Chapter 3 Effective Mass Approximation

Chapter 3 .1 Background

The area of mesoscopic physics is described by Landauer theory where the coherence length is greater than the relevant device region and transport is ballistic. All inelastic collisions occur outside the device region in ideal reservoirs. This is distinctly unlike Boltzman transport which is based on a series of scattering events that randomize the phase of wave functions and in which carrier wavelengths are much smaller than the device region. Ballistic transport occurs in small systems at high purity and particularly at low temperature.

The device structures described in the previous section are governed by a range of device models including quantum tunneling and interference, space charge, band mixing, and scattering. Design of these devices is dependent on the ability to simulate quantum tunneling and interference. This may be done using a ballistic transport model based on the time independent Schrödinger's equation with the effective mass approximation, for instance, to get steady state solutions. In the effective mass approximation, single band and valley, small wave number, and small spatial derivatives are assumed. When done self consistently with Poisson, space charge effects are included as well. This is a single or independent electron approximation to the many body problem assuming carrier-carrier interactions are insignificant. Steady state solutions determined by this method assume elastic transport with no scattering. These simulations tell us something about the devices while still requiring interpretation.

In some two dimensional (2D) devices behavior is controlled by ballistic transport in one direction and by Boltzman transport in the other direction. As an approximation, along the direction controlled by ballistic transport a series of one dimensional (1D) Schrödinger Poisson solutions may be determined and used in the Boltzman transport problem in the other direction. In some cases this is a poor approximation.

Ideally the space modeled should be subdivided into regions governed by different transport models. One region may be modeled by Boltzman transport and a second embedded region by ballistic transport. The boundary between two regions may be described in terms of potential, electron and hole concentration, and carrier flow. These quantities and their derivatives should be continuous across the boundary between model regions. Where the continuity equation is used to determine concentration and potential profiles, current through the ballistic region is required to describe the boundary. In addition, the boundary location may be restricted by other assumptions. As a result solutions in the two regions must be solved iteratively. In any case it is advantageous for the 2D Schrödinger Poisson solver to be as efficient as possible.

Chapter 3 .2 Green's Function

The causal surface Green's function method (CSGFM) [26], developed by Keldysh for systems far from equilibrium, determines the causal response of a system to injection of an electron at one surface and extraction from another. It only requires the Green's function be known at a surface enclosing the desired region. Knowledge of the Green's function at one surface is sufficient to calculate, by recursion, the Green's function at any other surface that encloses it. The recursion is unstable at some energies and for long models.[26] The Green's function is determined in terms of a Hamiltonian that must be separable with determinable eigenvalues and eigenvectors over the portion of the device that confines carriers. Solution generally is by a recursion relation which has a sublinear calculation time with the number of nodes (N) for 1D (one dimensional) simulations which compares favorably with N^3 inversion time. The terminology is valid whether recursion or another method is used to obtain a solution. Green's functions are particularly useful in the area of mesoscopic devices[27].

Chapter 3 .3 Time Independent Effective Mass Equation

The Hamiltonian is based on the effective mass equation, derived from Schrödinger's equation by considering only a single band and only energies near minimum k (wavenumber) where the spatial derivatives are small. The time independent effective mass equation is

$$-\frac{\hbar^2}{2m*}\nabla^2 F(x) + E_c F(x) = E \cdot F(x) \qquad \text{(Chapter 3 .1)}$$

where F(x) is the envelope function, E_c is the conduction band offset, E is the energy, and m^* is the electron effective mass[28]. If tight binding is assumed, discretization of the Hamiltonian may be done using only nearest neighbor nodes. A plane wave assumption gives the dispersion relation

$$E(k_x, k_y, k_z) = \frac{-\hbar^2}{m*}\left\{\left(\frac{\cos(k_x\Delta x)-1}{\Delta x^2}\right)+\left(\frac{\cos(k_y\Delta y)-1}{\Delta y^2}\right)+\left(\frac{\cos(k_z\Delta z-1}{\Delta z^2}\right)\right\}+E_c \qquad \text{(Chapter 3 .2)}$$

in three dimensions, where k_x, k_y, and k_z are wavenumbers and Δx, Δy and Δz are node spacing in the coordinate directions. This approximates the pseudopotential GaAs bandstructures[29] and is referred to as the simplest form of tight binding[30]. The 1D discretized Hamiltonian matrix is tridiagonal, the 2D matrix is pentadiagonal (five diagonals), and three dimensional (3D) matrix has seven diagonals.

Chapter 3 .4 2D Discretization

To model a 2D device with N_z nodes by N_y nodes, or $N=N_y*N_z$ total nodes, a N by N matrix is constructed. The coefficients of this matrix fall along 5 diagonals in this matrix. This space is discretized in y and z using the scheme shown in Figure Chapter 3 .1

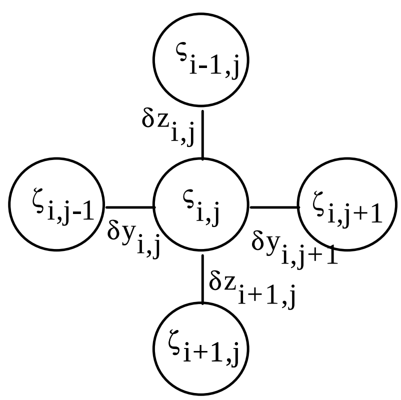

Figure Chapter 3 .1: This is the two dimensional discretization scheme. δz and δy are node spacings in z and y, respectively. The model space is indexed in i along z and j along j. $\zeta_{i,j}$ is a solution at the node location (i,j).

