Application of an Algebraic Multigrid Solver to Process Simulation Problems

T. Füllenbach, K. Stüben
Institute for Algorithms and Scientific Computing (SCAI)
German National Research Center for Information Technology (GMD)
Schloß Birlinghoven, D-53754 Sankt Augustin, Germany
tanja.fuellenbach@gmd.de, stueben@gmd.de

S. Mijalković
Department of Microelectronics - ECTM Laboratory
Faculty of Information Technology and Systems, Delft University of Technology
Feldmannweg 17, 2600 GB Delft, The Netherlands
s.mijalkovic@its.tudelft.nl

***Abstract* - In this paper the performance of a special algebraic multigrid (AMG) solver for the solution of the stress analysis problems in process simulation has been investigated. The discrete stress analysis equations are generated directly by the process simulator. The practical simulation examples include stress analysis during natively growing and deposited material films. It is demonstrated that a methodology where AMG is employed as a preconditioner for the standard iterative solvers results in a very effective solving procedure regarding computing times and convergence behaviour. A further comparison shows that AMG preconditioned iterative solvers are faster than the direct solver even for moderately small problems.**

I. Introduction

It is increasingly recognized that with the growing complexity of industrial process simulation applications, the performance of the solvers for the corresponding discrete problems becomes relatively much poorer. While direct solvers are approaching the problem of an unacceptable number of floating point operations, the standard iterative solvers are facing the problems of bad conditioning.

It is well known that multigrid methods offer the prospect of an (optimal) linear behaviour of the computing time dependently on the problem size. Standard multigrid methods have been already recognized as an efficient solving technique for process simulation problems if the underlying grid structures possess a natural hierarchy resulting from a local grid refinement [1]. However, the evolving geometry of the deformed multi-layer material regions in semiconductor fabrication is generally quite irregular. The unstructured grids describing such a geometry are typically not coarse enough to serve as the coarsest grid level in multigrid algorithms. This fact has significantly limited the applicability of multigrid methods in the process simulation. A promising approach to eliminate this deficiency is the usage of algebraic multigrid methods [2,3,4].

The main objective of this paper is to investigate the capability of an AMG solver to cope with discretized systems of the stress governing equations occurring in the semiconductor process simulation where standard iterative solvers (preconditioned BiCG and GMRes) have non-optimal convergence behaviour.

II. Problem formulation

The semiconductor process simulation involves generally two principal classes of PDE problems. The first one is related to the redistribution of dopants and point defects in thermal processes and requires the solution of multi-particle drift-diffusion-reaction equations. The second class of problems is related to the mechanical deformation of fabricated multi-layer material structures. In the latter case, the corresponding governing equations actually account for the distribution of the stresses and strains that develops in different material regions during fabrication. Due to the elliptic nature of the governing equations in the stress analysis problems they are more critical for the application of classical iterative solvers and therefore employed here in the formulation of the test problems.

The test problems for the numerical experiments are generated by the process simulator FLOOPS [5]. The stress analysis in FLOOPS, and generally in process simulation, is principally based on the momentum equation

$$-\nabla \cdot \boldsymbol{\sigma}_{\mathbf{d}} + \nabla p = \mathbf{f} \quad \text{in} \quad \Omega \tag{1}$$

where Ω is a bounded domain with boundary Γ, $\boldsymbol{\sigma}_{\mathbf{d}}$ is the symmetric deviatoric stress tensor, p is the mean pressure and $\mathbf{f}$ is the body force. The boundary conditions are given by

$$(-p\mathbf{I} + \boldsymbol{\sigma}_{\mathbf{d}}) \cdot \mathbf{n} = \mathbf{g} \quad \text{on} \quad \Gamma_{\mathrm{g}} \tag{2}$$

$$\mathbf{u} = \mathbf{h} \quad \text{on} \quad \Gamma_{\mathrm{u}} \ , \tag{3}$$

