 CMES, vol.37, no.1, pp.45-63, 2008

Efficient Numerical Solution of the 3-D Semiconductor Poisson Equation for Monte Carlo Device Simulation

Z. Aksamija[1,2] **and U. Ravaioli**[3]

Abstract: Finding the scalar potential from the Poisson equation is a common, yet challenging problem in semiconductor modeling. One of the central problems in traditional mesh-based methods is the assignment of charge to the regular mesh imposed for the discretisation. In order to avoid this problem, we create a mesh-free algorithm which starts by assigning each mesh point to each particle present in the problem. This algorithm is based on a Fourier series expansion coupled with point matching. An efficient algorithm for repeatedly solving the Poisson problem for moving charge distributions is presented. We demonstate that this approach is accurate and capable of solving the Poisson equation on any point distribution.

Keyword: Poisson, Monte Carlo, semiconductor, meshless, spectral, collocation

1 Introduction

Numerical simulation has established itself as an indispensable tool in nanotechnology [Srivastava and Atluri (2002)] and microelectronics [Klimeck, Oyafuso, Boykin, Bowen and von Allmen (2002)]. The Poisson equation arises frequently from problems in applied physics, fluid dynamics and electrical engineering [Pini, Mazzia and Sartoretto (2008)]. Here we are primarily concerned with a problem in Electrical Engineering and Computational Electronics, and more specifically Monte Carlo simulation of semiconductor devices [Jacoboni and Reggiani (1983); Hess (2000)]. We solve the Poisson equation in order to obtain the electric potential in a semiconductor device and find the related electric field for electronic simulation. Once we find them, we let the particles move for some time interval, at the end of which we recompute the charge configuration and solve the Poisson equation again. This forms a self-consistent loop which ensures the underlying physical problem is solved accurately [Hess (1991); Duncan, Ravaioli and Jakumeit

[1] Corresponding Author: aksamija@illinois.edu
[2] Electrical and Computer Engineering, University of Illinois at Urbana/Champaign
[3] Beckman Institute for Advanced Science and Technology and University of Illinois at Urbana/Champaign

(1998)]. This setup is common in semiconductor simulation, but applies to many other problems based on particles [Hockney and Eastwood (1988)], including cosmology and gravitational problems.

Finding a suitable solution mesh is complicated by the fact that there is usually a large region at the bottom of the device which has few particles and is of little interest. On the other hand, the channel region at the top of the device is small but the detailed solution in it is important. There are also numerous problems arising from assigning charge to a regular mesh [Laux (1996)]. Therefore a regular rectangular mesh is the least desirable solution. An alternative approach is to treat each individual particle as a mesh point. Meshless methods have been proposed for the semiconductor Poisson equation in the past by Wordelman, Aluru and Ravaioli (2000) using the Finite Cloud method of Aluru and Li (2001). This method is based on reproducing kernels [Aluru (2000); Jin, Li and Aluru (2004)] and has been shown to be equivalent to a lest-squres approach [Jin, Li and Aluru (2001); Atluri and Shen (2002)]. Another method suitable for the solution of the Poisson equation is the Meshless Local Petrov-Gallerkin (MLPG) method [Atluri and Zhu (1998)], and has been shown to be especially suitable for the 3-dimensional problems by Pini, Mazzia and Sartoretto (2008). The MLPG method has also been shown to be suitable for the simulation of a variety of electrical and mechanical systems [Aluru (1999)], as well as solid mechanics [Atluri, Liu and Han (2006b)], and heat conduction [Wu, Shen and Ttao (2007)]. There is a rich and growing literature on meshless methods (see Belytschko, Krongauz, Organ, Fleming and Krysl (1996) for an overview). This work also seeks to utilize a mesh-free formulation, but in particular a formulation which can be repeatedly solved efficiently throughout the course of simulation as the simulated particles move. while being efficient for Monte Carlo simulation of particles. Fourier series have been used successfully for the formulation of a meshless method by Liu (2007) and in 3-D spectral element approaches [Wu, Al-Khoury, Kasbergen, Liu and Scarpas (2007)], but not in conjuntion with collocation. On the other hand, collocation has been used in the MLPG method in an effort to further improve the efficiency of the MLPG implementation. Atluri et al. Atluri, Liu and Han (2006a) proposed a MLPG "mixed" collocation method, in which the Dirac delta function is adopted as the test function, and therefore the system of equations is established at nodal points only. This collocation method has achieves a stable convergence rate, and it is more efficient than the MLPG finite volume method [Atluri, Han and Rajendran (2004)]. Another possible choice of expansion is that of wavelets, as proposed by Mitra and Gopalakrishnan1 (2006). Due to their compact support, wavelets are a good choice for highly localized functions, but for a smooth charge distribution, we expect the solution to also be smooth and continuous. Therefore our approach starts with a

Fourier series expansion and aims at reaching a solution which is continuous and infinitely differentiable, and optimal in the least squares sense.

2 Background

The Poisson equation is a special case of the more general Sturm-Liouville [Logan (2004)] differential equation expressed as:

$$\nabla^2 u(\mathbf{r}) + s(\mathbf{r})u(\mathbf{r}) + \lambda u(\mathbf{r}) + \rho(\mathbf{r}) = 0$$

When $s(\mathbf{r}) = 0$ and $\lambda = 0$ the equation above reduces to the standard Poisson form:

$$\nabla^2 V(x,y,z) = -\rho(x,y,z)$$

Here we use notation typical of an electrical engineering problem. $V(x,y,z)$ represents electrical potential and $\rho(x,y,z)$ represents charge. The negative sign is due to electron charge. We choose a Cartesian co-ordinate system because of the rectangular geometry which is usually encountered in Monte Carlo simulation. More generally, all devices are made up of rectangular regions so Cartesian co-ordinates are quite natural and widely applicable. The only exception are devices based on Carbon Nanotubes (CNT-FETs), which have a cylindrical geometry, but the solution presented here can easily be adapted to this configuration.