These diagonals can be given by the r, a, d, c, and l in the equations

$$\beta = \frac{1}{m^{*2}_{i,j}} \cdot \left[\left(\frac{m^*_{i,j} - m^*_{i+1,j}}{\delta\, z_{i+1,j}} \right) + \left(\frac{m^*_{i,j} - m^*_{i,j+11}}{\delta\, y_{i,j+1}} \right) \right], \qquad \text{(Chapter 3 .3)}$$

$$d_k = \beta \cdot \left(\frac{1}{\delta\, z_{i+1,j}} + \frac{1}{\delta\, y_{i,j+1}} \right) - \frac{2}{m*_{i,j}} \cdot \left[\frac{1}{\delta\, z_{i,j} \cdot \delta\, z_{i+1,j}} + \frac{1}{\delta\, y_{i,j} \cdot \delta\, y_{i+1,j}} \right] + E_c - E \quad \text{,(Chapter 3 .4)}$$

$$a_k = \frac{2}{m*_{i,j}} \cdot \left(\frac{1}{\delta\, z_{i+1,j} \cdot \left(\delta\, z_{i,j} + \delta\, z_{i+1,j} \right)} \right), \qquad \text{(Chapter 3 .5)}$$

$$c_{k-1} = \frac{2}{m^*_{i,j}} \cdot \left(\frac{1}{\delta\, z_{i+1,j} \cdot \left(\delta\, z_{i,j} + \delta\, z_{i+1,j} \right)} \right) - \frac{\beta}{\delta\, z_{i+1,j}}, \qquad \text{(Chapter 3 .6)}$$

$$l_{k-Nz} = \frac{2}{m^*_{i,j}} \cdot \left(\frac{1}{\delta\, z_{i,j} \cdot \left(\delta\, y_{i,j} + \delta\, y_{i,j+1} \right)} \right) - \frac{\beta}{\delta\, y_{i,j+1}}, \qquad \text{(Chapter 3 .7)}$$

and

$$u_k = \frac{2}{m^*_{i,j}} \cdot \left(\frac{1}{\delta\, y_{i,j+1} \cdot \left(\delta\, y_{i,j} + \delta\, y_{i,j+1} \right)} \right), \qquad \text{(Chapter 3 .8)}$$

where k = i + j*N_z. The resulting Hamiltonian matrix is not diagonally dominant for arbitrary values E. It is shown in equation (Chapter 3 **.9)**.

$$H = \begin{bmatrix} d_1 & a_1 & & u_1 & & & \\ c_1 & d_2 & a_2 & & . & & \\ & c_2 & d_3 & . & & . & \\ l_1 & & . & . & . & & u_{N-Nz} \\ & . & & . & . & . & \\ & & . & & . & . & a_{N-1} \\ & & & l_{N-Nz} & & c_{N-1} & d_N \end{bmatrix} \qquad \text{(Chapter 3 .9)}$$

Chapter 3 .5 Homogeneous Solution

Since the matrix is generally symmetric and its elements are real, the matrix is both Hermitian and normal. As a result it may be assumed that the eigenvalues will be real and that there is an orthogonal set of eigenvectors that span the space of the matrix[31].

The Lanczos algorithm may be used to tridiagonalize the homogeneous problem[32] which is pentadiagonal in the 2D case. It finds a similar tridiagonal matrix which can then be solved for its eigenvalues. Though the similar matrix should be full rank, the terms are in order of those associated with the most to least dominant eigenvalues. The algorithm ends when calculated off diagonal elements are zero. Because of round off errors an off diagonal zero element does not always occur so it is difficult to get a complete set of eigenvalues. If the m most dominant eigenvalues are desired then the algorithm may be terminated when a tridiagonal matrix of rank m has been created. The eigenvalues generated from this reduced tridiagonal matrix approximate the actual eigenvalues[29].

A Lanczos based algorithm is good for finding extremal eigenvalues. It might not be appropriate for finding complete sets of eigenvalues. For sparse matrices the algorithm should require about (2k+8) • n flops per iteration where k is the number of non-zero elements per row of length n. In the pentadiagonal case this is 18 • n floating point operations. Though it is not as stable as Householders method[33], Lanczos is advantageous for sparse matrices because it does not require significant storage or decrease the sparseness of the matrix as Householders method does. Householders method is used on dense matrices where this is not a problem. There may, however, be a loss of orthogonality among Lanczos vectors. To solve this problem Lanczos vectors may be re-orthogonalized at significant computational cost.

The symmetric Lanczos algorithm is shown in Figure Chapter 3 .2.

$r_0 = q_1$ (1

$q_0 = 0$ (2

$\beta_0 = 1$ (3

$j = 0$ (4

while($\beta_j \neq 0$) (5

  $q_{j+1} = r_j/\beta_j$ (6

  j++ (7

  $\alpha_j = q_j^T A q_j$ (8

  $r_j = (A-\alpha_j I)q_j - \beta_{j-1} q_{j-1}$ (9

  $\beta_j = ||r_j||_2$ (10

end (11

Figure Chapter 3 .2: This is the symmetric Lanczos algorithm[34].

In this algorithm A is the matrix, r are the residual vectors, q are the normalized residual or Lanczos vectors, and α is an estimate of the eigenvalue by the Rayleigh coefficient and the diagonal elements in the tridiagonal output matrix. Each residual vector is linearly independent of all preceding residual vectors. The normalization coefficient β forms the off diagonal elements. If the rank is n, then there will be n β, and n α, coefficients as well as n q vectors, unless the algorithm is intentionally stopped at m elements, as suggested earlier. The first terms α, and β are associated with the dominant eigenvalue.

The residual vector calculation (line 9 of the algorithm in Figure Chapter 3 .2) is an Arnoldi process given by[32]

$$q_{k+1} = \frac{1}{\beta_{k+1,k}} \left(Aq_k - \sum_{j=0}^{k} q_j \beta_{j,k} \right), \qquad \text{(Chapter 3 .10)}$$

where

$$\beta_{j,k} = \left\langle Aq_k, q_j \right\rangle, \qquad \text{(Chapter 3 .11)}$$

and q are orthonormal vectors forming a Krylov basis, and elements β form a Hessemberg matrix B. For the case where Arnoldi is applied to a self adjoint matrix, the Arnoldi matrix B is tridiagonal, which is the only symmetric Hessemberg matrix. Equation (Chapter 3 .10) then becomes

$$q_{k+1}\beta_{k+1,k} = Aq_k - q_k \alpha_{k,k} - q_{k-1}\beta_{k-1,k}. \qquad \text{(Chapter 3 .12)}$$

The eigenvalues are ordered such that

$$\lambda_1 \rangle \lambda_2 \ldots \rangle \lambda_i \rangle \ldots \lambda_n. \qquad \text{(Chapter 3 .13)}$$

In this problem the smallest eigenvalues, not the largest or dominant eigenvalues, are needed. For a large matrix round off errors effect the accuracy of the small eigenvalues calculated later in the algorithm. Alternatively, two methods may be used to determine the small eigenvalues. The matrix may be inverted or shifted by the dominant eigenvalue. The eigenvalues of the inverted matrix are the inverted eigenvalues of the original matrix and so the order is reversed. The eigenvalues of the shifted matrix are then the sum of the original eigenvalue and the shift. Shifting the matrix is less computationally expensive than the inversion but the condition of the matrix may be, and usually is, adversely effected. LU factorization may be used rather than an inverse. Calculation of bound states is not the most computationally expensive part of the program, so extra time spent here may not significantly effect the overall run times of a simulation. Accuracy and efficiency of the results may be significantly affected when matrices become poorly conditioned. LU factorization is generally used rather than a matrix shift.