where $\mathbf{g}$ is the surface traction of the boundary segment $\Gamma_{\mathrm{g}} \subset \Gamma$, $\mathbf{h}$ is the displacement of the boundary segment $\Gamma_{\mathrm{u}} \subset \Gamma$ ($\Gamma_{\mathrm{u}} \cap \Gamma_{\mathrm{g}} = \emptyset$), $\mathbf{n}$ is the outward unit normal vector on the boundary and $\mathbf{I}$ is the identity tensor. Mechanical properties

of the materials involved in the semiconductor fabrication are varying from purely elastic solids to viscous fluids and therefore quite accurately modelled with the constitutive relationship of the Maxwell viscoelasticity. The Maxwell viscoelasticity is commonly implemented in process simulation in its incremental form based on the constitutive relationships of linear elasticity

$$\boldsymbol{\sigma}_{\mathbf{d}} = 2G_{\text{eff}} \left[\frac{1}{2} \left(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}} \right) - \frac{1}{3} \left(\nabla \cdot \mathbf{u} \right) \mathbf{I} \right] \quad \text{in} \quad \bar{\Omega} \quad (4)$$

$$p = -K \nabla \cdot \mathbf{u} \quad \text{in} \quad \bar{\Omega} \quad (5)$$

where the viscoelastic material properties are introduced by an effective shear modulus G_{eff} given by

$$G_{\text{eff}} = G \frac{\tau}{\Delta t} \left(1 - \exp\left(-\frac{\Delta t}{\tau} \right) \right) . \quad (6)$$

Here $\mathbf{u}$ is the incremental displacement vector, $G > 0$ and $K > 0$ are the shear and bulk moduli, Δt is the time step size, and τ is Maxwellian relaxation time defined as $\tau = \mu/G$, where μ is the material viscosity. Notice that G_{eff} provides a continuous modelling of the material mechanical behaviour from the purely elastic deformation to the viscous flow. Namely, for $G \ll \mu$ it reduces to the Hooke's law for the elasticity while for $\mu \ll G$, we obtain the Newton's law for the viscous fluids.

We have considered two practical text examples in our numerical experiments. The first one corresponds to the "single full integration stress solving step" in the simulation of the "Sealed Interface Local Oxidation" (SILO) process. The underlying grid structure is shown in Fig. 1.

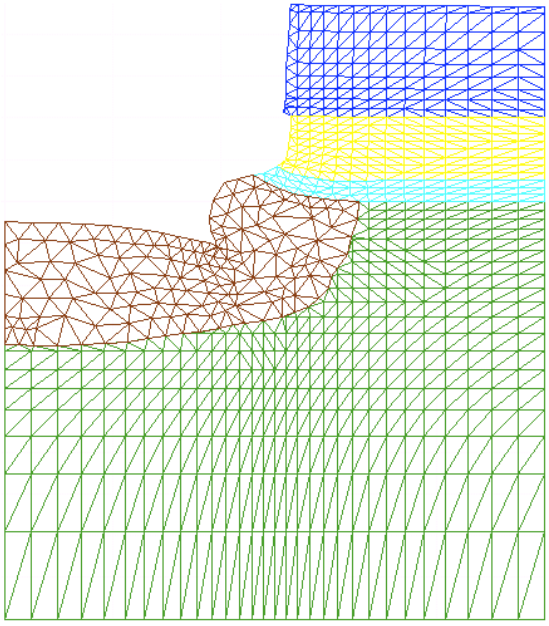

Figure 1: The grid structure for the SILO isolation problem (green=silicon, yellow & brown=oxide, blue & cyan=nitride).

