

# Efficiency Study of the “Black-Box” Component Decomposition Preconditioning for Discrete Stress Analysis Problems

M.D. Mihajlović<sup>1\*</sup> and S. Mijalković<sup>2</sup>

<sup>1</sup> Department of Computer Science, The University of Manchester, Manchester, UK

<sup>2</sup> Faculty of Electrical Engineering, Mathematics and Computer Science,  
Delft University of Technology, Delft, The Netherlands

**Abstract.** Efficiency of the preconditioning methodology for an iterative solution of discrete stress analysis problems is studied in this article. The preconditioning strategy is based on space decomposition and subspace correction framework. The principle idea is to decompose a global discrete system into a sequence of scalar subproblems, corresponding to the different Cartesian coordinates of the displacement vector. The scalar subproblems can be treated by a host of direct and iterative techniques, however we restrict ourselves to a “black-box” application of the direct sparse solvers and the scalar algebraic multigrid (AMG) method. The subspace correction is performed in either block diagonal or block lower triangular fashion. The efficiency and potential limitations of the proposed preconditioning methodology are studied on stress analysis for 2D and 3D model problems from microfabrication technology.

## 1 Introduction

Stress analysis is an essential part of modelling and design of complex physical and mechanical systems that consist of a variety of structural elements with different mechanical properties and intrinsic built-in stress distributions. The underlying multi-layer material geometries (with potentially very complex shapes) are usually partitioned by general unstructured grids. The governing stress equations are discretised by the finite element method. This procedure leads to the solution of large sparse linear algebraic systems with a non-regular sparsity pattern and poor equations scaling.

The problem of designing efficient solution methods for stress analysis has been studied by many authors. Some of this work covers specific topics, such as methods for thin domains in 3D [12], mixed and penalty methods for incompressible problems [8], and the AMLI methods for compressible problems [1]. The optimal and nearly optimal preconditioners for Krylov subspace methods can often be designed within a general framework of space decomposition and

---

\* This article was produced during the first author’s visit to Delft University of Technology. The financial support from the Leonardo da Vinci Staff Exchange Programme is greatly acknowledged.

subspace correction (SSC) methods [19]. The idea behind the SSC strategy is to decompose, in a stable manner, the global finite dimensional space into a set of local subspaces. The global solution is obtained by the suitable combination of the local subspace problem solutions. The basis for the space decomposition for stress analysis is formed from the Cartesian components of the displacement vector. In this way, the local subspace problems represent material displacement along a single Cartesian coordinate, provided that displacements in other directions are fixed [1].

The local subspace problems have the properties of the scalar PDEs. Thus, there exists a variety of efficient methods for solving and preconditioning of these discrete linear systems. In [2] approximate factorisation is used to solve the subproblems. In [13] the local subproblems are approximated by the additive multilevel method (AMLI). In our study we adopt two different strategies for the solution of the local subproblems: direct sparse factorisation and the approximation by the scalar algebraic multigrid (AMG) solver. The main idea behind this choice is to implement the whole solver in a “black-box” fashion by using publicly available codes for both the Krylov solver and the local subproblems’ solvers/preconditioners. Direct sparse solvers are not (asymptotically) the optimal choice in this context, both in terms of the execution time and storage requirements. On the other hand, AMG method offer the prospect of optimal scaling with problem size [16], [17]. It is developed essentially for stiffness matrices corresponding to scalar elliptic PDEs (M-matrices). Thus, the scalar AMG solver is a suitable choice for preconditioning of the local subproblems.

This article is organised as follows. In Section 2 we introduce the physical problem and give details of the finite element discretisation. Section 3 describes our preconditioning strategy. In Section 4 we present numerical examples from microfabrication technology modelling. The aim is to demonstrate efficiency of the proposed methodology, as well as to outline some of its limitations and potential problems.

## 2 Problem Formulation

Deformation of a continuous material body occupying a bounded domain  $\Omega \subset \mathbb{R}^d$  ( $d = 2, 3$ ) is described by the following boundary value problem

$$-\nabla \cdot \sigma(\bar{u}) = \bar{f} \quad \text{in } \Omega; \quad \bar{u} = \bar{u}_D \quad \text{on } \partial\Omega_D; \quad \sigma(\bar{u})\hat{n} = \bar{u}_N \quad \text{on } \partial\Omega_N. \quad (1)$$