Since we have to find the charge density $\rho(x,y,z)$ at each step, choosing a sensible grid is an important decision in the process. Another related issue is that in today's ultra-scaled devices the total number of electrons present is rather small, on the scale of a thousand. Even in larger devices, it is typical to let each simulated electron represent several actual electrons, so that the number of simulated particles is reduced to a similar number. Finding a suitable solution mesh is complicated by the fact that there is usually a large region at the bottom of the device called the Substrate which has few particles and is of little interest. On the other hand, the channel region at the top of the device is small but the detailed solution in it is important. Therefore a regular rectangular mesh is the least desirable solution. More complex meshes can yield better resolution, but the problem of assigning the electron charge, which is taken to be localized, to an irregular mesh point requires an integration known as box integration. The simplest and best approach would then be to consider each individual particle (or often super-particle representing several actual particles) to be a mesh point. Because of the irregularity of this approach, using the standard finite-difference approach is not feasible.

Another issue in 3-D simulation is that of size. A typical rectangular mesh will use between 20 and 100 points in each direction, reducing the Poisson equation to a

linear problem:

$$\mathbf{A}V = \rho$$

where **A** is the banded matrix representing the Laplace operator ∇^2 by some kind of suitable finite-difference approximation, or a generalized version for non-uniform grid [Kleiber (1998)], such as that of Sukumar and Bolander (2003) based on Voronoi cells. Therefore for a typical 3-D problem, **A** is a 64000x64000 to a million by a million entry matrix. In 1-D this operator is tridiagonal, and has a particularly simple solution. In 2-D the bandwidth depends on the finite-difference stencil, but is still a sparse matrix. In 3-D the bandwidth becomes even larger and straightforward solvers become inefficient. The Poisson equation is an elliptic PDE and can be solved efficiently by transform methods, and Wavelets can give especially efficient and sparse representations, but all require regular meshes.

3 Problem Formulation

If we choose from the start to make each particle a mesh point, we are dealing with an arbitrary mesh, as the particles move along arbitrary trajectories decided by the forces of the applied electric field, and randomized by the effect of scattering interactions with phonons, impurities, and other particles. In order to seek a solution, we expand the potential into sinusoidal components [Peyret (2002)]. Since we have fixed boundary conditions (the voltage at the edges/contacts is given), we know all solutions must be a super-position of sinusoidal harmonics, so we write the solution as follows:

$$V(x,y,z) = \sum_{l=1}^{N_l}\sum_{m=1}^{N_m}\sum_{n=1}^{N_n} \Phi(l,m,n) sin\left(\frac{\pi l}{L_x}x\right) sin\left(\frac{\pi m}{L_y}y\right) sin\left(\frac{\pi n}{L_z}z\right) \tag{1}$$

where L_x, L_y and L_z are the lengths of the solution domain in each respective direction. The Laplace operator in Cartesian co-ordinates is written as:

$$\nabla^2 = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right)$$

Applying the Laplace operator to the expansion for the potential gives:

$$-\nabla^2 V(x,y,z) = \sum_{lmn} \Phi(l,m,n) \left[\left(\frac{\pi l}{L_x}\right)^2 + \left(\frac{\pi n}{L_z}\right)^2 + \left(\frac{\pi m}{L_y}\right)^2\right] sin\left(\frac{\pi l}{L_x}x\right) sin\left(\frac{\pi m}{L_y}y\right) sin\left(\frac{\pi n}{L_z}z\right) \tag{2}$$

Inserting this into the non-linear semiconductor Poisson equation,

$$-\nabla\left(\varepsilon_r(x,y,z)\cdot\nabla\Phi(x,y,z)\right)=\rho\left(\Phi(x,y,z)\right) \tag{3}$$

we obtain an algebraic expression because sinusoids are eigenfunctions of the differential operator. We will return to the issue of non-linearity. Ignoring it for now, we get:

$$\sum_{l=1}^{N_l}\sum_{m=1}^{N_m}\sum_{n=1}^{N_n}\Phi(l,m,n)\left[\left(\frac{\pi l}{L_x}\right)^2+\left(\frac{\pi n}{L_z}\right)^2+\left(\frac{\pi m}{L_y}\right)^2\right]$$
$$sin\left(\frac{\pi l}{L_x}x\right)sin\left(\frac{\pi m}{L_y}y\right)sin\left(\frac{\pi n}{L_z}z\right)=\rho(x,y,z) \tag{4}$$

Here we have cancelled the negative signs on both sides. There are some arbitrary number of particles, say P, located at positions $(X,Y,Z)_1$ through $(X,Y,Z)_P$. Including this information into the system above, we get a set of P linear equations:

$$\begin{aligned}
&\sum_{lmn}\Phi(l,m,n)\left[\left(\frac{\pi l}{L_x}\right)^2+\left(\frac{\pi n}{L_z}\right)^2+\left(\frac{\pi m}{L_y}\right)^2\right]\\
&sin\left(\frac{\pi l}{L_x}X_1\right)sin\left(\frac{\pi m}{L_y}Y_1\right)sin\left(\frac{\pi n}{L_z}Z_1\right)=\rho(X_1,Y_1,Z_1)\\
&\vdots\\
&\sum_{lmn}\Phi(l,m,n)\left[\left(\frac{\pi l}{L_x}\right)^2+\left(\frac{\pi n}{L_z}\right)^2+\left(\frac{\pi m}{L_y}\right)^2\right]\\
&sin\left(\frac{\pi l}{L_x}X_i\right)sin\left(\frac{\pi m}{L_y}Y_i\right)sin\left(\frac{\pi n}{L_z}Z_i\right)=\rho(X_i,Y_i,Z_i)\\
&\vdots\\
&\sum_{lmn}\Phi(l,m,n)\left[\left(\frac{\pi l}{L_x}\right)^2+\left(\frac{\pi n}{L_z}\right)^2+\left(\frac{\pi m}{L_y}\right)^2\right]\\
&sin\left(\frac{\pi l}{L_x}X_P\right)sin\left(\frac{\pi m}{L_y}Y_P\right)sin\left(\frac{\pi n}{L_z}Z_P\right)=\rho(X_P,Y_P,Z_P)
\end{aligned} \tag{5}$$

