Algebraic Multigrid for Selected PDE Systems

T. Füllenbach, K. Stüben
Fraunhofer Institute for Algorithms and Scientific Computing (SCAI)
Schloss Birlinghoven, D-53754 Sankt Augustin, Germany
tanja.fuellenbach@scai.fraunhofer.de
klaus.stueben@ scai.fraunhofer.de

Abstract - **In this paper, strategies for solving systems of partial differential equations by algebraic multigrid are discussed. In particular, a general framework for so-called point-based strategies is introduced. For a demonstration, we have investigated several industrial applications from semiconductor process and device simulation. It is shown that this framework allows to construct robust and fast algebraic multigrid approaches even for cases, where iterative solvers of the type commonly used in such applications exhibit bad convergence or even fail.**

I. Introduction

Classical algebraic multigrid (AMG) [1, 3] is known to provide very efficient and robust solvers or preconditioners for large classes of matrix problems,

$$Au = f\,,$$

an important one being the class of (sparse) linear systems with matrices A which are “close” to being M-matrices. Problems like this widely occur in connection with discretized *scalar* elliptic partial differential equations (PDEs). In such cases, classical AMG is very mature and can handle millions of variables much more efficiently than any one-level method. Since explicit information on the geometry (such as grid data) is not needed, AMG is especially suited for unstructured grids both in 2D and 3D. In fact, the coarsening process is directly based on the connectivity pattern reflected by the matrix A and interpolation is constructed based on the matrix entries.

However, extensions of classical AMG are required to efficiently solve *systems* of PDEs involving two or more scalar *functions* (called *unknowns* in the following). This is because classical AMG realizes a *variable-based approach* which does not distinguish between different unknowns. Unless the coupling between different unknowns is very week, such an approach cannot work efficiently for systems of PDEs where, in general, the corresponding matrix A is far from being an M-matrix.

In the past, several ways to generalize AMG have been investigated and there is still an ongoing rapid development of new AMG and AMG-like approaches. For a review, we refer to [2]. However, there is no unique and best approach yet. In fact, none of the known approaches is really satisfactory in dealing with practically relevant problems and many problems cannot be tackled at all yet. All methods seem to have their range of applicability but all of them may fail to be efficient in certain other applications. In this paper, the focus is on extensions of AMG which are direct generalizations of the classical approach.

We first want to recall a rather popular AMG approach to solve systems of PDEs, the so-called *unknown-based approach*, which is very similar to the variable-based approach except that all unknowns are treated separately. To be more specific, let us assume the variables to be ordered by unknowns, that is, $Au = f$ has the form

$$\begin{bmatrix} A_{[1,1]} & \cdots & A_{[1,nsys]} \\ \vdots & \ddots & \vdots \\ A_{[nsys,1]} & \cdots & A_{[nsys,nsys]} \end{bmatrix} \begin{bmatrix} u_{[1]} \\ \vdots \\ u_{[nsys]} \end{bmatrix} = \begin{bmatrix} f_{[1]} \\ \vdots \\ f_{[nsys]} \end{bmatrix} \tag{1}$$

where $nsys > 1$ denotes the number of unknowns of the given system of PDEs, $u_{[n]}$ denotes the vector of variables corresponding to the n-th unknown and the matrices $A_{[n,m]}$ reflect the couplings between the n-th and the m-th unknown. Using this notation, coarsening the set of variables corresponding to the n-th unknown is strictly based on the connectivity structure reflected by the submatrix $A_{[n,n]}$ and interpolation is based on the corresponding matrix entries. In particular, interpolation to any variable i involves only coarse-level variables corresponding to the same unknown as i. The Galerkin matrices, however, are usually computed w.r.t. all unknowns.

The unknown-based approach, which has been proposed already in the very early papers on AMG (see [1]), is certainly the simplest approach for solving PDE systems. By now a lot of experience has been gained with this approach which, in practice, works quite efficiently for many applications. Compared to the variable-based approach, the only additional information required is information about the correspondence between variables and unknowns. The unknown-based approach is mainly used for applications where the diagonal matrix blocks $A_{[n,n]}$ are close to being M-matrices. The essential additional condition for the approach to work is that smoothing the individual equations is sufficient to cause the resulting error to be smooth separately for each unknown. One advantage of this approach is that it can easily cope with anisotropies which are different between the different unknowns. Another advantage is that unknowns can virtually be distributed arbitrarily across mesh points. However, this approach will become inefficient, for instance, if the coupling between different unknowns is too strong.

