matrices and corresponding preconditioners

Examples	SC-MFS		SC-PCG		SC-MLA	
	ops	*nnz*	*ops*	*nnz*	*ops*	*nnz*
pulley	1.67e09	2.99e06	5.33e08	2.00e05	5.27e08	4.99e05
shaft	2.58e09	3.28e06	8.53e08	3.26e05	7.10e08	5.80e05
f-pulley	1.37e10	4.30e07	4.45e09	1.08e06	4.64e09	3.77e06
f-shaft	2.35e10	5.17e07	7.00e09	1.45e06	6.44e09	3.94e06

Table 3 The number of iterations, ops_i (*ratio to the total ops*), elapsed times of SC-MLA and SC-PCG (ops_i : *ops* in the iterative procedure)

Examples	SC-PCG			SC-MLA		
	# of iter.	ops_i *(ratio %)*	*Time(sec.)*	*# of iter.*	ops_i *(ratio %)*	*Time(sec.)*
pulley	270	1.03e09 (66)	20.1	14	3.01e08 (36)	4.7
shaft	362	2.58e09 (75)	24.4	17	5.18e08 (42)	5.7
f-pulley	707	2.22e10 (83)	356.5	17	2.39e09 (38)	33.1
f-shaft	1124	5.61e10 (89)	651.5	21	4.70e09 (42)	64.0

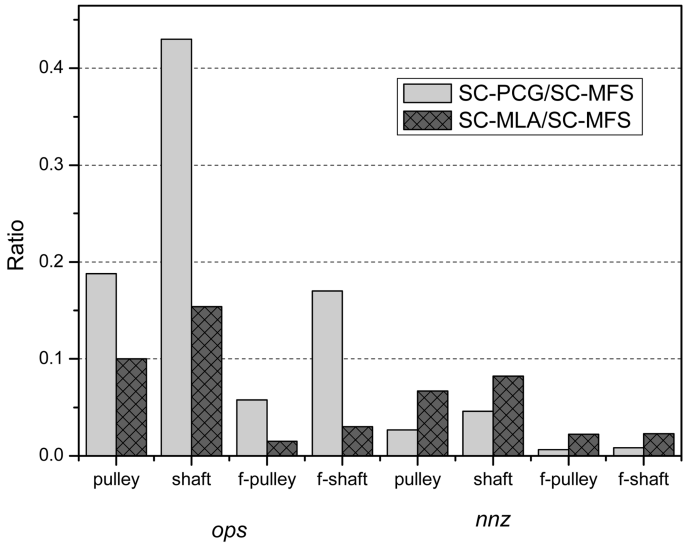


Fig. 2 The *ops* and *nnz* ratio of the examples discretized by the continuum elements

MFLOPS are not enough speed compared to 531 MFLOPS for factoring the SC matrix in SC-MFS because the two iterative routines do not have a high performance BLAS. Thus the software optimization in view of the speed will be required. For this reason, we plotted only *ops* and *nnz* in the next comparison.

4.4.3 Comparison of two iterative solvers to SC-MFS

SC-MFS should factor the SC matrix in the explicit form, which causes large amount of *nnz* and *ops*. For a single load case, the costs of the two iterative methods compared to SC-MFS are depicted in Fig. 2.

According to it, SC-MLA requires more memory usage than SC-PCG, but memory usage under 10% compared to SC-MFS. The smaller amount of the memory usage compared to a direct solver has been one of major advantages of an iterative solver. The results of Fig. 2 also show that SC-MLA requires less computational cost than SC-MFS and SC-PCG. Specially, *ops* of SC-MLA are under 16% compared to that of SC-MFS for all examples. This result is caused by the fact that the number of the iteration is considerably small compared with SC-PCG according to the results of Table 3 and the factorization is conducted on much smaller size subsystems and the SC matrix of the coarse grid space in SC-MLA. It is noticeable that the ratios of *ops* and *nnz* for the both iterative solvers decrease as the size of the problems increases. Therefore, SC-MLA is expected to spend less cost than SC-MFS in larger systems.

4.4.4 Multiple load cases

The multiple load cases have been a challenging problem for an iterative solver because the iterative procedure is required per a load case. Meanwhile, one back and forward substitution per a load case is demanded for a direct solver, but the cost of it is considerable smaller than that of the factorization. In addition, the construction of the right hand side of the SC system in Eq. (4) and the recovery procedure in Eq. (6) are needed per a load case for every method.