Choosing an ordering for l,m and n, say a lexicographical ordering, we represent this set of equations as a linear system $A\Phi=\rho$ where the matrix A absorbs all the coefficients arising from evaluating the sinusoidal factors. The entries of matrix A

will look like:

$$a_{ij} = \left[\left(\frac{\pi l}{L_x}\right)^2 + \left(\frac{\pi n}{L_z}\right)^2 + \left(\frac{\pi m}{L_y}\right)^2 \right] sin\left(\frac{\pi l}{L_x} X_i\right) sin\left(\frac{\pi m}{L_y} Y_i\right) sin\left(\frac{\pi n}{L_z} Z_i\right) \quad (6)$$

for the index j arranged so that

$$j = (l-1)N_m N_n + (m-1)N_n + n$$

The dimensions of this problem depend on the number of electrons presents, typically a thousand, and the number of spectral components we choose. Choosing more components gives a more over-determined system, while choosing fewer produces smoother solutions because how rapidly the solutions changes depends on the highest frequency present in the expansion. Therefore we have a trade-off between size, consequently speed, and precision. This issue highlights the underlying physical problem: in a modern device, in certain areas only few electrons are present and they are taken to represent a localized charge. In the classical limit, we have many particles in each mesh so we can "smear" them across the mesh, or essentially take them to represent a continuous function $\rho(x,y,z)$. But when few particles are present, we have no good way of re-assigning that charge unless we take into account the full quantum-mechanical problem and de-localize the electron charge, but that is beyond our present scope. Under the present assumption that electron charge is localized, our approach corresponds to a collocation method [Atluri (2005)].

An advantage of this approach is that the non-uniform dielectric can now be inserted and treated exactly, rather than being left to be approximated by the discretisation. Returning to (3), we note that the expression can be expanded as

$$\nabla\left(\varepsilon_r(x,y,z)\cdot\nabla\Phi(x,y,z)\right) = \nabla\varepsilon_r(x,y,z)\cdot\nabla\Phi(x,y,z) + \varepsilon_r(x,y,z)\nabla^2\Phi(x,y,z) \quad (7)$$

which allows the gradient of the dielectric constant to be applied analytically and exactly, rather than approximated by the discretisation, which is relevant at the oxide interface of modern device channels. This is possible because the gradient of the potential is available analytically as

$$\nabla\Phi(x,y,z) = \sum_{l,m,n} \Phi(l,m,n) \left[\left(\frac{\pi l}{L_x}\right) + \left(\frac{\pi m}{L_y}\right) + \left(\frac{\pi n}{L_z}\right) \right] \exp\left(\frac{i\pi l}{L_x} x\right) \exp\left(\frac{i\pi m}{L_y} y\right) \exp\left(\frac{i\pi n}{L_z} z\right)$$

The expression (refeq:field) also gives us the field analytically, as the real part of the gradient $E(x,y,z) = \Re\{\nabla\Phi(x,y,z)\}$. Given that the gradient is available analytically, Dirichelet, Neumann, and mixed boundary conditions can be also included

by augmenting the system matrix and adding a line for each boundary point. We will return to the issue of boundary conditions in more detail later. If we have some number of points where, for example, Dirichelet boundary conditions must be enforced, say N_B, located at points $(X,Y,Z)_1$, through $(X,Y,Z)_j$ up to $(X,Y,Z)_{N_B}$, then we end up again with N_B equations like

$$\sum_{l,m,n} \Phi(l,m,n) \exp\left(\frac{i\pi l}{L_x} X_j\right) \exp\left(\frac{i\pi m}{L_y} Y_j\right) \exp\left(\frac{i\pi n}{L_z} Z_j\right) = g(X_j,Y_j,Z_j) \tag{8}$$

for $j = 1,2,\ldots,N_B$, and $g(x,y,z)$ the value at these boundary points. The augmented system can now be solved to find a solution which includes boundary conditions. Now we return to the issue of non-linear systems. The nonlinearity present in (3) can be treated by the usual Newton-Raphson approach [Wordelman, Aluru and Ravaioli (2000)]. This sets up an outer iteration where, at each step, we solve a linearized system by the method given herein.

The total system of equations is now posed as a full matrix of size $(N_P + N_B) \times (N_l N_m N_n)$. Because of this, we can now apply any of the standard tools to solve it. The system can be solved by standard lest-squares techniques or by fast iterative methods [Watkins (2002)]. If we choose fewer spectral components and make the system nearly square, then the size of the problem is greatly reduced from its finite-difference counterpart, while reflecting the underlying physics in a more meaningful way. The computational complexity can be reduced significantly by using the non-equispaced Fast Fourier Transform (NFFT) [Pöplau and Potts (1994)]. This approach produces a solution which is $\mathscr{O}(N)\log(N)$, which is much more efficient than the cubic time required for the direct solution of a full matrix system.

In any case, the matrix A is a full matrix of size $P \times (N_l N_m N_n)$. It requires a regularization approach to the solution. Because of this, we can now pose the problem as a linear system and apply any of the standard tools to solve it. If we choose fewer spectral components and make the system nearly square, then the size of the problem is greatly reduced from its finite-difference counterpart, while reflecting the underlying physics in a more meaningful way.