Since only the first m dominant eigenvalues of the inverse matrix are desired, the order of the tridiagonal matrix is less than the original matrix. This order reduced tridiagonal matrix may be used to determine the dominant eigenvalues of the inverse matrix. There are two similar algorithms that may be used to do this step. One is QR factorization. This can be used if the tridiagonal matrix is symmetric, which is true if the original matrix was symmetric. The second is LR factorization, which is used if the tridiagonal matrix is not assumed to be symmetric. Generally QR factorization is more stable but LR factorization is used because of the potential application to asymmetric matrices[29]. Appendix A shows the generalized LR algorithm.

Determinants of large matrices are difficult to evaluate and change rapidly for small changes in a shift of the diagonal that is near an eigenvalue. The Lanczos/LR algorithms give approximate eigenvalue locations. The accuracy of these eigenvalues may then be improved by the secant method. In the vicinity of the eigenvalue the estimated determinant is minimized. Calculation of the determinant is generally done by LU factoring the matrix and then taking the product of the diagonal terms. For large matrices this product generally causes overflows, so the log of each diagonal element is summed instead. This method does not work well for very large matrices because for a given eigenvalue there may be many large diagonal LU elements and one small one. Because of machine precision the product may be a poor function of this one small diagonal element. The minimum LU diagonal element may be used instead of the determinant for refinement. This process is illustrated in Figure Chapter 3 .3.

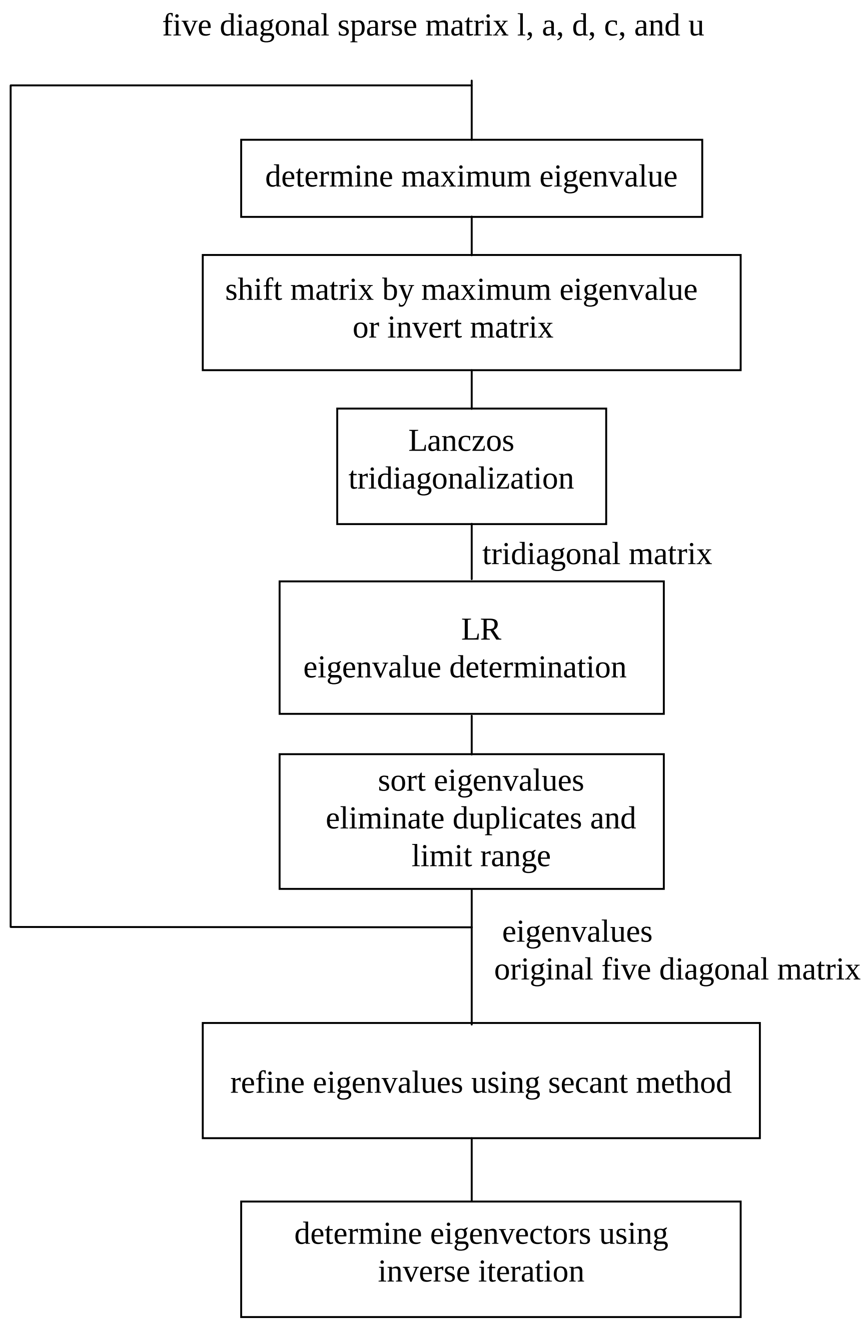

Figure Chapter 3 .3: This is a flow chart of the process used to determine eigenvalue and eigenvectors.

Inverse iteration is generally used to refine eigenvalues and calculate the eigenvectors. Because care has been taken to get good eigenvalues, eigenvectors may be generated using inverse iteration without updating eigenvalues. Inverse iteration may be described by

$$\left(A-\lambda_k I\right)\cdot y=b_k \qquad \text{(Chapter 3 .14)}$$

where A is the matrix, λ_k is the eigenvalue, y is the trial eigenvector, and b_{k-1} replaces it on each iteration. If eigenvalue λ_k improvement is also desired, it is iterated after convergence of the eigenvector stops. Eigenvectors need only be calculated for quasi bound state energies. The quasi bound state energies are those states associated with eigenvalues that are located below the conduction band energy at the contacts or ends of the device[35]. Based on the eigenvectors at eigenvalues identifying bound state energies, the electron concentration in those bound states may be calculated. The algorithm shown in Figure Chapter 3 .3 is applied as a test to a matrix where it may be compared to a reference[36].

