Efficient Poisson Equation Solvers for Large Scale 3D Simulations[+]

G. Speyer, D. Vasileska and S. M. Goodnick

Department of Electrical Engineering and Center for Solid State Electronics Research
Arizona State University, Tempe, AZ, 85287-5706, USA
e-mails: speyer@asu.edu; vasilesk@imap2.asu.edu; goodnick@asu.edu

ABSTRACT

Self-consistent semiconductor device modeling requires repeated solution of the 2D or 3D Poisson equation that describes the potential profile in the device for a given charge distribution. As a result, efficient methods for the solution of 2D and 3D Poisson's equations are desired. The preconditioned bi-conjugate gradient stabilized (BiCGSTAB) method has demonstrated efficient, albeit nonmonotonic, convergence properties. It has also been shown that for a large number of points in the spatial domain, multi-grid techniques offer enhanced performance over other iterative solvers due to their multi-mode reduction of error and the fact that the codes can be easily implemented on parallel computers. In this paper, a particular combination of these methods, the multi-grid preconditioned BiCGSTAB method, is explored. Varying results on the number of grids and the level of over-relaxation of the smoothers reveal the underlying strength of the combined scheme.

The convergence properties of the developed solver are tested on a variety of device structures, including a 3D *pn*-junction diode and a split-gate silicon on insulator (SOI) device, schematically shown in Fig. 1. This later device structure is currently being fabricated within the Nanostructures Research Group at Arizona State University. The 3D Poisson equation solver is also used in examining the role of the top and bottom depletion gates on the inversion layer electron density in the channel region of the SOI device from Fig. 1.

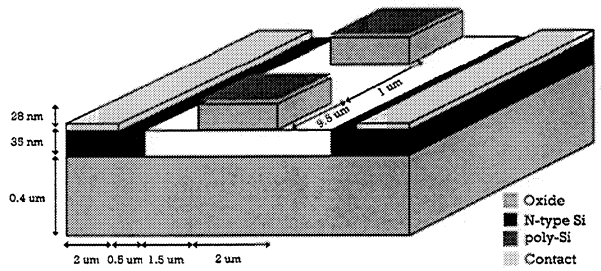

Figure 1: SOI Device structure

1 SOLVER DESCRIPTIONS

Poisson's equation can be solved by updating the potential terms in an outer loop while solving a linearized approximation for the updates in an inner loop. From the outset, initial guess will have a significant effect on the forcing function of the first few outer loop iterations. However, the inner loop solver will dominate thereafter.

Conjugate gradient methods approximate the solution in a polynomial, or Krylov subspace [1], iterating in a series of directions conjugate to the Hessian and using a minimizing line search at each step. When the method is preconditioned, the search direction has a greater chance of minimizing more toward the solution as the eigenspectrum of the Hessian is manipulated to cluster modes closer together, reducing the eccentricity of the multi-dimensional, ellipsoidal solution space. A variant of the basic method, called BiCGSTAB, can be applied to nonsymmetric matrices [2]. This method implicitly minimizes in spaces correlated to the Hessian and its transpose while maintaining orthogonality between the dual search directions. In addition, the "shadow" transpose polynomial space is designed to minimize the zigzagging convergence behavior. Nevertheless, the nonmonotonic character of the convergence often remains prominent.

The multi-grid method [3], on the other hand, solves for the error on a set of grids of increasing coarseness. As such, this method exploits the fact that the general tendency of most iterative solvers is to solve for the most oscillatory components of the error most quickly. By restricting the residuals onto coarser grids, the smooth error modes are effectively "roughened" and therefore primed for efficient reduction by a relaxation scheme such as symmetric successive over-relaxation (SSOR). However, not all error modes can be reduced with the same efficiency due to the rigid restriction of double coarsening. In addition, aliasing of oscillatory error components onto coarser grids can introduce further error.

From the above discussion it follows that using multi-grid method as a preconditioner to BiCGSTAB can offer a powerful, efficient implementation of the hybrid method. In this combined scheme, conjugate gradient methods could be expected to cull out poorly reduced error modes from multi-grid. Multi-grid, on the other hand, with its core of relaxation steps, could shape the eigenspectrum, and in its

[+] Work supported in part by NSF sponsored Descartes Center, under Contract No. ECS-9802596.

reduction across all the modes increase stability of the error norm.