4 Efficient Implementation

So far our problem has been posed as a dense problem to be solved by a regularization approach. Typically this will be a least squares method which, although implemented in various ways, always can be considered in terms of the singular value decomposition. Such approaches produce excellent solutions, but can be inefficient and costly for large dense systems like ours. Therefore we have to seek a more efficient implementation. One possible path is to consider the fact that, in a

typical Monte Carlo simulation, the Poisson equation is solved at each step, or every few steps, and the total number of steps, or simulation iterations, is very large, on the order of a hundred thousand. Each step represents a time interval of a duration often less then a femto-second. The drift velocity of electrons in silicon saturates at about $10^7 \frac{m}{s}$ so the distance traveled by an electron between two solutions of the Poisson equation can be very small, on the order of a nanometer. Therefore we can apply perturbation analysis to try to come up with a more efficient solution which will avoid a complete SVD (Singular Value Decomposition) at each step.

Let us denote the position of particle i at iteration k as (X_i^k, Y_i^k, Z_i^k). Then the position at iteration $k+1$ will be

$$(X_i^{k+1}, Y_i^{k+1}, Z_i^{k+1}) = (X_i^k + \delta X_i^k, Y_i^k + \delta Y_i^k, Z_i^k + \delta Z_i^k)$$

The only position dependence in the problem are the sinusoidal components in the matrix A, so we can analyze them as

$$\begin{aligned} sin(x+\delta x) &= sin(x)cos(\delta x) + cos(x)sin(\delta x) \\ &\approx sin(x) + \delta x cos(x) \end{aligned}$$

Here we've used the first terms in the Taylor series expansion of sine and cosine to obtain a first order approximation. There are terms in x,y and z, so we proceed as follows

$$\begin{aligned} &(sin(x)cos(\delta x))(sin(y)cos(\delta y))(sin(z)cos(\delta z)) \\ &\quad \approx sin(x)sin(y)sin(z) + \delta x cos(x)sin(y)sin(z) \\ &\quad + \delta y sin(x)cos(y)sin(z) + \delta z sin(x)sin(y)cos(z) \end{aligned}$$

In this derivation we dropped all terms that had two or three δ terms because they are assumed to be small. We can now write the entries of the matrix A at the $k+1$ instant as

$$\begin{aligned} a_{ij}^{k+1} &= \left[\left(\frac{\pi l}{L_x}\right)^2 + \left(\frac{\pi n}{L_z}\right)^2 + \left(\frac{\pi m}{L_y}\right)^2 \right] sin\left(\frac{\pi l}{L_x} X_i^k\right) sin\left(\frac{\pi m}{L_y} Y_i^k\right) sin\left(\frac{\pi n}{L_z} Z_i^k\right) \\ &\quad + \frac{\pi l}{L_x} \delta X_i^{k+1} cos\left(\frac{\pi l}{L_x} X_i^k\right) sin\left(\frac{\pi m}{L_y} Y_i^k\right) sin\left(\frac{\pi n}{L_z} Z_i^k\right) \\ &\quad + sin\left(\frac{\pi l}{L_x} X_i^k\right) \frac{\pi m}{L_y} \delta Y_i^{k+1} cos\left(\frac{\pi m}{L_y} Y_i^k\right) sin\left(\frac{\pi n}{L_z} Z_i^k\right) \\ &\quad + sin\left(\frac{\pi l}{L_x} X_i^k\right) sin\left(\frac{\pi m}{L_y} Y_i^k\right) \frac{\pi n}{L_z} \delta Z_i^{k+1} cos\left(\frac{\pi n}{L_z} Z_i^k\right) \end{aligned} \tag{9}$$

Then the complete matrix at the $k+1$ instant can be written in a decomposed form as

$$A^{k+1} = A^k + \delta A_x^{k+1} + \delta A_y^{k+1} + \delta A_z^{k+1}$$

where the δ terms represent the respective portions of the matrix as expressed above. Then the entire Poisson system can be written as

$$A^{k+1}\Phi^{k+1} = \rho$$
$$\left(A^k + \delta A_x^{k+1} + \delta A_y^{k+1} + \delta A_z^{k+1}\right)(\Phi + \delta\Phi) = \rho$$

Then we cancel the terms that have two δ variations.

$$A^k\Phi^k + A^k\delta\Phi^{k+1} + \left(A^k + \delta A_x^{k+1} + \delta A_y^{k+1} + \delta A_z^{k+1}\right)\Phi^k = \rho$$

Making use of the previous solution

$$A^k\Phi^k = \rho$$

and the fact that the charge density is just a constant because only position of the electrons changes and not their individual charge, the above simplifies to

$$A^k\delta\Phi^{k+1} = -\left(A^k + \delta A_x^{k+1} + \delta A_y^{k+1} + \delta A_z^{k+1}\right)\Phi^k$$

We can, for simplicity, denote the right-hand-side of the above by a variable

$$A^k\delta\Phi^{k+1} = -\gamma^{k+1}$$

Assuming we have a decomposition for the matrix A as

$$A^k = U_k\Sigma^k V_k^T$$

where Σ is a diagonal matrix of singular values. We can form the solution using this decomposition

$$\delta\Phi^{k+1} = V_k^T diag\{\frac{-\gamma_1^{k+1}}{\sigma_1^k}, \ldots, \frac{-\gamma_P^{k+1}}{\sigma_P^k}\}U_k$$

Adding together the previous solution and the update gives

$$\Phi^{k+1} = \Phi^k + \delta\Phi^{k+1} = V_k^T diag\{\frac{\rho_1 - \gamma_1^{k+1}}{\sigma_1^k}, \ldots, \frac{\rho_P - \gamma_P^{k+1}}{\sigma_P^k}\}U_k$$

Therefore we can solve the problem at each step using the previous decomposition, as long as the overall motion of electrons in small compared to the wavelength of the highest spectral component. All that is required at each step is the knowledge of the positions of the particles. The amount of work required to form the update

scales as $P \times (N_l N_m N_n)$, which is the amount of work required just to write the entries of the matrix, and cannot be improved upon by any method.