In (1)  $\sigma(\bar{u})$  denotes the Cauchy stress tensor,  $\bar{u}$  is the displacement vector, and  $\bar{f}$  is the body force vector. The vector  $\bar{u}_D$  represents the prescribed displacement of the boundary segment  $\partial\Omega_D$ , while  $\bar{u}_N$  is the surface traction of the boundary segment  $\partial\Omega_N$  with outward unit normal vector  $\hat{n}$  ( $\partial\Omega_D \cap \partial\Omega_N = \emptyset$ ,  $\partial\Omega_D \cup \partial\Omega_N = \partial\Omega$ ). For linear elasticity, the stress tensor  $\sigma(\bar{u})$  is related to the strain tensor  $\varepsilon(\bar{u})$  by Hooke’s law

$$\sigma(\bar{u}) = 2\mu\varepsilon(\bar{u}) + \lambda(\nabla \cdot \bar{u})I$$

where  $I$  is the identity tensor, and  $\mu > 0$  and  $\lambda > 0$  are Lamé’s coefficients, which, in turn, depend on Young’s modulus and the Poisson ratio [8]. Note that

in this study we consider only the cases of compressible linear elasticity, when  $\nu$  is bounded away from 0.5.

The variational stress analysis problem reads as follows: Find  $\bar{u} \in [V(\Omega)]^d$  such that

$$E(\bar{u}, \bar{v}) = F(\bar{v}) \quad \forall \bar{v} \in [V_0(\Omega)]^d, \quad (2)$$

where  $V_0(\Omega) = H_0^1(\Omega)$  is the standard Sobolev space of functions defined on  $\Omega$  with the homogeneous Dirichlet boundary conditions on  $\partial\Omega_D$ . It is possible to prove that problem (2) has a unique solution (see [5]). In (2)  $E(\bar{u}, \bar{v})$  represents the bilinear energy functional, and  $F(\bar{v})$  is the load vector functional (see [10]).

The discrete stress analysis problem can be obtained from (2) by applying the Galerkin projection of  $V(\Omega)$  onto  $V_h(\Omega) \subset V(\Omega)$ . In this way we obtain the problem: find  $\bar{u}_h \in [V_h(\Omega)]^d$  such that

$$E(\bar{u}_h, \bar{v}_h) = F(\bar{v}_h) \quad \forall \bar{v}_h \in [V_{h0}(\Omega)]^d.$$

In our case,  $V_h(\Omega)$  is the space of piecewise linear polynomials that correspond to a given partition of the domain  $\Omega$  into disjoint triangles/tetrahedra (2D/3D). If we introduce the nodal basis set  $V_h(\Omega) = \text{span}\{\phi_r\}_{r=1}^N$ , the discrete displacement vector component  $u_i$  can be written as  $u_i = \sum_{r=1}^N u_i^r \phi_r$ ,  $i = 1, \dots, d$ , with  $u_i^r$  being the unknown displacement nodal coefficients along the  $i$ th Cartesian coordinate. By adopting for the discrete test space  $V_{h0}(\Omega)$  the same basis set as for  $V_h(\Omega)$ , we obtain a system of  $d \cdot N$  linear algebraic equations

$$\sum_{j=1}^d \sum_{s=1}^N A_{ij}^{rs} u_j^s = f_i^r, \quad i = 1, \dots, d; \quad r = 1, \dots, N, \quad (3)$$

with

$$\begin{aligned} A_{ij}^{rs} &= \mu \sum_{k=1}^d (\partial_k \phi_r, \partial_k \phi_s) \delta_{ij} + \mu (\partial_j \phi_r, \partial_i \phi_s) + \lambda (\partial_j \phi_s, \partial_i \phi_r) \\ f_r^i &= (f_i, \phi_r) + \text{b.t.}, \end{aligned}$$

where  $(\cdot, \cdot)$  denotes the usual scalar product in  $L_2(\Omega)$ .

### 3 Preconditioning Methodology

The large, sparse linear system (3) can be solved by a host of iterative and direct techniques [3]. In this article we study the efficiency of the preconditioned Krylov subspace methods. In preconditioning, the aim is to design a matrix  $M$  (called the preconditioner) which is spectrally close to the coefficient matrix  $A$ , but simple to assemble and compute the action of its inverse. The aim is that the preconditioned matrix  $M^{-1}A$  has a small number of distinct eigenvalues with large multiplicity, or that the spectrum of  $M^{-1}A$  is tightly clustered and

bounded by the quantities that are (ideally) independent of the discretisation parameter  $h$  and other problem parameters.