In this paper, the convergence properties of the hybrid (multi-grid preconditioned conjugate gradient method) are examined and compared against standard multi-grid and BiCGSTAB methods. Object-oriented programming language (C++) is used to implement each of the solvers as member functions of the data structure. In this way, they can easily be swapped in and out transparent to the problem being solved and be easily monitored using a profiler. For multi-grid, multiple grid objects could be instantiated in a larger "Multigrid" object with the finest grid mapped to the memory space of the original problem. This data structure is schematically illustrated in Fig. 2.

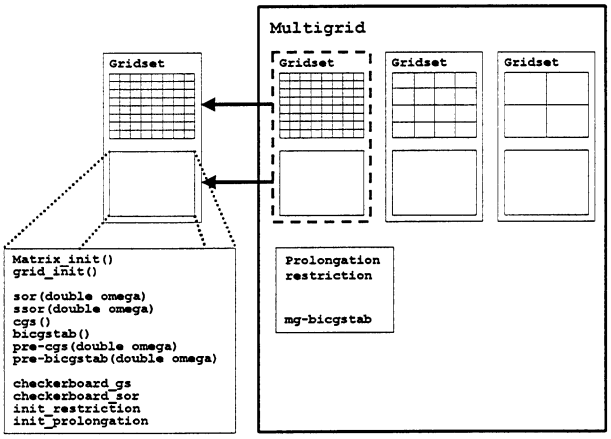

Figure 2: Data structure used in the code

2 PRECONDITIONING

The group of solvers called basic iterative methods (BIMs) can be characterized by matrix splitting, where the Hessian is split as $A = M - N$. The BIM update scheme, can be written as

$$x^{m+1} = Sx^m + M^{-1}b \ ,$$

where $S = M^{-1}N$ is called the iteration matrix. Looking at the update equation, the better M^{-1} (often called the *preconditioning* matrix) approximates the inverse of the Hessian, the less significant the efficacy of the iteration matrix becomes. One of the simpler BIMs, the Jacobi method, gives a preconditioning matrix which simply inverts the diagonal of the Hessian, i.e. assumes $M^{-1} = D^{-1}$. For SSOR, the preconditioning matrix is of the form

$$M^{-1} = (D-\omega L)\, D^{-1}\, (D-\omega L) \ ,$$

where L and U are the strictly-lower and the strictly-upper triangular parts of the Hessian and ω is the relaxation parameter (>1 for over-relaxation scheme).

In a conjugate gradient-type algorithm, preconditioning could be simply implemented by calculating $M^{-1}A$. However, this matrix multiplication can be very time-consuming, making this approach prohibitively inefficient when dealing with large-scale problems, as is usually the case in 2D and 3D device simulations needed to investigate realistic device structures. As a result of the above, in the actual implementation of the BiCGSTAB method, this multiplication is performed implicitly by preconditioning the residual-based direction vectors. This observation is more clearly seen by looking at the preconditioned BiCGSTAB algorithm given in Fig. 3 and described in more details in Ref. [4].

Let $x_0 \in \Re^n$, and set $p_0 = r_0 = b - Ax_0$, r_0^* arbitary.

For $k = 0, 1, 2, \ldots$ compute

$$\beta_k = \frac{r_0^{*\prime} r_{k+1}}{r_0^{*\prime} r_k} \times \frac{\alpha_k}{\omega_k}$$

$$p_{k+1} = r_{k+1} + \beta_k \left(p_k - \omega_k A\tilde{p}_k\right)$$

$$\tilde{p}_k = M^{-1} p_k$$

$$\alpha_k = \frac{r_0^{*\prime} r_k}{r_0^{*\prime} A\tilde{p}_k}$$

$$s_k = r_k - \alpha_k A\tilde{p}_k$$

$$\tilde{s}_k = M^{-1} s_k$$

$$\omega_k = \frac{\tilde{s}_k' A s_k}{A\tilde{s}_k' A\tilde{s}_k}$$

$$x_{k+1} = x_k + \alpha_k \tilde{p}_k + \omega_k \tilde{s}_k$$

$$r_{k+1} = s_k - \omega_k A\tilde{s}_k$$

while r_k > tolerance

Figure 3: Preconditioned BiCGSTAB algorithm [4].