As we iterate and the electrons move, the error made using this approach will first show up in the highest frequencies, and thus will be small. Then it is possible to correct the solution every few iteration by re-computing the decomposition and starting over with the updates. This way our solution will not incur the large cost of a de-composition at every iteration and will be more efficient by a constant factor depending on how many iterations we allow between decompositions.

Other efficient approaches are also possible. For example, it is efficient to solve the problem by an iterative method without resorting to a decomposition at all. Then, because the change in the problem matrix A is small, the solution will not vary greatly and we can expect fast convergence at each iteration using the previous solution as a starting point.

Another obvious choice that can offer savings over a full SVD is the LU factorization. The LU factorization can be applied to arbitrary matrices, such as the one in this problem, and used to create a solution more efficiently. The matrices L and U are lower and uppertriangular, respectively. If we factor the system matrix as

$$A\Phi = LU\Phi = \rho$$

then we can form the solution can be formed by solving the simpler uppertriangular problem by back substitution

$$\Phi = U^{-1}L^T\rho$$

Efficiency can still be improved by keeping the same LU factorization for several iterations, and only solving the update system at each step

$$LU\,\delta\Phi^{k+1} = -\gamma^{k+1}$$

$$\delta\Phi^{k+1} = -U^{-1}L^T\gamma^{k+1}$$

The cost of back substitution is only quadratic while the LU factorization is still cubic, but less costly than a full SVD decomposition. Therefore, combined with the fact that the full decomposition is not necessary at each iteration, the solution can be very efficient.

5 Cost Analysis

We can partition the cost of this algorithm into a few sections. First we have the cost of forming the problem at the onset. This involves making the matrix A based

on the list of electron positions. In the rest of this analysis we shall assume, for the sake of simplicity, that $N_l = N_m = N_n = N$ as there is no apparent reason to make a distinction in the number of spectral components by direction. Then the matrix size will be $P \times N^3$. Each entry in the matrix requires a total of 27 flops (floating point operations) so the cost is $27PN^3$. There is no cost associated with ρ because this is just a constant given by electron charge and dielectric constant. In the case where the dielectric constant changes throughout the device, ρ may depend on the position, but even then it is constant throughout different regions of the device.

Next we have a similar analysis for the cost of actually forming the solution for the potential. The cost of this is 18 flops per harmonic so there is a total of $18N^3$ for each point where we want to find out the potential. Similar is the cost of computing updates at each step between re-computing a full decomposition. This has a cost of $30PN^3$ flops in each direction plus an addition and a matrix-vector multiplication so the total is $92PN^3$ flops. The constant factor here can be reduced by re-cycling some intermediate values at a cost of higher storage requirements.

Finally, when we decide to do a full de-composition, this will incur a large cost penalty. The cost is cubic in the number of singular values, but this is where the strength of this method lies. The number of singular values is P, equal to the number of electrons in the system. As we said before, this may be as small as a thousand for a typical simulation, so the total cost may be acceptable in comparison to a more traditional finite-difference method which may be of a lower order, but not in the number of electrons P, but in the total number of mesh points, which exceeds N^3. Therefore this approach can be very competitive in terms of computational cost, while offering a solution which avoids costly and un-physical charge assignments. There may well be additional properties of the system which allow a more efficient solution due to the properties of the sine and cosine functions, but they are not at this point obvious.

6 Conditioning

Since the number of electrons in the system is not necessarily equal to the total number of harmonics $P \neq N^3$, we are faced with a non-square system. If we have more points than harmonics, meaning $P > N^3$, then the system is overdetermined, and we must seek a solution which minimizes the norm of the error in some norm, typically the ℓ_2 norm

$$\Phi = \arg\min ||A\Phi - \rho||_2^2$$

Otherwise, if the number of harmonics is larger than the number of points, or equivalently electrons, we have $P < N^3$, and the system is underdetermined, and there is a

space of possible solutions which all satisfy the given equality. Then we must seek a solution which satisfies additional constraints, such as, for example, a minimum energy solution

$$\Phi = \arg\min ||\Phi||_2^2 \quad s.t \quad A\Phi = \rho$$

The decision to be made here is how many harmonics to take in each orthogonal direction, in other words, the values of N_l, N_m, and N_n. Conditioning of the inverse problem depends heavily on the number of harmonics. Experience has shown that the condition number increases with the number of harmonics. When N^3 exceeds P and the system becomes underdetermined. Finding a solution then turns into a more complicated search that demands additional knowledge of the nature of the solution, which may not be available. On the other hand, as the problem becomes more overdetermined, finding a solution gets easier and the condition number decreases, thereby increasing our ability to select the best solution. Therefore it was decided to keep the number of harmonics less than the number of electrons. This is easily done in practice as the number of harmonics N is a free parameter, unlike the number of electrons P. This also allows us to reduce the size of the problem arbitrarily and trade off precision for speed even further. Reducing N creates smoother solutions, as mentioned before, but also makes the cost of a decomposition smaller.