Chapter 3 .6 Inhomogeneous Solution

To solve an open system the inhomogeneous or traveling wave solution must be considered. To do this the model space may be divided into three regions. Regions one and three are semi infinite boundary regions containing plane waves sandwiching region two which is the device region described by the homogeneous equations. The solutions to the inhomogeneous problem are unbound. Incident, reflected, and transmitted plane waves in the boundary regions are coupled to the device region by the boundary conditions. Schrödinger's equation is

$$(H-E)|G\rangle = 0\,, \qquad \text{(Chapter 3 .15)}$$

where G is a Green's function, H is the Hamiltonian, and E is the energy. Assuming a 1D model for simplicity for the discretized Hamiltonian described in section 3.4, the equation at the first node, numbered zero, is

$$H_{0,-1}G_{-1} + H_{0,0}G_0 + H_{0.1}G_1 = 0\,, \qquad \text{(Chapter 3 .16)}$$

where G_{-1} may then be determined by

$$G_{-1} = -H_{0,-1}{}^{-1}\left(H_{0,0}G_0 + H_{0,1}G_1\right). \qquad \text{(Chapter 3 .17)}$$

The wavefunctions inside the device are coupled to the incident, reflected, and transmitted waves outside the device by[37]

$$\begin{aligned} &G_{-1} = I + r \\ &G_1 = I \cdot e^{-ik_\perp d} + r \cdot e^{-ik_\perp d} \\ &\bullet \\ &\bullet \\ &\bullet \\ &G_N = t \cdot e^{ikd} + r' \cdot e^{-ikd} \\ &G_{N+1} = t + r' \end{aligned}\,, \qquad \text{(Chapter 3 .18)}$$

where nodes -1 and N+1 and corresponding equations are the boundary conditions added to nodes 0 through N and corresponding equations describing the device region. These boundary layers are chosen such that no reflection occurs at the interface between boundary layers and the device region. Solving for the incident and reflection coefficient equations in terms of the Green's function solution at nodes gives

$$I = \frac{e^{ikd}}{e^{ikd} - e^{-ikd}} G_{-1} - \frac{1}{e^{ikd} - e^{-ikd}} G_0 \text{, and} \qquad \text{(Chapter 3 .19)}$$

$$r' = 0 = \frac{-1}{e^{ikd} - e^{-kd}} G_n + \frac{e^{ikd}}{e^{ikd} - e^{-ikd}} G_{n+1} \text{,} \qquad \text{(Chapter 3 .20)}$$

where the reflection coefficient r' back into the device from infinite boundary region III, may be assumed to be zero.

The transmission coefficient may also be related to the Green's functions by

$$\tau = \frac{e^{ikd}}{e^{ikd} - e^{-kd}} G_n - \frac{1}{e^{ikd} - e^{-ikd}} G_{n+1}. \qquad \text{(Chapter 3 .21)}$$

These coefficients are generally complex. The resulting matrix is asymmetric complex and rank N+2. The wavefunction solutions to this problem may be used to calculate electron concentrations and currents.

In the 2D case this same concept may be applied. The relationship between the incident and reflected plane waves in the boundary regions and wavefunction node values at the boundary of the device region is

$$\begin{bmatrix} G_{i,j-1} \\ G_{i,j} \\ G_{i+1,j} \\ G_{i,j+1} \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{\lambda_y} & 1 & \lambda_y \\ 1 & 1 & 1 & 1 \\ \lambda_z & 1 & \frac{1}{\lambda_z} & 1 \\ 1 & \lambda_y & 1 & \frac{1}{\lambda_y} \end{bmatrix} \cdot \begin{bmatrix} I_z \\ I_y \\ r_z \\ r_y \end{bmatrix} \qquad \text{(Chapter 3 .22)}$$

where λ_z is $e^{ik_z d}$ and λ_y is $e^{ik_y d}$. Inverting this matrix and using $I_z = 1$ an equation relating the Green's function at these nodes may be written. The equation

$$
\begin{aligned}
I_z = & \frac{-\lambda_y}{\left(\lambda_y{}^2\lambda_z - 2\lambda_y\lambda_z + \lambda_y{}^2 - 2\lambda_y + \lambda_z + 1\right)} G_{n+1-Nz} + \\
& \frac{\lambda_z}{\left(\lambda_z{}^2 - 1\right)} G_n + \\
& \frac{-\left(-2\lambda_y\lambda_z + \lambda_z{}^2 + 1\right)}{\left(-2\lambda_y\lambda_z{}^2 + \lambda_y{}^2\lambda_z{}^2 - 2\lambda_y + \lambda_z{}^2 - 1\right)} G_{n+1} + \\
& \frac{-\lambda_y}{\left(\lambda_y{}^2\lambda_z - 2\lambda_y\lambda_z + \lambda_y{}^2 - 2\lambda_y + \lambda_z + 1\right)} G_{n+1-Nz}
\end{aligned}
\qquad \text{(Chapter 3 .23)}
$$

relates I_z to the wavefunction solution. Similar equations may be written at each node along y and z boundaries for I_z, I_y, r_z, and r_y. This adds $2\bullet N_y + 2\bullet N_z$ equations. For the top layer

$$
l \cdot G_{n-Nz} + c \cdot G_{n-1} + d \cdot G_n + a \cdot G_{n+1} + u \cdot G_{n+Nz} = 0, \qquad \text{(Chapter 3 .24)}
$$

where l, c, d, a, and u are discretization coefficients. Here an incident plane wave propagating along the z axis may be assumed. This can be used to write the matrix equation

$$
\begin{pmatrix} G_{n-1} \\ G_n \end{pmatrix} = \begin{bmatrix} -{(d+l+u)}/{c} & -{a}/{c} \\ 1 & 0 \end{bmatrix} \begin{pmatrix} G_n \\ G_{n+1} \end{pmatrix}, \qquad \text{(Chapter 3 .25)}
$$

which may be rewritten

$$\begin{pmatrix} G_n \\ G_{n+1} \end{pmatrix} = \begin{bmatrix} -{(d+l+u)}\big/{c} & -{a}\big/{c} \\ 1 & 0 \end{bmatrix} \begin{pmatrix} G_n \\ G_{n+1} \end{pmatrix} e^{-ik_z d} \qquad \text{(Chapter 3 .26)}$$

using the tight binding assumption[37]. The eigenvalues of this matrix are $e^{ik_z d}$ (λ_z) and $e^{-ik_z d}$ ($1/\lambda_z$) in equation (Chapter 3 .22). Similar equations may be written at boundaries for waves propagating in the y direction.