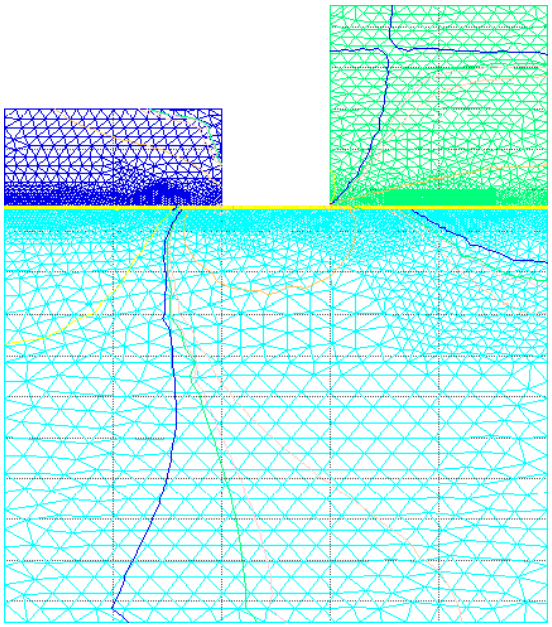

Figure 2: The grid structure (and some of the solution contour lines) for the DEPO3 problem with 8954 variables (blue=nitride, green=oxide, yellow=polysilicon, cyan=silicon).

The second class of simulation examples (DEPO) is related to the stress distributions in multilayer material regions after thin film deposition processes. The origin of the stress are intrinsic stress distributions in the deposited material films. Fig. 2 shows an example of the grid structure used in discretization of the model problem DEPO. It contains four different material regions. In order to test different problem scales, the problem DEPO is formulated with three different grid structures. The stress governing equations are discretized using standard piecewise linear Galerkin finite elements and the assembled global stiffness matrices are submitted directly to the AMG solver.

III. Algebraic multigrid

Today various different algebraic multigrid approaches exist. We here refer to a very flexible, robust and efficient approach in practice. It has been implemented in the code SAMG16 (called AMG below), a further development of RAMG05 [3] for PDE systems, which incorporates more efficient and more flexible interpolation and coarsening strategies than its predecessor AMG1R5 [6]. But because the research version we have used is under permanent development, it is not optimized especially in terms of computing times, and substantial improvements can be expected for a "final version".

Belonging to the group of multi-level methods, AMG is designed to solve certain types of sparse-matrix equations such as those typically arising from the discretization of elliptic partial differential equations (PDEs) or structurally similar problems. Although it employs the ideas of smoothing and coarse grid correction, it works differently than geometric multigrid and uses only information contained in the given matrix, and additionally (in the case of a discretized system of PDEs) simple informations about unknown-point-variable relations. Therefore it is especially suited for problems based on unstructured grids. For the same reason there is no need to completely restructure existing software packages: AMG can be seen as a kind of "plug-in" or "black box" solver.

For large classes of problems AMG is an efficient alternative to standard numerical iterative methods such as conjugate gradient (CG) or BiCGstab accelerated by typical (one-level) preconditioners. AMG can handle millions of unknowns and shows better convergence rates, to a large extent independent of the size of the given problem. Often the efficiency can be increased further by using (cheaper, low-memory) AMG variants as a preconditioner for GMRes or BiCGstab. There is also the possibility to parallelize AMG which has been shown recently [4].

The classes of problems, AMG has successfully been applied to, include for example problems on very complex, unstructured grids and problems with strongly anisotropic and discontinuous coefficients (e.g. oil reservoir simulation). But most of these cases require after all the solution of a single partial differential equation. Although the AMG approach has already successfully been used for solving various types of PDE systems, the development has not yet reached a state where a particular approach is well-settled.

IV. Numerical results

A. Comparison of different iterative approaches

The basic iterative approaches and their preconditioners, which are used for solving the created problems, are listed in Table I. Results of numerical tests with the SILO and the DEPO examples are given in Tables II to VII.