The multi-grid scheme also makes use of BIMs for its relaxation steps on each grid. This is due to the quick convergence behavior on oscillatory modes characteristic of BIMs. In conjunction with the coarse-grid operators for restriction and prolongation (given as I_h^{2h} and I_{2h}^h, respectively), the smoothers reduce the oscillatory modes on each grid while minimizing the negative affect of aliasing upon further coarsening. At the coarsest level, a BIM can be used to smooth, or even attain the solution explicitly.

Therefore, the multi-grid preconditioner could be expected to include BIM preconditioning matrices in its formulation. For a two-grid scheme, this preconditioning matrix can be represented as,

$$\hat{Q} = M^{-1}S - SI_{2h}^h \left(A^{2h}\right)^{-1} I_h^{2h} \left(M^{-1}A - I\right) + M^{-1} \ ,$$

where $(A^{2h})^{-1}$ represents the coarser grid inverse Hessian [5]. In a multi-grid scheme, this would be replaced by recursive calls of the preconditioner $\hat{Q}$ at each level.

3 SOLVER RESULTS

Solvers were run on a 500 MHz DEC Alpha workstation with 1 GB RAM. The non-uniform tensor-product meshes, used for both the diode and the SOI structure, held roughly 400,000 points, or 65×145×49 nodes along the x, y and z-direction, respectively. For the 3D diode, the convergence properties of 5 different solvers were examined and the results obtained are shown in Fig. 4. The various solvers used include simple SSOR, a conventional 4 level multi-grid (MG), Jacobi preconditioned BiCGSTAB (J-BiCGSTAB), SSOR preconditioned BiCGSTAB (SSOR-BiCGSTAB) and multi-grid preconditioned BiCGSTAB (MGCG). Although there was some improvement, the additional computational overhead of the multi-grid scheme mitigated the advantage. On a parallel processing architecture, this problem could be alleviated. Jacobi preconditioned BiCGSTAB converged a bit more quickly, but the zigzagging behavior became very evident. SSOR preconditioned and multi-grid preconditioned BiCGSTAB schemes showed significant enhancement over the other solvers.

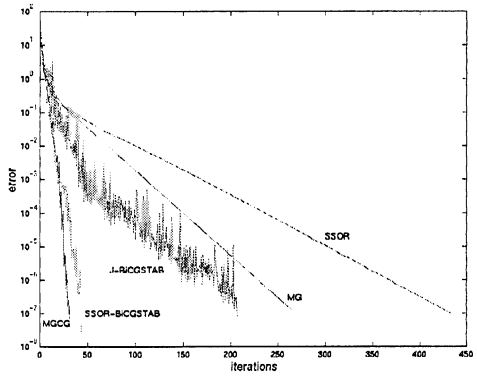

Figure 4: Error reduction for diode structure for the five different solvers examined in this work.

When a more complicated device structure, such as the SOI MOSFET structure from Fig. 1 is solved for, which exhibits nonhomogeneous dielectric permittivity and pronounced 3D features, more iterations are needed toward a given level of error convergence. This is clearly seen from the simulation results shown in Figs. 5 and 6. For example, in Fig. 5 we show the integrated charge density in the plane parallel to the top gate oxide-semiconductor interface. For this particular simulation run, the top gate bias equals V_{tg} = -6V, while the bottom gate bias is V_{bg} = 10V. The accumulation regions under the top gate are evident. The charge density in the constriction region, which shows very rapid spatial variation at the edges of the top gate gap, is an important figure of merit.