In this paper, we focus on applications for which the unknown-based approach does not work, unless we introduce very special modifications. In particular, we consider reaction-diffusion equations from semiconductor process simulation which lead to matrices A for which the submatrices $A_{[n,n]}$ are far from being M-matrices. In fact, off-diagonal entries may be larger than the diagonal entry by *orders of magnitude*. Hence, the size of matrix entries is no measure any more to decide about the strength of connectivity in the AMG context.

In Section II, we outline a flexible framework for constructing new AMG approaches to solve various types of PDE systems. In contrast to the previous approach, all of the new ones operate on the level of grid points rather than variables. Based on this framework, our AMG code "RAMG", described in detail in [3], has been substantially generalized to provide more flexibility in solving PDE systems. Recent results for industrial applications in semiconductor process and device simulation, obtained by the generalized code (called "SAMG"), are presented in Section III, showing that suitable point-based AMG approaches yield efficient solution processes.

II. A GENERAL FRAMEWORK FOR POINT-BASED APPROACHES

We talk about a *point-based approach* if, geometrically speaking, coarsening takes place on the level of points (rather than variables as before) and all unknowns are defined on the same hierarchy. Note that this is different from the unknown-based approach where each unknown is associated with its own hierarchy.

Since we have the solution of PDEs in mind, we think of points as being real physical (grid) points in space. However, we want to point out that, from AMG's point of view, it is not important whether "points" really correspond to physical points. Instead, one may think of the nodes of a graph representing the connectivity structure of A. Regarding a point-based approach, it is only relevant for AMG to know whether there are "blocks" of variables (corresponding to different unknowns) which may be treated (coarsened and interpolated) simultaneously. We assume that corresponding information is available to AMG.

In all our point-based approaches the coarsening process is performed based on some auxiliary (sparse) $(npnts \times npnts)$-matrix $P = (p_{kl})$, called the *primary matrix*, with $npnts$ denoting the number of points. The same coarse levels are then assigned to all unknowns. For this process to make sense, the employed primary matrix should reflect the physical connectivity (the general structure as well as the strength of connections) of neighboring variables reasonably well, *simultaneously for all unknowns*.

A special point-based approach, sometimes called "block approach", has already been introduced in the very early paper [1] and has been further investigated, for instance, in [4]. To be more specific, we assume the variables to be ordered pointwise, that is, $Au = f$ has the form

$$\begin{bmatrix} A_{(1,1)} & \cdots & A_{(1,npnts)} \\ \vdots & \ddots & \vdots \\ A_{(npnts,1)} & \cdots & A_{(npnts,npnts)} \end{bmatrix} \begin{bmatrix} u_{(1)} \\ \vdots \\ u_{(npnts)} \end{bmatrix} = \begin{bmatrix} f_{(1)} \\ \vdots \\ f_{(npnts)} \end{bmatrix}, \qquad (2)$$

where $u_{(k)}$ denotes the "block" of variables located at point k and the $(nsys \times nsys)$-matrix $A_{(k,l)}$ represents the "block coupling" between $u_{(k)}$ and $u_{(l)}$. Block coarsening corresponds to defining the primary matrix P by

$$p_{kl} = -||A_{(k,l)}|| \quad (k \neq l) \quad \text{and} \quad p_{kk} = -\sum_{l \neq k} p_{kl} \qquad (3)$$

with $||.||$ denoting a suitable norm. Various different norms have been considered in practice.

Depending on the type of application, there are many other possibilities for defining a primary matrix. Often, this can be done automatically as part of AMG's setup phase. In other cases, it may be better to let the user of AMG provide a reasonable matrix himself, based on his knowledge of the underlying physics of the given problem. In such cases, a primary matrix can usually be interpreted as describing the connectivity structure of some auxiliary (scalar) *primary unknown*. Clearly, this primary unknown should represent the connectivity structure of all "real" unknowns in the given system of PDEs reasonably well.

For instance, in simple cases, one may select $P = A_{[n,n]}$ with n being any of the unknowns of the given system of PDEs. Whether or not this makes sense, depends on the application, in particular, whether the connectivity structure of the n-th unknown is also representative for the other unknowns. If anisotropies in a given problem are mainly due to non-uniform mesh spacings, a simple primary matrix might be given by a discretization of the Laplace operator. One can also imagine cases where it makes sense to define a primary matrix based on some natural physical quantity for which there is no reasonable equation contained in the original system of PDEs, an example being the pressure in the context of the Navier-Stokes equations.