TABLE I
DESCRIPTION OF THE DIFFERENT APPROACHES

Name	Basic approach[a]	Preconditioner[b]
AMG	standard AMG	none
AMG-BiCG	BiCGstab	2 AMG it
BiCG	BiCGstab	ILUT(9)
AMG-GM20	GMRes(20)	1 AMG it
GM20	GMRes(20)	ILUT(9)
AMG-GM4	GMRes(4)	1 AMG it
GM4	GMRes(4)	ILUT(3)

[a]GMRes(x): x=dimension of the Krylov space
[b]it=iteration(s)

TABLE II
RESULTS FOR THE SILO EXAMPLE WITH 1810 VARIABLES AND 23079 NON-ZERO MATRIX ENTRIES: AVERAGE RESIDUAL REDUCTION FACTORS (ARF), NUMBER OF CYCLES NEEDED TO REDUCE STARTING RESIDUAL BY 10^{10} AND RUN TIME (IN SECONDS).

Approach	ARF	cycles	time
AMG-GM4	0.265	18	0.36
AMG-GM20	0.250	17	0.37
BiCG	0.519	36	0.39
GM20	0.744	78	0.52
AMG-BiCG	0.199	15	0.54
AMG	0.537	38	0.63
GM4	0.846	138	0.82

TABLE III
RESULTS FOR THE DEPO1 EXAMPLE WITH 1432 VARIABLES AND 15912 NON-ZERO MATRIX ENTRIES

Approach	ARF	cycles	time
AMG-GM20	0.117	11	0.22
AMG-GM4	0.123	12	0.23
AMG-BiCG	0.016	7	0.26
BiCG	0.340	25	0.27
GM20	0.634	51	0.28
AMG	0.267	18	0.42
GM4	0.955	505	1.65

TABLE IV
RESULTS FOR THE DEPO2 EXAMPLE WITH 3224 VARIABLES AND 38425 NON-ZERO MATRIX ENTRIES

Approach	ARF	cycles	time
AMG-GM20	0.211	15	1.86
AMG-GM4	0.239	17	1.90
AMG-BiCG	0.052	9	2.04
BiCG	0.553	39	2.13
AMG	0.475	31	2.95
GM20	0.824	119	4.15
GM4	0.985	1526	29.96

TABLE V
RESULTS FOR THE DEPO3 EXAMPLE WITH 8954 VARIABLES AND 121868 NON-ZERO MATRIX ENTRIES

Approach	ARF	cycles	time
AMG-GM20	0.344	22	8.0
AMG-BiCG	0.174	14	9.3
AMG-GM4	0.528	37	12.0
BiCG	0.749	82	13.5
AMG	0.854	146	38.6
GM20	0.949	437	44.6
GM4	0.990	(1000)[a]	(64.7)[a]

[a]Residual reduction by a factor of 10^4 only.

Notice that AMG-GM20 and AMG-GM4 (for smaller numbers of variables) or BiCG (for larger numbers of variables) are the fastest approaches (for the residual reduction of 10 orders of magnitude). AMG-BiCG reduces the residuals best. Especially for DEPO1 and DEPO2 it convergences exceptionally well, and although two AMG steps are used for preconditioning, for a residual reduction by more than 10^{10}, it is faster than AMG-GM.

TABLE VI
RESULTS FOR THE DEPO4 EXAMPLE WITH 16836 VARIABLES AND 230896 NON-ZERO MATRIX ENTRIES

Approach	ARF	cycles	time
AMG-GM20	0.498	34	23.6
AMG-BiCG	0.273	18	24.2
BiCG	0.812	118	37.3
AMG-GM4	0.742	78	48.3
GM20	0.953	483	99.9
AMG	0.947	422	210.9
GM4	0.996	(2000)[a]	(288.9)[a]

[a]Residual reduction by a factor of 10^3 only.

TABLE VII
RESULTS FOR THE DEPO5 EXAMPLE WITH 21794 VARIABLES AND 299593 NON-ZERO MATRIX ENTRIES

Approach	ARF	cycles	time
AMG-BiCG	0.251	19	34.3
AMG-GM20	0.523	36	36.3
BiCG	0.837	130	55.5
AMG-GM4	0.749	80	66.0
GM20	0.962	597	158.2
AMG	0.950	449	297.8
GM4	0.996	(2000)[a]	(380.8)[a]

[a]Residual reduction by a factor of 10^3 only.