7 Boundary Conditions

The issue of boundary conditions is now a simple one to solve in the present setup. The potential can be equated to a given value at any number of boundary points. In fact, the points do not have to be at the boundary, although they typically are. This is formulated at the i'th point as

$$V_B(X_i,Y_i,Z_i) = \sum_{l=1}^{N_l}\sum_{m=1}^{N_m}\sum_{n=1}^{N_n}\Phi(l,m,n)sin\left(\frac{\pi l}{L_x}X_i\right)sin\left(\frac{\pi m}{L_y}Y_i\right)sin\left(\frac{\pi n}{L_z}Z_i\right) \tag{10}$$

Since sinusoids are always zero at the boundary, we must make the formulation more general and extend it to arbitrary and complex values. Therefore the sines have to be replaced by complex exponentials, which is equivalent to combining both sine and cosine solutions. Then the system reads as

$$V_B(X_i,Y_i,Z_i) = \sum_{l=1}^{N_l}\sum_{m=1}^{N_m}\sum_{n=1}^{N_n}\Phi(l,m,n)exp\left(\frac{i\pi l}{L_x}X_i\right)exp\left(\frac{i\pi m}{L_y}Y_i\right)exp\left(\frac{i\pi n}{L_z}Z_i\right) \tag{11}$$

For multiple boundary points a system of equations is again obtained

$$B\Phi = V_B$$

where the entries of the matrix are given by

$$b_{ij} = \Phi(l,m,n) exp\left(\frac{i\pi l}{L_x} X_i\right) exp\left(\frac{i\pi m}{L_y} Y_i\right) exp\left(\frac{i\pi n}{L_z} Z_i\right)$$

as before. Now the complete system can be solved for the potential.

Solutions for the potential can also be obtained at any arbitrary point, and not just at those locations where the electrons are. For some arbitrary point, we have

$$V(X_S,Y_S,Z_S) = \sum_{l=1}^{N_l} \sum_{m=1}^{N_m} \sum_{n=1}^{N_n} \Phi(l,m,n) exp\left(\frac{i\pi l}{L_x} X_S\right) exp\left(\frac{i\pi m}{L_y} Y_S\right) exp\left(\frac{i\pi n}{L_z} Z_S\right) \quad (12)$$

The problem of determining the potential from a charge distribution on a arbitrary mesh is now complete.

8 Implementation Details and Results

A test was performed by solving the system above for a rectangular region with zero boundary conditions and uniformly randomly distributed charges. We compare the solutions obtained by a finite difference discretization with the sine series expansion system solved with an LU decomposition. The two solutions and their running times were compared for a total of 1000 uniformly distributed points in a 3-D region of dimensions $L_x = L_y = L_z = 100$. Dielectric constant was taken to be 1, and so was the electron charge. This only affects the scaling of the problem and does not affect the shape or properties of the solution at all. The number of harmonic components was taken to be $N_x = N_y = N_z = 8$, while the finite difference solutions was computed on a $50x50x20$ grid in space. The total cpu time required to solve the finite difference system using an efficient conjugate gradients iteration with a relative residual set to 10^{-6} was around 2.7 seconds. For comparison, the LU factorization and backsubstitution took only 0.17 seconds. The second approach was more than an order of magnitude faster. This is due to a relatively small number of harmonics used, dictated by the total number of points. The sine series solution is still more appealing than finite differences because the finite difference solution suffers from problems arising from the charge assignment scheme. This can be improved upon by better cloud-in-cell (CIC) charge assignments [Wordelman and Ravaioli (2000)], but it remains always a problem due to the unphysical need to discretize the charge positions. Finally, the Poisson equation in this example case has a constant on the right-hand side. The 3-D results in Fig. 1 shows good smoothness and precision. We contrast this to the usual approach based on finite differences (FD) and a cloud-in-cell (CIC) scheme for assigning charge to the rectangular mesh imposed by the finite difference discretisation [Hockney and

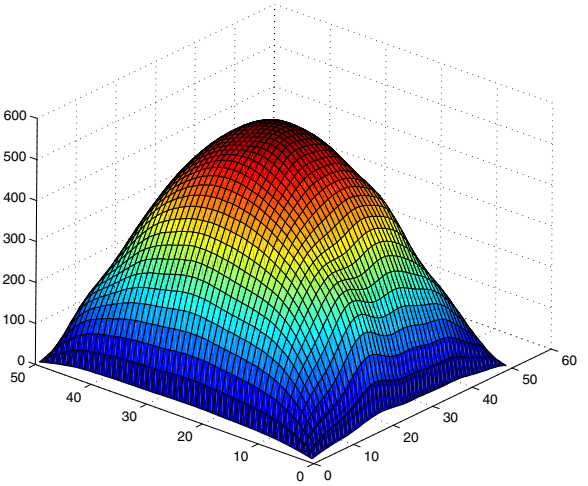

Figure 1: Solution obtained from the meshless approach. Only a horizontal cut through the 3-D computational domain is shown.

Eastwood (1988)]. The FD-CIC solution for the same uniform random distribution of charges is shown in Fig. 2. We demonstrate that the FD-CIC approach, while being capable of representing potentials due to uniform distributions well, fails in representing non-uniform clumps of charge. The meshless solution, on the other hand, is based on global smooth functions which do not suffer from such problems.

The Fig.3 compares the numerical solution to the analytical solution we would expect from a uniform charge distribution. Integrating a constant charge distribution twice and applying the boundary conditions produces a quadratic analytical solution. This gives a simple solution

$$V(x,y,z) = (x-50)^2 + (y-50)^2 + (z-50)^2 + C$$

where the constant C is used to match boundary conditions. The shape of the sine series solution is a very good match to the expected parabolic solution, unlike the finite difference solution, which exhibits many irregularities, as shown by the dashed line in Fig.3. This is further proof of the usefulness of meshless approaches to the solution of the Poisson equation. From this comparison we conclude that algorithm presented herein is capable of great accuracy when representing charges as mesh points. We can also note from Fig. 4 that the resulting system is well-conditioned and therefore poses no challenge to solve.