The resulting matrix contains boundary equations along the z and y boundaries. Solution of these matrix equations may be done by LU decomposition or by iterative methods. For sparse matrices iterative methods such as Conjugate gradients have advantages. The most stable iterative methods require symmetric matrices that are diagonally dominant. Strict diagonal dominance requires

$$a_{i,i} \rangle \sum_j a_{i,j}\, ,\ j \neq i \qquad \text{(Chapter 3 .27)}$$

for the diagonal and off diagonal terms. With (Chapter 3 .24) this becomes

$$\begin{aligned} & d > a + c + u + l \\ & E_c - E > 0 \end{aligned} \; . \qquad \text{(Chapter 3 .28)}$$

So for E >E_c the matrix is not strictly diagonally dominant and iterative packages may not converge or converge only slowly. Conjugate gradient algorithms that work on asymmetric matrices have been developed, for instance, by solving a matrix of the form A^TA which is symmetric but more poorly conditioned than A. A preconditioned conjugate gradient algorithm PCGSTAB was tested for this problem [38]. The preconditioning is done by incomplete LU factorization. In order to calculate concentrations and current densities integration is done over the energy spectrum so that a solution is desired at each energy in the integration. These are solved in order of monotonically increasing energy so that the last solution is a good starting point for the next iteration. The solution at a given energy should be a small perturbation of the solution at the previous energy assuming small energy steps. The problem was sufficiently poorly conditioned that the PCGSTAB algorithm did not converge, or converged slowly, and the accuracy of the final solution was poor. ITPACK algorithms were also tried with similar results[39].

The LU factorization provided in Sparse [40] was much faster and more accurate for all matrix sizes tried. The Sparse data structure is an orthogonal link list with the element structure shown in Figure Chapter 3 .4. The density fraction is given by

$$\frac{5-{}^{4}\!/\!_{N_y}}{N}. \qquad \text{(Chapter 3 .29)}$$

This is small for large matrices. The process of LU decomposition typically causes growth in the density of a few percent. The data structure is justified when the density is less than 50%.

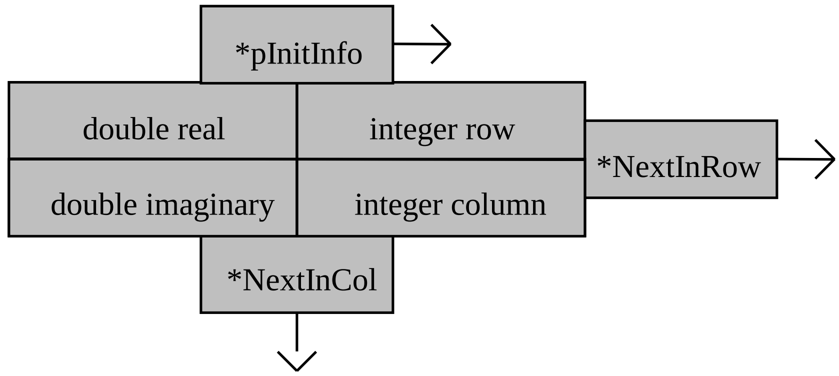

Figure Chapter 3 .4: This is the sparse matrix element structure.

This element structure contains double word complex data, integer row and column numbers, a pointer *pInitInfo to an initialization vector, a pointer *NextInRow to the next row element, and a pointer *NextInCol to the next column element. There are at least eight words dedicated to each stored element, which is equivalent to two complex double words. In addition to pointer arrays pointing to the first row and column elements, there is a pointer array pointing to diagonal elements. Fill-ins with this structure are created during the LU factorization increasing matrix density.

Integration of the energy spectrum to determine concentration and transmission coefficients uses Gaussian Quadrature coefficients based on a fit to the data. At any point in a device structure there are peaks and nodes due to quantum interference effects. Where these peaks and nodes are sharp it is important to integrate this portion of the spectrum carefully. Failure to integrate a peak accurately causes too little concentration to be calculated for this resonance, affecting a specific region of the device. Failure to integrate a node accurately causes too much concentration to be calculated in the interference node. As shown in Figure Chapter 3 .5 there are peaks and nodes due to quantum interference. Near the ends this is due to waves reflected back from the barriers reinforcing and canceling with incident waves creating a predictable position dependent pattern of peaks and nodes. There are also peaks in the transmission spectrum due to the resonance in the heterostructure quantum well corresponding to peaks in the wave function solution, as well in the concentration. All other locations in the model demonstrate a node at this energy. These patterns are determined to optimize the integration by scanning the spectrum previous to integration.