The approaches with AMG as preconditioner show much better ARFs than the other approaches tested. AMG as a stand-alone solver is much better than GM4 and has a better ARF than GM20, but it is slower than BiCG and GM20 (exceptions: DEPO2 and DEPO3). For the DEPO1 and DEPO2 examples AMG has a better ARF than BiCG, for the other examples it is the other way around.

By looking at the residual development for larger problem sizes (DEPO3 to DEPO5), one can see that both BiCGstab and GMRes show residual reduction rates close to or even larger than 1 in some steps and a partially oscillating behaviour, but the residual reduction factors for AMG (as a stand-alone solver) increase monotonously, at first in quick steps, then slowly, and converge against a final value smaller than, but close to 1. For both cases the conclusion can be drawn that the iteration matrices have some eigenvalues close to 1, and therefore the approaches are not able to reduce the parts of the error that belong to the corresponding eigenvectors.

AMG as a preconditioner for BiCGstab or GMRes improves the spectral radius of the corresponding iteration matrix substantially. The residual reduction rates are bounded away from 1 but are varying.

Although AMG-BiCG or AMG-GM (example dependent) are the best approaches for the problems tested, it is obviously necessary to improve the capability of the AMG solver to cope with discretized systems of PDEs or structurally similar problems, e.g. by preconditioning the matrices to be solved in consideration of the special system structure.

B. Comparison with SuperLU

To get a first impression how fast the AMG approaches are, we have compared AMG-BiCG to the direct solver SuperLU (a fast incomplete LU factorization built into FLOOPS) by measuring total wall-clock times for computing the solution (including grid generation). The results, which are shown in Table VIII, are of great promise, particularly because SAMG16 can be optimized further.

TABLE VIII
WALL-CLOCK TIMES FOR COMPUTING THE SOLUTION WITH SuperLU AND AMG-BiCG (IN MINUTES)

Example	DEPO4	DEPO5
AMG-BiCG	1:37	2:26
SuperLU	2:26	3:13

V. CONCLUSION

In summary the numerical experiments have demonstrated the following: The usage of the AMG solver SAMG16 as a preconditioner for BiCGstab or GMRes (with a large dimensional Krylov space) results in approaches which are better than the standard iterative solvers BiCGstab and GMRes which are currently used in process simulation tools. The AMG approaches are faster and their convergence rates are smaller and more stable. Since the benefits of AMG solvers increase with the problem size and since SAMG16 can be optimized and developed further, it can be expected that AMG could play an essential role in the next generation of process simulation tools.

REFERENCES

[1] W. Joppich and S. Mijalković, *Multigrid Methods for Process Simulation*, Springer-Verlag, Wien, New York, 1993.

[2] K. Stüben, *A Review of Algebraic Multigrid*, GMD Report 69, November 1999, to appear in *J. Comp. Appl. Math.*, 2000.

[3] K. Stüben, *Algebraic Multigrid (AMG): An Introduction with Applications*, GMD Report 70, November 1999. This report will appear as a guest contribution (appendix) in the book "Multigrid" by U. Trottenberg, C.W. Oosterlee, A. Schüller; Academic Press, to appear 2000.

[4] A. Krechel and K. Stüben, *Parallel Algebraic Multigrid Based on Subdomain Blocking*, GMD Report 71, December 1999.

[5] M. E. Law, *Florida Object-Oriented Process Simulator, FLOOPS/FLOODS Manual*, University of Florida, Gainesville, 1993.

[6] J. W. Ruge and K. Stüben, *Parallel Algebraic Multigrid (AMG)*, in "Multigrid Methods" (McCormick, S.F., ed.), SIAM, Frontiers in Applied Mathematics, vol. 5, Philadelphia, 1986.