In order to solve the system (3) by the SSC method, it is necessary to decompose the discrete finite element space  $V_h(\Omega)$  into a sum of subspaces  $V_h^{(i)}(\Omega)$ . Furthermore, the robust and effective implementation of SSC methods requires an efficient solution of the subspace problems. In this article we study two techniques: direct sparse solution of the subproblems, and the approximate solution of the subproblems by the scalar AMG. In our methodology we adopt a component-wise space decomposition, where for  $\bar{u} \in V_h^{(i)}(\Omega)$  we have  $u_j = 0$  if  $i \neq j$  [1]. In the case of block diagonal subspace correction the preconditioner is defined as  $M_D = D$ , while in the case of block lower triangular subspace correction we have  $M_L = D + L$  ( $L$  is the block lower triangular part of  $A$ ).

We comment briefly on the suitability of the AMG as an approximate solver for the local subproblems. The local subspace operators

$$\mathcal{D}(u, v) = \mu \sum_i (\nabla u_i, \nabla v_i) + (\lambda + \mu) \sum_i (\partial_i u_i, \partial_i v_i) \quad (4)$$

are anisotropic as the second sum contains div-type operators. In our approach we apply scalar AMG as a “black-box” solver to the entire discrete local subproblems (4). Because of the coarsening strategy adopted in AMG (coarsening is performed in the directions in which the error varies smoothly [17]), we expect efficient performance of the AMG with respect to anisotropies and discontinuities along the different material interfaces. The possible scenarios where the AMG strategy may encounter difficulties are the cases of nearly incompressible materials ( $\lambda \rightarrow \infty$ ), very thin domain structures (with large ratios between the principal dimensions), and highly non-convex domains (see Example 2 in Section 4).

## 4 Numerical Examples

A component based preconditioning strategy is practically tested on stress analysis problems in 2D and 3D that arise in microfabrication technology. However, the suggested methodology is not restricted in any way to this particular application. Microfabrication technology is concerned with the design and production of microelectronic and micromechanical components [9].

Having established superiority of the component based preconditioning over the standard ILU(0) preconditioning [10], we now examine the efficiency (in terms of wall clock time and memory requirements) of our methodology when various strategies are employed for the solution of local subproblems. For completeness we compare our results with a general direct sparse solver from the HSL library [6]. First we give some implementation details. We adopt BiCGSTAB(2) as a representative Krylov subspace method (see [15]). In our implementation, in line with the strategy of developing a fully “black-box” solver, we take a BiCGSTAB( $\ell$ ) implementation from the NAG library [11]. The exact version of

the SSC preconditioner (with sparse direct factorisation of the blocks) is implemented using the code HSL\_MA42 from the HSL library [6]. This code implements the frontal Gaussian elimination and is suitable for non-symmetric linear systems [4]. We also use the MSRO strategy for reordering the equations within each diagonal block [14], as implemented in the routine MC62 from the HSL library [6]. An inexact version of the SSC preconditioner (with diagonal blocks preconditioned by a small fixed number of AMG cycles) is implemented using the well-established and publicly available code AMG1R5 [16]. The approximate AMG solving procedure consists of one V(1,1) cycle. Two model problems are studied (one in 2D and one in 3D). The discrete linear systems (3) are generated by the Taurus simulator [18]. The right hand side vector is chosen randomly (although the elements of  $\mathbf{f}$  are scaled appropriately in order to simulate the realistic situation of a randomly distributed load).

Next we describe our experiment environment. All tests are performed on an Origin 3400 architecture with 16 R12000 processors (although no parallelism is considered). A single R12000 processor has a clock speed of 400MHz, corresponding to a theoretical peak performance of 800MFlops. The system has 4GB of RAM. All our codes are written in Fortran 90 and optimised using the high compiler optimisation.

**Example 1.** We consider the model problem in 3D. The domain represents one quarter of the intermediate structure (the domain is restricted by using suitable symmetries) during fabrication of the Fin Field Effect Transistor (FinFet) (see [18]). The FinFet technology offers the possibility for further miniaturisation of the transistor structure to nanoscale dimensions ( $< 50\text{nm}$ ). The present structure consists of four different material domains, which have different coefficients of thermal expansion. During production the whole structure is subject to high temperature variations. This process causes the build-up of stress in the structure (see [7]). The domain is discretised by unstructured grids of various sizes. The idea of studying this domain geometry is to test the robustness of the proposed iterative technique in the case of relatively “thin” domain geometries in 3D. The standard discretisation and solution techniques may encounter difficulties in this context [12]. Table 1 gives the number of iterations ( $N_{it}$ ) and the total execution time ( $T_t$ ) required by the preconditioned BiCGSTAB(2) algorithm to reduce the initial residual below  $10^{-6}$  for both the exact (E) and inexact (I) version of the SSC preconditioner. In Table 1,  $n$  denotes the total dimension of the problem and  $M_D$  and  $M_L$  stand for the block diagonal and the block triangular versions of the preconditioner.