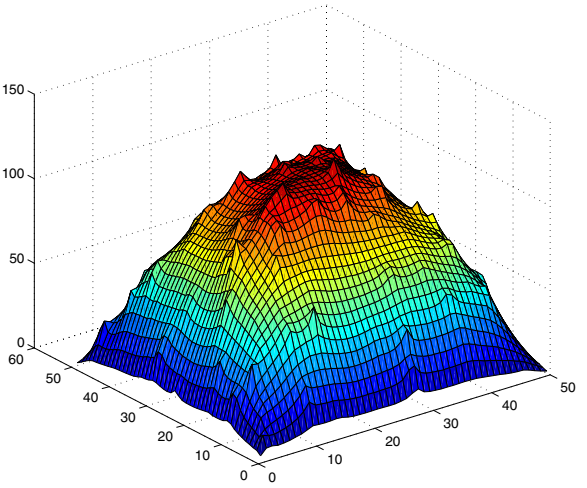

Figure 2: Solution obtained from finite differences with a cloud-in-cell (CIC) charge assignment. Only a horizontal cut through the 3-D computational domain is shown.

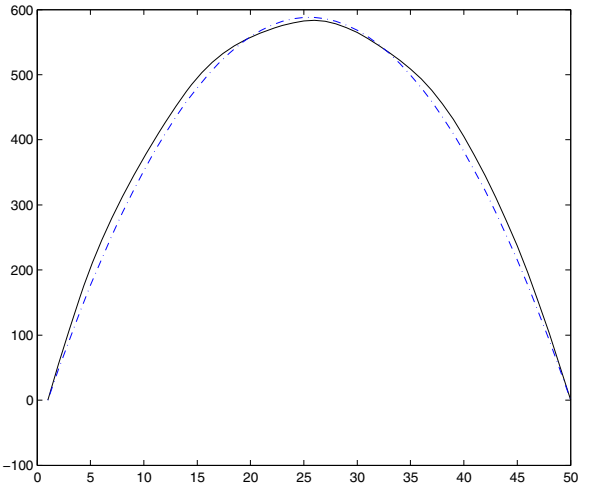

Figure 3: Comparison of solutions. The solid line is the numerically computed solution, and the dashed line is the analytical quadratic solution which assumes a uniform distribution. The agreement is excellent and the small discrepancy is due to the random assignment of charge locations.

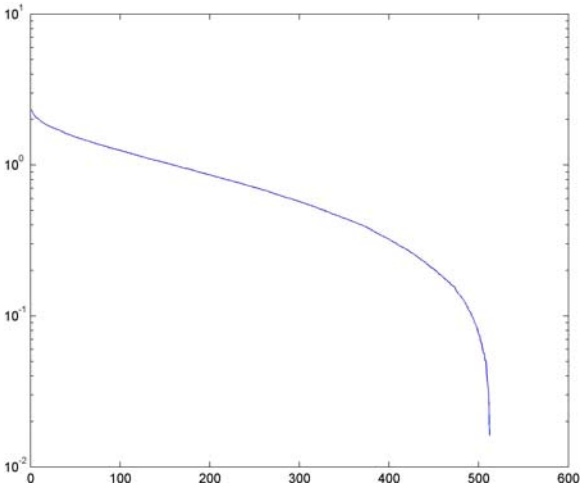

Figure 4: Logarithmic plot of the singular values of the matrix A. The ratio of the largest to the smallest singular value is around 100, meaning that the solution is sufficiently well conditioned. Experience has shown that increasing the number of harmonics N beyond the number of points P drastically increases the condition number and makes the solution ill-conditioned.

9 Conclusions

We have demonstrated the feasibility and efficiency of a spectral collocation approach for the non-linear semiconductor Poisson problem. One issue is deciding how many spectral components N to choose. This can be tailored to the particular application and may well depend on the actual problem. This will also determine the size of the problem and an appropriate method for its solution. We can also choose how long to run the efficient update scheme between re-computing a full decomposition. Fewer iterations between re-computing a full decomposition means we have a more efficient algorithm, but we sacrifice some accuracy. Both of these parameters can be tailored to the problem at hand, thus allowing for a very efficient implementation of the semiconductor Poisson problem.

Acknowledgement: This work was supported by the Department of Energy Computational Science Graduate Fellowship Program of the Office of Science and National Nuclear Security Administration in the Department of Energy under contract DE-FG02-97ER25308.

References

Aluru, N. R. (1999): A reproducing kernel particle method for meshless analysis of microelectromechanical systems. *Computational Mechanics*, vol. 23, no. 1, pp. 324–338.

Aluru, N. R. (2000): A point collocation method based on reproducing kernel approximations. *International Journal for Numerical Methods in Engineering*, vol. 47, no. 3, pp. 1083–1121.

Aluru, N. R.; Li, G. (2001): Finite cloud method: A true meshless technique based on a fixed reproducing kernel approximation. *International Journal for Numerical Methods in Engineering*, vol. 50, no. 3, pp. 2373–2410.

Atluri, S. N. (2005): *Methods of Computer Modeling in Engineering & the Sciences*. Forsyth, Tech Science Press.

Atluri, S. N.; Han, Z. D.; Rajendran, A. M. (2004): A new implementation of the meshless finite volume method, through the mlpg "mixed" approach. *CMES-Computer Modeling in Engineering & the Sciences*, vol. 6, no. 6, pp. 491–514.

Atluri, S. N.; Liu, H. T.; Han, Z. D. (2006): Meshless local petrov-galerkin (mlpg) mixed collocation method for elasticity problems. *CMES-Computer Modeling in Engineering & the Sciences*, vol. 14, no. 3, pp. 141–152.