The original AMG did not exploit any information on the given problem apart from the matrix A itself. In many PDE applications, this unnecessarily limits the possibilities for an efficient coarsening and interpolation. As a matter of fact, geometric information such as the coordinates of grid points, is usually available and can be exploited in AMG's setup phase. Note that this does not restrict the generality of the grid shape in any respect. If we assume coordinates to be known, P may often most easily (and automatically) be defined based on distances of points, leading to coarsening processes which are closely related to geometric coarsening. The most simple definition would be

$$p_{kl} = -1/\delta_{kl}^2 \quad (k \neq l) \quad \text{and} \quad p_{kk} = -\sum_{l \neq k} p_{kl} \tag{4}$$

where δ_{kl} denotes the distance between points k and l. (Clearly, since P has to be sparse, only points in small neighborhoods - corresponding to the non-zero pattern of A - are taken into account here.)

Remark: In practice, often, not all unknowns are represented at a point, that is, the number of variables may vary from point to point (cf. the applications discussed in Section III). If this is the case, a reasonable primary matrix is required to "represent" all points.

The main purpose of a primary matrix is to define an AMG hierarchy in terms of points. As part of the (recursive) coarsening process, also interpolation operators have to be constructed. In practice, there are various possibilities to generalize the interpolation approaches used in classical AMG.

First, the use of *block interpolation* seems most natural, in particular, if P is defined according to (3). That is, a formula to interpolate the error $e_{(k)}$ at a point k is constructed by approximating the block equations

$$e_{(k)} = -A_{(k,k)}^{-1} \sum_{l \neq k} A_{(k,l)} e_{(l)} \tag{5}$$

in a way which is completely analogous to the classical approaches to define interpolation. However, this type of interpolation is very expensive. In practice, simpler types of interpolation often lead to more efficient AMG processes. Thus, besides the above block interpolation, we consider (variable-wise defined) interpolation formulas which are either

- separate for each unknown, or
- the same for each unknown.

This classifies interpolation just according to its general *structure*. Typical ways to define the interpolation *weights* are based on entries in the original matrix A, based on distances and/or positions of points, or based on entries in the primary matrix P. We cannot go into further details here but just want to mention that "classical" interpolation schemes, as described in [3], such as *direct*, *standard* or *multi-pass interpolation*, can be generalized to this setting in a straightforward way. Also the concept of *aggressive coarsening* carries over.

The general framework outlined above formally allows to define various concrete algorithms. It seems clear that there exists no unique AMG procedure which will work satisfactorily for all systems of PDEs. Instead, major work consists in developing concrete algorithms separately for certain classes of industrial applications.

III. Applications

Due to the complexity of the models and grids used, industrial semiconductor process and device simulation is increasingly recognized as an important and challenging area for numerical simulation. Corresponding PDE systems include *stress governing*, *reaction-diffusion* and *drift-diffusion equations*, all of which exhibit different numerical difficulties. That simple unknown-based AMG is suitable to speed up stress simulations has already been shown in [5]. For reaction-diffusion and drift-diffusion equations, the situation is considerably more complicated. Where classical iterative solvers often converge only slowly (or even break down) and straightforward unknown-based AMG is not sufficient any more, suitable point-based AMG approaches, accelerated by BiCGstab or GMRes, can still cause remarkable speedups. In the following, we present some typical examples. We will see that reaction-diffusion problems can efficiently be solved by using a primary matrix based on geometric distances (4) and an interpolation which is separate for each unknown with weights being also defined based on distances. Drift-diffusion problems, on the other hand, are solved efficiently by selecting the primary matrix based on norms (3) and choosing interpolation to be the same for each unknown with weights being based on the entries of P.

3.1. Process simulation: reaction-diffusion equations

Systems of reaction-diffusion equations occur, for instance, in the simulation of annealing steps after ion implantation into a wafer. They consist of a sequence of balance equations of the form ([6, 7])

$$\frac{\partial u_i}{\partial t} + \nabla \cdot J_i = R_i \quad (i = 1, \dots, N) \tag{6}$$

where the J_i denote (diffusion and field driven) fluxes given by

$$J_i = -D_i \left(\nabla u_i + \frac{q}{k_B T} u_i \nabla \psi \right) . \tag{7}$$

Here u_i denotes the concentration of the i-th species, N their number, ψ the electrostatic potential, $R_i = R_i(u_1, \dots, u_N)$ a reaction term (often a polynomial), D_i the diffusivity, q the elementary charge, k_B the Boltzmann constant and T the absolute temperature.