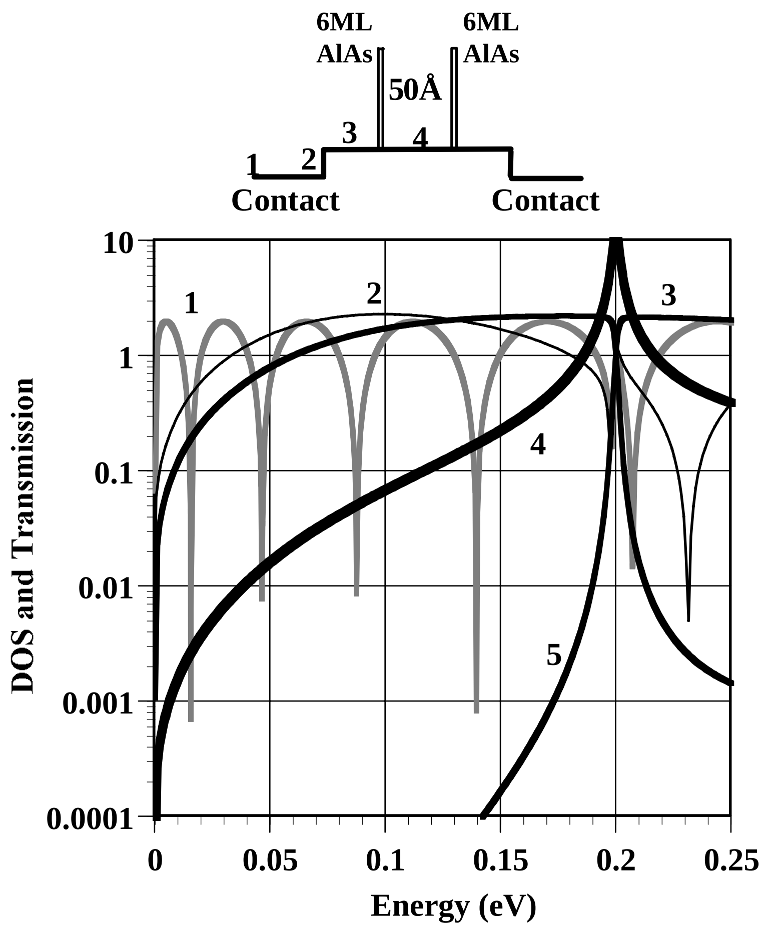

Figure Chapter 3 .5: This is the density of states (DOS) and transmission coefficint spectrum (τ) at several locations in the DBRTD (Double Barrier Resonant Tunneling Diode) device shown above the graph. Curve 1 corresponds the beginning of the device at the contact, curve 2 corresponds to the end of the N+ region, curve 3 corresponds to the N- region adjacent to the barrier and curve 4 corresponds to the heterostructure quantum well. Note that the transmission coefficient in curve 5 peaks at about 0.2 eV. This coincides with the peak in the DOS spectrum of curve 4 which is the heterostructure quantum well. All other curves show a minimum at this energy indicating the electron lifetime is small except in the well. The other maxima and minima particularly in curve 1 are due to interference between incident wave and the wave reflected from the barrier. Here DOS is defined as G*G.

Chapter 3 .7 Concentration Calculation

Concentration may be calculated from the wave function or eigenvector solution as described in sections 3.5 and 3.6. Assuming 2D density of states concentration may be calculated by

$$C_i^e = \frac{m^* kT}{2\pi^2 \hbar^2} \int_{\varepsilon}^{\infty} \frac{\partial k}{\partial \xi} d\xi \cdot \log\left(1 + e^{\left(E_F - \xi\right)/kT}\right) \left|G_i(\varepsilon)\right|^2 , \qquad \text{(Chapter 3 .30)}$$

based on the dispersion relation[41]

(Chapter 3 .2)

$$\frac{\partial k}{\partial \xi} = \frac{1}{\left[2 \cdot \varepsilon\,(k) - \frac{m^* a^2 \cdot \varepsilon\,(k)^2}{\hbar^2}\right]^{1/2}} . \qquad \text{(Chapter 3 .31)}$$

Assuming a 1D density of states the concentration is given by

$$C_i^e = \frac{1}{2\pi^2} \sqrt{\frac{2m^*}{\hbar^2}} \int_0^{\infty} k \frac{\partial k \left|G_i(\varepsilon)\right|^2}{\partial \xi} d\xi \int_{\xi}^{\infty} \frac{d\xi}{\sqrt{E}\left(1 + e^{\left(E - E_F\right)/kT}\right)} , \qquad \text{(Chapter 3 .32)}$$

where

$$k \frac{\partial k}{\partial \xi} = \frac{\cos^{-1}\left(1 - \frac{m^* a^2 \xi\,(k)}{\hbar^2}\right)}{a\left[2 \cdot \varepsilon\,(k) - \frac{m^* a^2 \cdot \varepsilon\,(k)^2}{\hbar^2}\right]^{1/2}} . \qquad \text{(Chapter 3 .33)}$$

Since the second integral in equation (Chapter 3 .32) has no closed form it is interpolated from a table of values generated numerically. These equations are derived in Appendix A.

The concentration integral is evaluated iteratively until solution of Poisson's equation does not change significantly. Several options are available in aiding convergence. Aitken acceleration[33] may be used to improve convergence speed. To prevent rapid divergence starting from a potential function far from the correct solution the change in concentration and potential may be limited in each iteration. As the solution converges the potential change and space charge may oscillate between positive and negative values. When this oscillation is detected, a projected solution is determined by bisection of previous solutions.

At equilibrium the global space charge is zero. Figure Chapter 3 .6 is a plot of the space charge error versus iteration number for a self-consistent solution. The space charge error converges to about 10^{15} cm^{-3} per cell. This is about 0.025% of the maximum concentration in the model. This corresponds to an error in ΣG_i of about $4x10^{-7}$. The maximum potential difference is about 10^{-6} eV or about 0.0015% error. Convergence requires about 10 iterations.

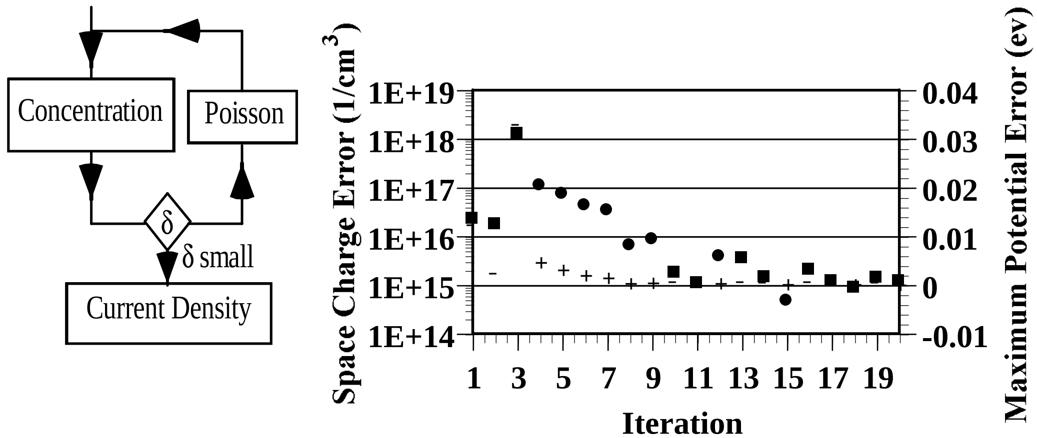

Figure Chapter 3 .6: On the left is a self-consistent solver flow chart and on the right is an illustration of the convergence of the space charge and maximum potential update versus iteration. Positive space charge errors are symbolized by boxes and negative errors by circles. The + and - symbols show maximum potential update on each iteration. Note that after about 10 iterations the space charge error is $\pm 10^{15}$ and the potential update is near zero. In each case there is some oscillation between negative and positive values.

Chapter 3 .8 Current Calculation

The current may be calculated using the equation

$$J = \int_0^\infty dk_l \int_0^\infty dk_t [\, f(E) - f(E')] \left(\tau_{E,V}{}^* \tau_{E,V} \right), \qquad \text{(Chapter 3 .34)}$$

where J is the longitudinal current, k_l is the wavenumber in the longitudinal direction, k_t is the wavenumber in the transverse direction, f(E) is the Fermi-Dirac probability, and τ is the transmission coefficient.