The asterisk in the last column of Table 1, part (E) indicates that the matrix size is beyond the memory capacity of a particular architecture (we did not use direct access files for storing partially factorised matrices, see [6]). It can be concluded that this implementation gives a spectrally independent preconditioner, however, neither the execution time nor the storage requirements are optimal.

Next we report the results obtained for the inexact version of the SSC preconditioner. Part (I) of Table 1 presents the iteration counts and the execution time of the BiCGSTAB(2) algorithm with the inexact SSC preconditioner. In this case we observe a very moderate growth in the iteration counts. However, AMG

is an optimal solver both in terms of execution time and memory requirements. The total storage for the largest case from Table 1, part (I) is approximately 900MB. The optimality in terms of execution time is reflected in the rows  $T_t$  of Table 1, part (I). These times scale near linearly with the problem size and are much smaller than in the exact case.

Finally, for completeness, we contrast the timing results from parts (E) and (I) in Table 1 with the execution times required by a general sparse direct solver to complete this task (part (D) of Table 1). In this context we employ HSL\_MA42 code with the MSRO reordering of the equations (MC62 code) [6].

**Table 1.** Iteration counts and the execution time (in seconds) of the BiCGSTAB(2) algorithm for the case of the exact (E) SSC preconditioner (direct factorisation of the discrete local subproblems by HSL\_MA42) and the inexact SSC preconditioner (I) (preconditioning of the local subproblems by one V(1,1) AMG cycle). Execution time (in seconds) of the direct (D) sparse solver HSL\_MA42.

$n$		21987	38550	51120	113085	224292
E	$M_D$	$N_{it}$	6	6	6	*
		$T_t$	17.16	32.12	65.90	698.31
	$M_L$	$N_{it}$	4	4	4	*
		$T_t$	16.51	30.00	62.32	675.98
	$M_D$	$N_{it}$	8	10	10	12
		$T_t$	4.54	9.57	12.57	37.37
I	$M_D$	$N_{it}$	6	8	6	8
		$T_t$	4.38	9.42	10.58	30.73
	$M_L$	$T_t$	47.76	86.77	223.48	*
D	$T_t$	47.76	86.77	223.48	*	*

**Example 2.** We examine the efficiency of our methodology when applied to a model problem in 2D. The structure represents a trench obtained by etching the silicon substrate and depositing the nitride over it (see [18]). In microfabrication technology trenches are frequently used for the electromagnetic isolation of integrated active and passive components [20]. The main feature of this problem, as opposed to Example 1 and the example from [10] is that the domain  $\Omega$  is not convex. This can potentially cause some difficulties to the iterative solver using the AMG preconditioner. In this example, stress in the structure is caused by the deposition of a material (nitride in this case) with intrinsic built-in stress. In the following experiments we are following exactly the same procedure as in Example 1. Table 2 presents the iteration counts and the execution time of the preconditioned BiCGSTAB(2) algorithm with the exact (E) and inexact (I) version of the SSC preconditioner needed to reduce the infinite norm of the residual below  $10^{-6}$ .

**Table 2.** Iteration counts and the execution time of the BiCGSTAB(2) algorithm for the case of the exact (E) version SSC preconditioner (direct factorisation of the discrete local subproblems by HSL\_MA42) and the inexact SSC preconditioner (I) (preconditioning of the local subproblems by one V(1,1) AMG cycle). Execution time (in seconds) of the direct (D) sparse solver HSL\_MA42.

$n$		4000	9860	17464	34988	69316	136770
E	$M_D$	$N_{it}$	20	24	20	24	22
		$T_t$	1.08	4.07	8.42	24.37	212.58
	$M_L$	$N_{it}$	14	16	16	18	18
		$T_t$	0.89	3.13	7.55	23.50	61.52
		$T_t$	200.14				
I	$M_D$	$N_{it}$	50	58	72	84	92
		$T_t$	1.15	3.66	9.37	25.09	59.80
		$T_t$	135.95				
	$M_L$	$N_{it}$	38	46	58	66	72
		$T_t$	0.91	3.38	8.19	21.40	51.86
		$T_t$	115.20				
D		$T_t$	0.56	2.44	9.98	27.52	118.92
							*

From Table 2, part (E) we can conclude that the exact SSC preconditioner exhibits nearly optimal convergence characteristics in this context (although a higher number of iterations is required than in Example 1 to achieve the prescribed tolerance). From Table 2, part (I) it can be concluded that the inexact version of the SSC preconditioner encounters some problems with the convergence in this context. Moreover, the convergence characteristics exhibit an irregular pattern. Although the inexact version of the SSC preconditioner is not vastly superior in terms of the execution time as in Example 1 (compared to the exact version), it still has the advantage of substantially lower memory requirements (for the case  $n = 136770$  the total memory requirement is 260MB, compared to 3.7GB for the exact case, if the direct access files are not used). Finally, we contrast the timing results of the iterative method with the time required by the sparse direct solver HSL\_MA42 (Table 2, part (D)) to complete the same task. Again, direct access files are not used in this case.