After inserting (7) into (6) for each i, the above system consists of typically 30 to 40 equations. By employing some equilibrium assumptions, a reduction to a system of 3 to 6 equations of a similar form as (6) can be performed, with unknowns being the concentration of species such as interstitials, vacancies, Arsenic, Boron, Phosphorus or others. For the potential ψ an additional Poisson's equation can be solved, which could be coupled to the above system. We have only investigated the typical, uncoupled case.

In common process simulators, an implicit approach is chosen for the time discretization. The spatial discretization is performed by the so called "box method" on Delaunay grids, and the resulting nonlinear system is linearized by a modified Newton-Raphson method. ILU-preconditioned BiCGstab or GMRes are commonly used as solvers for the resulting linear systems. More precisely, modified ILUT [8] or even ILU(0) methods are employed.

Of particular interest are the concentration profiles in and near the *reaction front*, a narrow region, moving from the "implantation" surface of the wafer towards the interior, where fast reactions occur due to large concentration gradients (see for example [9]). In this region, the reaction terms cause very large positive or negative off-diagonal

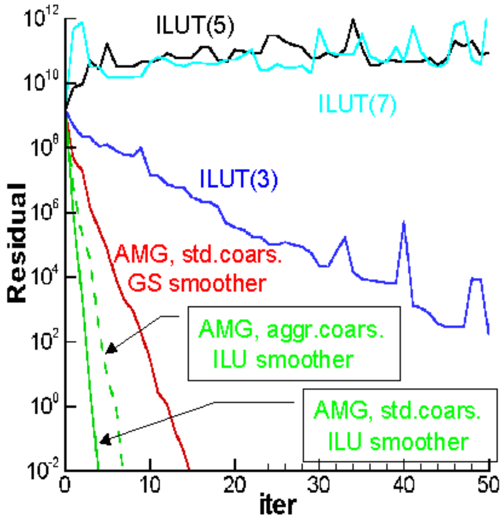

Figure 1: Convergence histories for a 3D reaction-diffusion example. All AMG and ILUT variants are accelerated by BiCGstab ("aggr." or "std.coars.": aggressive or standard coarsening, resp.; GS: Gauss-Seidel).

entries in the corresponding rows of the matrices A, leading to serious problems for the standard iterative solvers mentioned above. It can be observed that these solvers are getting less efficient or even stagnate in an unpredictable way. The difficulties often increase during later time steps of a simulation. As an example, Fig. 1 depicts the convergence histories for several ILUT-variants used as preconditioners for BiCGstab. The solvers are applied to a typical system, arising in a 3D-simulation in a particular step within Newton's iteration.

Because of the very large off-diagonal elements, not only the full matrices A, but also their submatrices $A_{[n,n]}$ are far from being M-matrices. Therefore a straightforward unknown-based AMG method does not make sense here (see, however, the remark below).

As can be seen in Fig. 2, the underlying grids are adaptively refined in the reaction front. Hence, it seems promising to define a point-based AMG method with a primary matrix based on distances and a separate interpolation with weights also based on distances. This combination yields an AMG method which treats both the "pure" diffusion outside the reaction front and the fast reactions inside properly (see Fig. 3). As can be seen from Fig. 1, three AMG variants (with standard or aggressive coarsening, Gauss-Seidel- or ILU-smoother), accelerated by BiCGstab, all converge much faster than the ILUT-preconditioned ones. Moreover, this behavior is typical for a whole simulation: AMG yields stable and superior convergence rates for residuals *and* errors in all Newton iterations for *each* time step, which makes it more robust and efficient than the ILUT-preconditioners.