In the 1D quantization, or confinement, case assuming parabolic dispersion relation this equation becomes

$$J = \frac{qm^* kT}{2\pi^2\hbar^3}\int_0^{\infty} dE_z \ln\left(\frac{1+e^{\left(E_F - E_L/kT\right)}}{1+e^{\left(E_F - E_R/kT\right)}}\right)\left(\tau_{E,V}{}^{*}\tau_{E,V}\right). \qquad \text{(Chapter 3 .35)}$$

There are two methods of determining the transmission coefficients. In the first method the inhomogeneous problem is first solved and the τ is calculated using (Chapter 3 .21).

The second method of determining the transmission coefficient is to determine the eigenvalues of the inhomogeneous matrix as a function of energy[29]. This can be used to determine the total transmission spectrum. The inhomogeneous matrix itself changes with energy. At small energy increments eigenvalues are determined using the process detailed in Figure Chapter 3 .3. This requires the asymmetric matrix version of the Lanczos algorithm shown in Appendix B. Eigenvalue problems of asymmetric matrices cannot be treated as generally as symmetric matrices[34]. The matrix may be defective so that there is no complete set of eigenvalues and/or the matrix is sensitive to small changes so that eigenvalues cannot be determined because numerical round off changes the answer significantly.

For the 2D quantization case this becomes

$$\mathrm{J} = \frac{\mathrm{qm}^*}{\mathrm{p}^2\hbar^3}\int_0^{\infty} \mathrm{dE_l} \int_{\mathrm{E_t}}^{\mathrm{E_b}} \sqrt{\frac{\mathrm{E_l}}{\mathrm{E_t}}} \cdot \frac{\mathrm{dE_t}}{1+\mathrm{e}^{(\mathrm{E_f}-\mathrm{E_l}-\mathrm{E_t})/\mathrm{kT}}}\left(\tau_{\mathrm{E,V}}{}^{*}\tau_{\mathrm{E,V}}\right) \qquad \text{(Chapter 3 .36)}$$

where E_l is the energy due to longitudinal transport and E_t is due to transverse transport. This equation permits motion in y and z directions. The transmission coefficient $\tau_{E,V}$ is the transmission coefficient at energy E and bias V as determined by (Chapter 3 .21) for the 1D case and (Chapter 3 .22) for 2D. In principle, it should be possible to determine the transmission spectrum in the 2D case as well from eigenvalue solutions as described by Bowen for 1D[29]. It is not clear how this would be implemented.

Since the plane wave assumption should be violated in the device a 2D current calculation may be made as derived in Appendix C.

$$\overline{J} = \frac{q\sqrt{m_o}}{\hbar^2} \cdot \frac{i\sqrt{m^*}}{2\sqrt{2}\pi^2} \int_0^\infty dE_l \int_{E_1}^{E_2} \frac{dE_T}{\sqrt{E_T}} \frac{\left(G^* \overline{\nabla} G - G \overline{\nabla} G^*\right)}{\left(1 + e^{\left(E_l + E_T - E_F\right)/kT}\right)} \quad \text{(Chapter 3 .37)}$$

Chapter 3 .9 Tests of the Algorithm

Basic structures for which the results are known will be used to demonstrate the performance of the algorithm. Because of shorter run times and simplicity 1D simulations may be used to demonstrate general properties of the simulations.

Chapter 3 .9.1 One Dimensional Simulation

Results of simulations of increasing complexity will be shown. The simplest structure is a length of uniform material. The wavefunctions for this structure are unity and the results are completely controlled by the Fermi level. A short uniform structure may be used to determine these Fermi levels. For reference the Fermi level for a donor concentration of $4.0\text{x}10^{18}$ cm^{-3} is 0.133 eV.

A Single Barrier Diode (SBD), a Double Barrier Resonant Tunneling Diode (DBRTD), a Triple Barrier Resonant Tunneling Diode (TBRTD), a Modulation Doped Field Effect Transistor (MODFET) and other structures are modeled with this method. These results will be shown for comparison in Chapters 4 and 5.

Chapter 3 .9.2 Two Dimensional Simulation

Two dimensional Schrödinger Poisson simulations have been used to simulate quantum wires and other low dimensional structures. Run times are generally long and are dominated by the time required to solve the discretized inhomogeneous Schrödinger equation. A polynomial fit between integration time of Equation 12, which is dominated by the matrix solution time, and device size results in the third order polynomial

$$time = x0 + x1 \cdot n + x2 \cdot n^2 + x3 \cdot n^3, \qquad \text{(Chapter 3 .38)}$$

where x_0=-1.18, x_1=0.243, x_2=-0.009, x_3=$3.16 \bullet 10^{-4}$, n is the square root of the number of nodes in the device, and time is the number of seconds to do an 18 point gaussian quadrature at 100 points. Since the size of the discretization matrix goes as the square of the device size the time of the matrix solution is order 3/2 to matrix rank.

To illustrate this algorithm two DBRTD device models are shown. In this case a 1.0 eV barrier is used to simulate Fermi level pinning on the physical boundaries of the device and the air interface beyond it. Where the device is very wide the results are very similar to running independent simulations at intervals across the device as shown in Figure Chapter 3 .7.

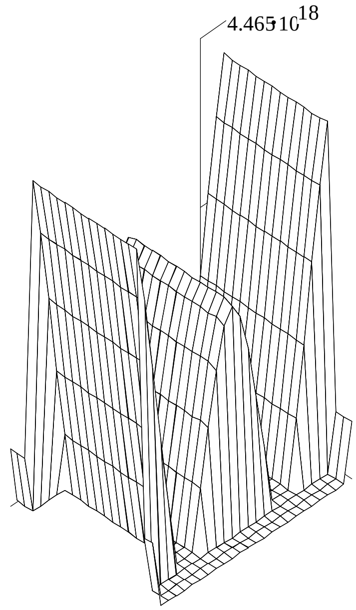

Figure Chapter 3 .7: The electron concentration profile of a wide DBRTD. Here the concentration on either end is in the contact region and in between concentration is in the heterostructure quantum well. This is a wide model with 565Å between nodes. The solution is similar to independent solutions at 565Å spacing across the device.