Atluri, S. N.; Liu, H. T.; Han, Z. D. (2006): Meshless local petrov-galerkin (mlpg) mixed finite differencemethod for solid mechanics. *CMES-Computer Modeling in Engineering & the Sciences*, vol. 15, no. 1, pp. 1–16.

Atluri, S. N.; Shen, S. (2002): The meshless local petrol-gallerkin (mlpg) method: A simple and less-costly alternative to the finite element and boundary element methods. *CMES-Computer Modeling in Engineering & the Sciences*, vol. 3, no. 1, pp. 11–51.

Atluri, S. N.; Zhu, T. (1998): A new meshless local petrol-gallerkin (mlpg) approach in computational mechanics. *Computational Mechanics*, vol. 22, no. 1, pp. 117–127.

Belytschko, T.; Krongauz, Y.; Organ, D.; Fleming, M.; Krysl, P. (1996): Meshless methods: An overview of recent developments. *Computational Methods in Applied Mechanical Engineering*, vol. 139, no. 1, pp. 3–47.

Duncan, A.; Ravaioli, U.; Jakumeit, J. (1998): Full-band monte carlo investigation of hot carrier trends in the scaling of metal-oxide-semiconductor field-effect transistors. *IEEE Transactions on Electron Devices*, vol. 45, no. 4, pp. 867–876.

Hess, K. (1991): *Monte Carlo Device Simulation: Full Band and Beyond*. Kluwer Academic Press, Boston.

Hess, K. (2000): *Advanced Theory of Semiconductor Devices*. IEEE Press, New York.

Hockney, R. W.; Eastwood, J. W. (1988): *Computer Simulation using Particles*. Adam Hilger, New York.

Jacoboni, C.; Reggiani, L. (1983): The Monte Carlo method for the solution of charge transport in semiconductors with applications to covalent materials. *Reviews of Modern Physics*, vol. 55, no. 3, pp. 645–705.

Jin, X.; Li, G.; Aluru, N. R. (2001): On the equivalence between least-squares and kernel approximations in meshless methods. *CMES-Computer Modeling in Engineering & the Sciences*, vol. 2, no. 4, pp. 447–462.

Jin, X.; Li, G.; Aluru, N. R. (2004): Positivity conditions in meshless collocation methods. *Computational Methods in Applied Mechanical Engineering*, vol. 193, no. 3, pp. 1171–1202.

Kleiber, M.(Ed): *Handbook of Computational Solid Mechanics*. Springer-Verlag, Berlin, 1998.

Klimeck, G.; Oyafuso, F.; Boykin, T. B.; Bowen, R. C.; von Allmen, P. (2002): Development of a nanoelectronic 3-d (nemo 3-d) simulator for multimillion atom simulations and its application to alloyed quantum dots. *CMES-Computer Modeling in Engineering & the Sciences*, vol. 3, no. 5, pp. 601–642.

Laux, S. E. (1996): On particle-mesh coupling in monte carlo semiconductor device simulation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, pp. 10.

Liu, C.-S. (2007): A mriem for solving the laplace equation in the doubly-connected domain. *CMES-Computer Modeling in Engineering & the Sciences*, vol. 19, no. 2, pp. 145–161.

Logan, J. D. (2004): *Applied Partial Differential Equations*. New York, Springer.

Mitra, M.; Gopalakrishnan1, S. (2006): Wavelet based 2-d spectral finite element formulation forwave propagation analysis in isotropic plates. *CMES-Computer Modeling in Engineering & the Sciences*, vol. 15, no. 1, pp. 49–67.

Peyret, R. (2002): *Spectral Methods for Incompressible Viscous Flow*. New York, Springer.

Pini, G.; Mazzia, A.; Sartoretto, F. (2008): Accurate mlpg solution of 3d potential problems. *CMES-Computer Modeling in Engineering & the Sciences*, vol. 36, no. 1, pp. 43–64.

Pöplau, G.; Potts, D. (1994): Fast poisson solvers on nonequispaced grids: Multigrid and fourier methods compared. In Luk, F. T.(Ed): *Proceedings of SPIE: Advanced Signal Processing Algorithms, Architectures and Implementations XIII*, pg. 5205.

Srivastava, D.; Atluri, S. N. (2002): Computational nanotechnology: A current perspective. *CMES-Computer Modeling in Engineering & the Sciences*, vol. 3, no. 5, pp. 531–538.

Sukumar, N.; Bolander, J. E. (2003): Numerical computation of discrete differential operators on non-uniform grids. *CMES-Computer Modeling in Engineering & the Sciences*, vol. 4, no. 6, pp. 691–705.

Watkins, D. S. (2002): *Fundamentals of Matrix Computations*. John Wiley and Sons, Inc., New York.

Wordelman, C. J.; Aluru, N. R.; Ravaioli, U. (2000): A meshless method for the numerical solution of the 2- and 3-d semiconductor poisson equation. *CMES-Computer Modeling in Engineering & the Sciences*, vol. 1, no. 1, pp. 121–126.

Wordelman, C. J.; Ravaioli, U. (2000): Integration of a particle-particle-particle-mesh algorithm with te ensamble monte carlo method for the simulation of ultra-small semiconductor devices. *IEEE Transactions on Electron Devices*, vol. 47, pp. 410.

Wu, C.-Y.; Al-Khoury, R.; Kasbergen, C.; Liu, X.-Y.; Scarpas, A. (2007): Spectral element approach for inverse models of 3d layered pavement. *CMES-Computer Modeling in Engineering & the Sciences*, vol. 17, no. 3, pp. 163–172.

Wu, X.; Shen, S.; Tao, W. (2007): Meshless local petrov-galerkin collocation method for two-dimensional heat conduction problems. *CMES-Computer Modeling in Engineering & the Sciences*, vol. 22, no. 1, pp. 65–76.