Remark: As mentioned above, due to the large off-diagonals arising from fast reactions, a straightforward unknown-based approach does not work. However, while these reactions take place only inside a narrow area, outside a relatively 'harmless' diffusion problem remains to be solved. Numerical experiments have shown that, by introducing a very simple modification of AMG's coarsening process, one can make the unknown-based AMG work again: Simply do not coarsen at all inside the narrow reaction area. To demonstrate this, we have forced all those variables i to stay in the coarse levels, whose corresponding rows strongly violate diagonal dominance,

$$\sum_{j \neq i} |a_{ij}| > \epsilon |a_{ii}| \,. \tag{8}$$

Depending on the threshold parameter $\epsilon > 1$, the resulting unknown-based AMG (employed as a preconditioner) converges similar to the point-based approach described before. However, the choice of ϵ is crucial and example-dependent. Clearly, the above criterion is not always suitable to reliably distinguish the reaction-dominant from the slow-diffusion part, and therefore the resulting approach is not robust enough for a use in practice.

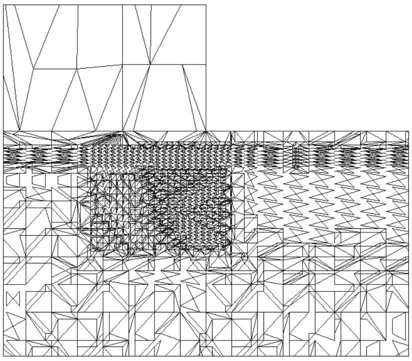

Figure 2: 3D reaction-diffusion example: (part of a) 2D cross-section of the grid which is adaptively refined near an edge of the interface oxide/wafer.

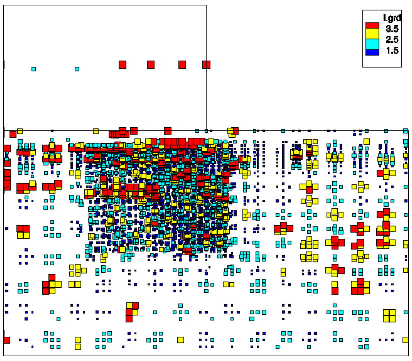

Figure 3: Coarse levels for the grid on the left: the larger the box, the longer the corresponding point stays in the coarse levels.

3.2. Device simulation: drift-diffusion equations

The second class of examples is originating from device simulation, where for instance the behavior of transistors is simulated. The simulation domain usually consists of two parts, Σ and Ω. The subdomain Σ usually represents the semiconductor region(s) (doped silicon, the wafer), in which typically three coupled equations, a Poisson-like equation and the electron and hole continuity equations, are solved for the electrostatic potential ψ and the electron and hole carrier concentrations n and p, respectively:

$$-\nabla \cdot (\epsilon_S \nabla \psi) + q(n - p - N) = 0 \,, \tag{9}$$

$$q\frac{\partial n}{\partial t} - \nabla \cdot J_n - qR_n = 0 \,, \tag{10}$$

$$q\frac{\partial p}{\partial t} + \nabla \cdot J_p - qR_p = 0 \,. \tag{11}$$

The parts J_n and J_p of the (diffusion and electric field driven) flux $J = J_n + J_p$ are given by

$$J_n = -q\mu_n n \nabla \psi + qD_n \nabla n \,, \tag{12}$$

$$J_p = -q\mu_p p \nabla \psi - qD_p \nabla p \,. \tag{13}$$

ϵ_S is the dielectric constant of the semiconductor, q the elementary charge, $N = N(x)$ the net impurity concentration, $R_n = R_n(\psi, n, p)$ and $R_p = R_p(\psi, n, p)$ the recombination-generation terms, $\mu_n = \mu_n(x, \nabla\psi, ...) > 0$ and $\mu_p = \mu_p(x, \nabla\psi, ...) > 0$ the mobilities, $D_n = \frac{k_B T}{q}\mu_n > 0$ and $D_p = \frac{k_B T}{q}\mu_p > 0$ the diffusivities, k_B the Boltzmann constant and T the device temperature. All of them are given (functions).

The second subdomain Ω, consisting of at least one region (usually an oxide), is treated as an insulator, so that (nearly) no charge carrier currents can occur. Therefore the above PDE system degenerates to Laplace's equation:

$$-\nabla \cdot (\epsilon_O \nabla\psi) = 0 \, , \tag{14}$$

where ϵ_O represents the dielectric constants of the corresponding material layers in Ω.

A detailed discussion of the properties of such systems can be found in [10, 11, 12], for instance. In the following, we concentrate on the steady-state simulation, where the time-derivatives $\frac{\partial n}{\partial t}$ and $\frac{\partial p}{\partial t}$ are vanishing. In practice, the system is discretized by a box method, employing special analytical 1D solutions along mesh edges (Scharfetter-Gummel approach [11]). The resulting system is linearized by a (modified) Newton method and usually solved by ILU-preconditioned BiCGstab. More precisely, some modified ILU(0)-approach is used.