Table 4 Additional cost and total cost for 50 duplicated load cases

Examples	SC-MFS		SC-PCG		SC-MLA	
	Add. ops	*50 nrhs*	*Add .ops*	*50 nrhs*	*Add. ops*	*50 nrhs*
pulley	2.27e07	9.49e09	1.04e09	5.37e10	6.38e08	1.63e10
shaft	2.07e07	9.04e09	2.59e09	1.33e11	8.36e08	2.75e10
f-pulley	3.64e08	4.79e11	2.23e10	1.14e12	5.83e09	1.28e11
f-shaft	3.86e08	3.91e11	5.62e10	2.87e12	6.44e09	2.48e11

The convergence rate of SC-MLA when applied to different load cases has been investigated in the previous research Ko (2004), which says that the difference of the convergence rate is not considerable. Hence, we chose duplicated load cases as a typical case of the multiple load cases. We summarized the additional cost and the total cost for 50 duplicated load cases in Table 4.

It is indicated from the table that the additional cost of SC-MLA are larger than that of SC-MFS, but smaller than that of SC-PCG. Subsequently, SC-PCG requires much more *ops* than SC-MFS in case of 50 *nrhs* for all problems. Meanwhile, SC-MLA needs more cost for the first two models, but smaller *ops* than SC-MFS for two finer models in the same case. Therefore, SC-MLA can be a competitive iterative solver in case of 50 *nrhs* for the two finer examples considering the much less amounts of the memory usage compared to SC-MFS.

5. Conclusions

In this paper, we presented the extended application of the aggregation multigrid method for solving the Schur Complement system (SC-MLA) in the FE analysis of the continuum elements. Also, we have investigated the performance of SC-MLA for a certain number of the duplicated load cases. SC-MLA showed competitive performance compared to the direct solver, SC-MFS for the finer examples under the given number of the loads. Namely, SC-MLA only needed less than 65% float operations and 3% non-zero terms of SC-MFS for the finer examples with 50 *nrhs*. The numerical results of Fig. 2 show that the performance of SC-MLA is expected to be better in larger systems. Therefore, SC-MLA can be a good alternative of a direct solver for considerably large FE systems under the multiple load cases. Future work includes the parallelization and the applications to dynamic problems such as an eigenproblem or a frequency response problem.

Acknowledgements

This work is supported by the National Research Laboratory program, Korea Science and Engineering Foundation Grant (R0A-2007-000-20012-0).

References

Adams, M. (2002), "Evaluation of three unstructured multigrid methods on 3D finite element problems in solid mechanics", *Int. J. Numer. Meth. Eng.*, **55**, 519-534.

Ashcraft, C. and Grimes, R. (1999). "SPOOLES: An object-oriented sparse matrix library", *9th SIAM Conference on Parallel Processing for Scientific Computing*, San Antonio, Texas. http://citeseer.ist.psu.edu/ashcraft99spooles.html

Brenner, S.C. (1999), "The condition number of the Schur complement in domain decomposition", *Numer. Math.*, **83**, 187-203.

Bulgakov, V.E. (1995), "High performance of multi-level iterative aggregation solver for large finite-element structural analysis problems", *Int. J. Numer. Meth. Eng.*, **38**, 3529-3544.

Bulgakov, V.E (1997), "The use of the multi-level iterative aggregation method in 3-d finite element analysis of solid, truss, frame and shell structure", *Comp. Struct.*, **63**(5), 927-938.

Carvalho, L.M., Giraud, L. and Le Tallec, P. (2001), "Algebraic two-level preconditioners for the Schur Complement method", *SIAM J. Sci. Comput.*, **22**(6), 1987-2005.

Fish, J. and Belsky, V. (1997), "Generalized aggregation multi-level solver", *Int. J. Numer. Meth. Eng.*, **40**, 4341-4361.

Karypis, G. and Kumar, V. (1995), METIS: unstructured graph partitioning and sparse matrix ordering system, Tech. rep., Department of Computer Science, University of Minnesota, available on the WWW at URL http://www-users.cs.umn.edu/~karypis/metis/metis/index.html

Ko. J.H. and Lee, B.C. (2006), "Preconditioning Schur complement matrices based on an aggregation multigrid method for shell structures", *Comp. Struct.*, **84**(29-30), 1853-1865.

Ko, J.H. (2004), "A preconditioner for schur complement matrices based on aggregation multigrid method in linear finite element analysis," Ph. D. dissertation, KAIST.

Saad, Y. and Sosonkina, M. (1999), "Distributed schur complement technique for general sparse linear systems", *SIAM J. Sci. Comput.*, **21**(4), 1337-1356.

Saad, Y. (1996), *Iterative Methods for Sparse Linear System*, New York: PWS publishing.

Saint-Georges, P., Warzee, G., Notay, Y. and Beauwens, R. (1999), "Problem-dependent preconditioners for iterative solvers in FE elastostatics", *Comp. Struct.*, **73**, 33-42.

Sonneveld, P. (1989), "CGS, a fast Lanczos-type solver for nonsymmetric linear system", *SIAM J. Sci. Statistical Comput.*, **10**(1), 36-52.

Vanek, P., Mandel, J. and Brezina, M. (1996), "Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems", *Computing*, **56**, 179-196.