From Table 2 it can be concluded that in the case of the 2D problem with non-convex domain geometry the iterative method is still faster than the direct solver, but the superiority is not as comprehensive as in the 3D case. Of course, the storage requirements argument is still on the side of the iterative solver. The potential reason for slow convergence of the inexact solver is that the coarse level approximations of the local subproblems do not reflect the domain geometry with sufficient accuracy. This problem can be partially alleviated by restricting the number of coarse levels in the automatic coarsening process [16]. However, this can cause relatively large problem sizes at the coarsest level (with fairly dense coefficient matrices). This can potentially reduce the number of iterations, but will lead to increase of the execution times.

## References

1. Axelsson, O.: On iterative solvers in structural mechanics; separate displacement orderings and mixed variable methods, *Math. Comp. Simulat.*, **50**(1999), 11–30.
2. O. Axelsson, O., Gustafsson, I.: Iterative methods for the solution of the Navier equations of elasticity, *Comp. Meth. Appl. Mech. Engng*, **15**(1978), 241–258.
3. Dongarra, J.J., Duff, I.S., Sorensen, D.C., van der Vorst, H.A.: *Numerical Linear Algebra for High-Performance Computers*, SIAM, Philadelphia, PA, 1998.
4. Duff, I.S., Scott, J.A.: The design of a new frontal code for solving sparse unsymmetric systems, *ACM Trans. Math. Soft.*, **22**(1)(1996), 30–45.
5. Fichera, G.: Existence theorems in elasticity, In: *Handbuch der Physik*, Vol. 6a/2 (S. Flüge, C. Truesdell eds.), Springer, Berlin, 1972, 347–389.
6. HSL: A collection of Fortran codes for large-scale scientific computation, 2002, (see <http://hsl.rl.ac.uk/>).
7. Hu, S.M.: Stress related problems in silicon technology, *Int. J. Appl. Phys.*, **15**(1991), R53–R80.
8. Hughes, T.J.R.: *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Prentice–Hall, Englewood–Cliffs, NJ, 1987.
9. Joppich, W., Mijalković, S.: Semiconductor Process Modelling, in: *Wiley Encyclopedia of Electrical and Eletronics Engineering* (J.G. Webster ed.), Wiley, New York, 1999, 127–139.
10. Mihajlović, M.D., Mijalković, S.: A component decomposition preconditioning for 3D stress analysis problems, *Numer. Linear Algebra Appl.*, **9**(2002), 567–583.
11. Numerical Algorithms Group: *NAG Manual*, Fortran Library, Mark 20, Oxford, 2002.
12. Ovtchinikov, E.E., Xanthis, L.S.: Iterative subspace correction methods for thin elastic structures and Korn’s type inequality in subspaces, *Proc. Roy. Soc. (London)*, Ser. A **454**(1998), 2023–2039.
13. Padiy, A.: On a robust multilevel method applied for solving large-scale linear elasticity problems, *Commun. Numer. Meth. Engng*, **15**(1999), 153–165.
14. Scott, J.A.: A new row ordering strategy for frontal solvers, *RAL Technical Report*, RAL-TR-1998-056, 1998.
15. Sleipen, G.L.G., Fokkema, D.R.: BiCGSTAB( $\ell$ ) for linear equations involving unsymmetric matrices with complex spectrum, *ETNA* **1**(1993), 11–32.
16. Stüben, K.: Algebraic multigrid (AMG): experiences and comparisons, *Appl. Maths and Comput.*, **13**(1983), 419–452.
17. Stüben, K.: A review of algebraic multigrid, *J. Comput. Appl. Maths*, **128**(2001), 281–309.
18. Taurus Process and Device: Multi-Dimensional Process and Device Modelling Program, Version 2002.4, User’s Manual, Synopsis, 2002.
19. Xu, J.: Iterative methods by space decomposition and subspace correction, *SIAM Review* **34**(1992), 581–613.
20. van Zant, P.: *Microchip Fabrication: A Practical Guide to Semiconductor Processing*, McGraw Hill, 2000.