For each of the following four different types of devices we have considered one example:

- a shallow trench isolated transistor (STI) [13],
- an electrically erasable programmable read-only memory cell (EEPROM) [13],
- a FinFET (a double-gate MOSFET structure in which a thin, fin-shaped body is straddled by the gate forming two self-aligned channels that run along the sides of the fin) [14],
- a power bipolar transistor (PBT) [13].

Details on these problems can be found in Table 1. Layouts of the STI, the FinFET and the PBT example are shown in Fig. 4, 5, and 6, respectively. For each example, a whole simulation series was run (using the commercial device simulator TAURUS by Avant! Corporation) by applying increasing voltages. All matrix equations, produced during the Newton iterations and for each voltage step, were used to test the performance of AMG.

Example	dim	ns	no	np	nv	na
STI	2D	1	7	5516	9212	125620
EEPROM	3D	1	9	10493	15415	310361
FinFET	3D	2	5	27173	42489	987123
PBT	3D	1	3	76714	149100	2908954

Table 1: Details on the four examples (dim: spatial dimension, np: number of points, ns: number of regions in Σ, no: number of regions in Ω, nv: number of variables, na: number of non-zero matrix elements).

Because of the tight coupling between the different PDEs in Σ, it turns out that unknown-based AMG fails for such applications. Instead, we have chosen a point-based AMG, in which the primary matrix P is defined by norms (3), and the same interpolation is used for all unknowns with weights being based on the entries of P.

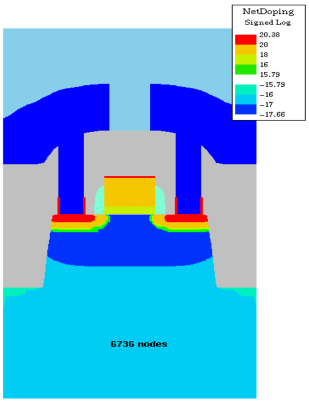

Figure 4: STI example: layout and doping profile. Courtesy of Avant! Corp.

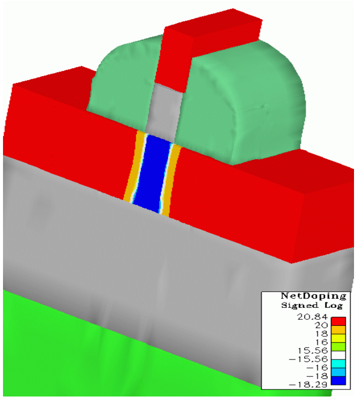

Figure 5: FinFET example: layout and doping profile. Courtesy of Avant! Corp.

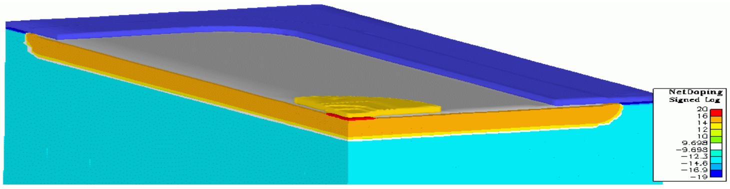

Figure 6: PBT example: layout and doping profile. Courtesy of Avant! Corp.

In Table 2, we compare the convergence behavior of this AMG approach with that of a standard ILU method of the type commonly used in device simulations (both accelerated by BiCGstab). The results represent the typical behavior of the solvers during a whole simulation run. If two exemplary results for a device (FinFET and PBT) are given, they represent the ’best’ and the ’worst’ case regarding ILU, the first of which occurs usually at an early stage of the simulation, the second near the end.

For the 2D simulations, AMG with standard coarsening gives the best convergence rates and fastest timings. Compared to ILU, AMG yields a speedup of approximately 1.5 here. In the 3D cases, it is more favorable to use AMG with *aggressive* coarsening. This reduces memory requirements, which is particularly important in 3D where relevant problems are much larger than in 2D. As can be seen in Table 2, for the largest two examples, ILU is inefficient already in the ’best’ case, and in the ’worst’ case it virtually stagnates (PBT) or even fails (FinFET). In contrast to this, AMG exhibits a stable convergence behavior in all cases and is always faster than ILU.