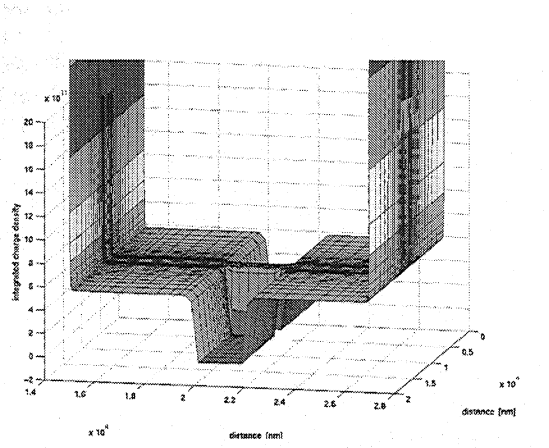

Figure 5: Integrated charge density at top interface.

In Fig. 6, we show the integrated charge density in the constriction as a function of distance from the gate edge for several top gate voltages.

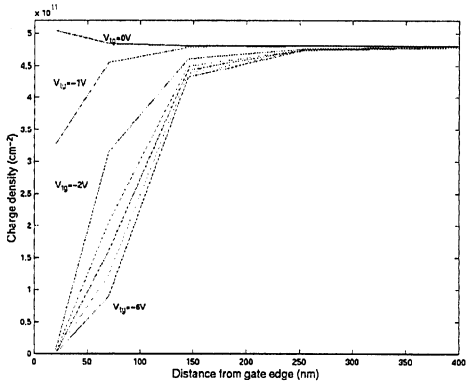

Figure 6: Integrated charge density in constriction

The convergence properties of the SSOR and the multigrid preconditioned BiCGSTAB method, for top gate V_{tg} = 0V and back-gate V_{bg} = 0V, are shown in Fig. 7.

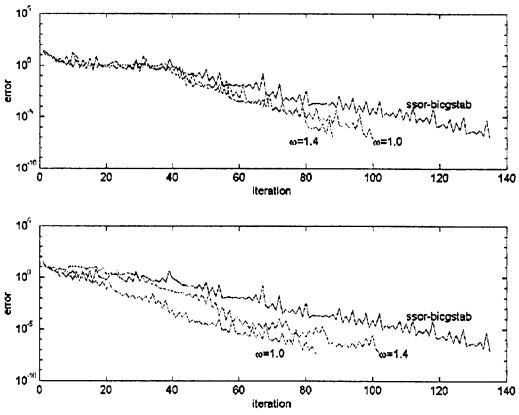

Figure 7: Error reduction for SOI MOSFET.

In the top panel of Fig. 7, two-grid multi-grid preconditioned BiCGSTAB is compared to the SSOR preconditioned BiCGSTAB. The over-relaxation constant, $\omega = 1.4$ for SSOR, is an optimum value for this problem. In the bottom panel, four grids are used in the multi-grid solver. We can see that with more grids, the error reduces more quickly and more smoothly, However, changing ω alters the smooth reduction. As evident from the results shown in Fig. 7, changing ω for the multi-grid preconditioners alters the smooth reduction.

4 CONCLUSIONS AND FUTURE WORK

In this work, we have examined the convergence properties of the hybrid, i.e. multi-grid preconditioned BiCGSTAB method (MGCG) on two different test structures: a 3D *pn*-diode and SOI device. We found out that the MGCG method shows superior convergence properties in both cases, when examined in terms of the number of iterations needed to achieve certain convergence threshold.

The results of multi-grid preconditioning of BiCGSTAB, in terms of computational demand were not presented here, but for the simulations that were run, fewer machine cycles were required than for the SSOR preconditioned scheme. This is significant because the computational demand per iteration for the multi-grid preconditioner was higher. Future work would explicitly show this trend.

REFERENCES

[1] Y. Saad, *Iterative Methods for Sparse Linear Systems*. The PWS Series in Computer Science, 1996.

[2] H. A. Van der Vorst, "Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems," *SIAM J. Sci. Stat. Comput.* 13, pp. 631-644 (1992).

[3] M. Saraniti, A. Rein, G. Zandler, P. Vogl, and P. Lugli, "An efficient multigrid poisson solver for device simulation," *IEEE Trans. CAD* 15(2), pp. 141-150 (1996).

[4] W. Pommerel, *Solution of Large Unsymmetric Systems of Linear Equations*. Konstanz:Hartungen-Gorre, 1992.

[5] O. Tatebe, "MGCG Method: A Robust and Highly Parallel Iterative Method." PhD Thesis. Univ. of Tokyo, 1996.