IV. Conclusions

AMG approaches for solving systems of PDEs were presented and discussed. Especially a general framework for point-based approaches was introduced, which employs a primary matrix to construct a point-based coarsening. Several possibilities for selecting a primary matrix and for the computation of the final interpolation weights were outlined. Recent results for applications in semiconductor process and device simulation were presented, which demonstrate that robust and fast point-based AMG methods can be obtained using this framework.

Example	Preconditioner	Cycles	ARF	Time
STI 2D	AMG	13	0.273	9.95
	ILU	44	0.670	15.74
EEPROM	aggr-AMG	12	0.303	14.32
	ILU	22	0.518	17.27
FinFET	aggr-AMG	14	0.303	87.58
	ILU	(50)[1]	0.894	(111.62)[1]
FinFET	aggr-AMG	15	0.365	136.19
	ILU	—[2]	1.05	—[2]
PBT	aggr-AMG	9	0.202	203.66
	ILU	(50)[3]	0.811	(462.27)[3]
PBT	aggr-AMG	16	0.407	269.12
	ILU	(50)[4]	0.991	(582.80)[4]

[1] residual reduction by 10^{-3} only!

[2] method diverges!

[3] residual reduction by 10^{-4} only!

[4] magnitude of residual not reduced!

Table 2: Number of cycles, average reduction factors (ARFs) and wall-clock timings (in seconds) for different examples. Accelerator always BiCGstab. Residual reduction by 10^{-6}, except stated otherwise. “aggr” means aggressive coarsening.

References

[1] J.W. Ruge and K. Stüben, *Algebraic Multigrid (AMG)*, in “Multigrid Methods”, S.F. McCormick (ed.), Frontiers in Applied Mathematics, Vol. 3, pp. 73-130, SIAM, Philadelphia, 1987.

[2] K. Stüben, *A Review of Algebraic Multigrid*, Journal of Computational and Applied Mathematics 128, pp. 281-309, 2001. Also available as GMD Report 69, November 1999.

[3] K. Stüben, *An Introduction to Algebraic Multigrid*, in “Multigrid” by U. Trottenberg, C.W. Oosterlee and A. Schüller, Academic Press, London, San Diego, pp. 413-532, 2001. Also available as GMD Report 70, November 1999.

[4] D. Oehlz, *Algebraische Mehrgittermethoden für Systeme partieller Differentialgleichungen*, Diplomarbeit, Rheinische Friedrich-Wilhelms-Universität, Bonn, April 2001.

[5] T. Füllenbach, K. Stüben and S. Mijalković, *Application of an algebraic multigrid solver to process simulation problems*, Proceedings 2000 International Conference on Simulation of Semiconductor Processes and Devices, IEEE Catalog Number: 00TH8502, ISBN 0-7803-6279-9, pp. 225-228, 2000.

[6] A. Höfler and N. Strecker, *On the Coupled Diffusion of Dopants and Silicon Point Defects*, Technical Report 94/11, Integrated Systems Laboratory, Swiss. Fed. Inst. of Technology (ETH), Zürich, 1994.

[7] A. Höfler, *Development and Application of a Model Hierarchy for Silicon Process Simulation*, PhD thesis, Swiss. Fed. Inst. of Technology (ETH), Zürich, 1997. Also available as Series in Microelectronics, vol. 69, Hartung-Gorre, Konstanz, 1997.

[8] Y. Saad, *ILUT: A dual threshold incomplete ILU factorization*, Numerical Linear Algebra with Applications 1, pp. 387-402, 1994.

[9] A. Pomp, O. Schenk and W. Fichtner, *An ILU Preconditioner Adapted to Diffusion Processes in Semiconductors*, Technical Report 99/11, Integrated Systems Laboratory, Swiss. Fed. Inst. of Technology (ETH), Zürich, 1999.

[10] S. Selberherr, *Analysis and Simulation of Semiconductor Devices*, Springer, Wien, New York, 1984.

[11] P.A. Markowich, *The Stationary Semiconductor Device Equations*, Springer, Wien, New York, 1986.

[12] P.A. Markowich, C.A. Ringhofer and C. Schmeiser, *Semiconductor Equations*, Springer, Wien, New York, 1990.

[13] S.M.. Sze (ed.), *Modern Semiconductor Device Physics*, Wiley, New York, Chichester, 1998.

[14] X. Huang et al., *Sub-50 nm P-Channel FinFET*, IEEE Transactions on Electron Devices 48, pp. 880-886, 2001.