A narrow DBRTD device structure is shown in Figure Chapter 3 .8. This DBRTD will be described in greater detail in Chapter 6. It has a modulation doped quantum well, composed of a N- N++ N- regions, near a 50Å heterostructure quantum well. This simulation shows a narrow device 288 Å wide. There are lateral undulations in the N++ region due to interference effects.

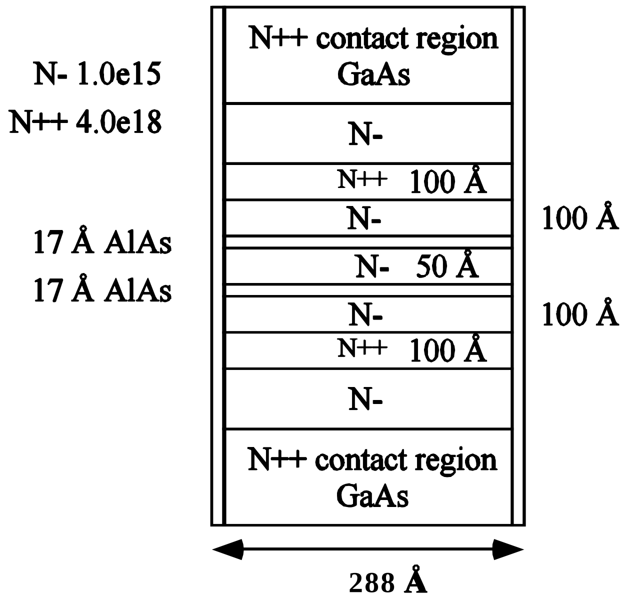

Figure Chapter 3 .8: This is the structure of the two dimensional DBRTD model.

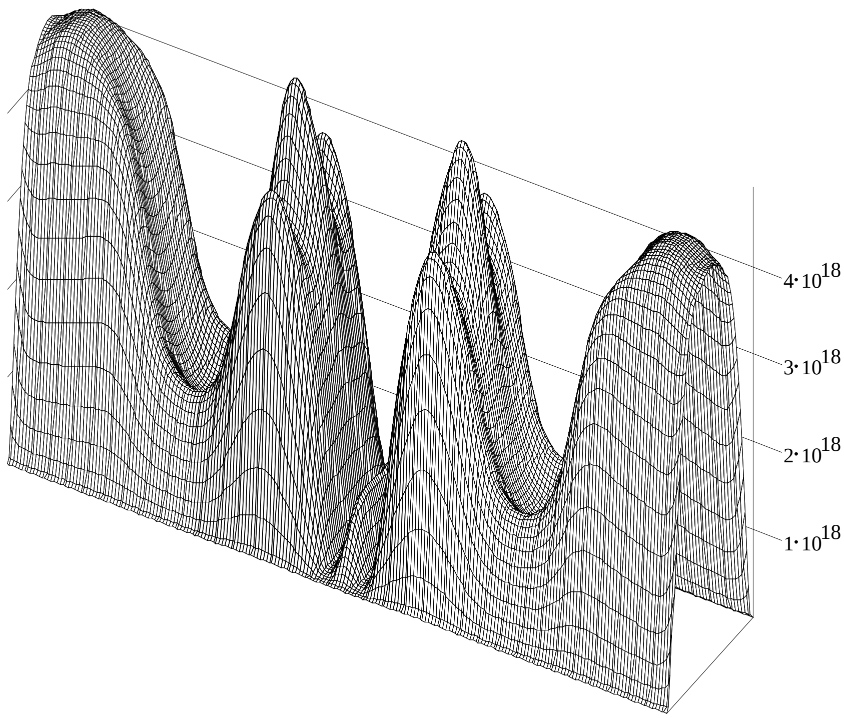

Figure Chapter 3 .9 This is the concentration profile in a very narrow DBTRD. A barrier is used on the sides to simulate Fermi level pinning. The high concentration on either end is in the contact region. The N++ regions show lateral interference effects.

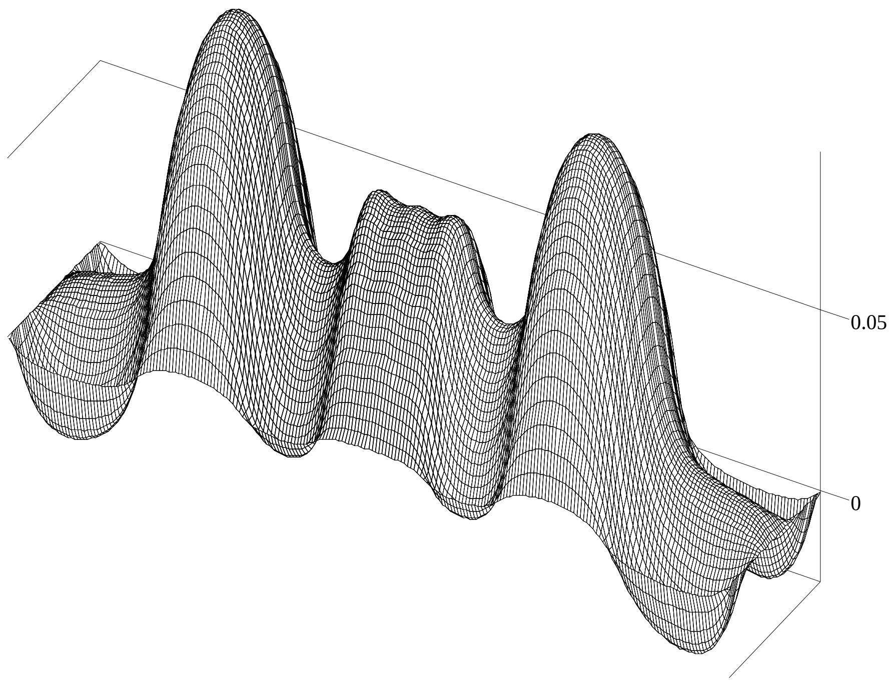

Figure Chapter 3 .10: This is the self-consistent potential profile.

Chapter 3 .10 Summary

One and two dimensional Schrödinger Poisson self consistent simulation provide an insight into tunneling, quantum interferrence, and low dimensional effects. Ultimately simulators should seamlessly include these effects in simulations of devices where these physical models are dominant. Recursion algorithms may be used with an adaptive solver to solve these problems or they can be solved, as presented here, using sparse LU decomposition. Run times are scale with the number of nodes to the 3/2 power. Sparse matrix implementation greatly adds to the efficiency. This has proven to be particularly important in employing effective integration procedures in problems with transmission spectra composed of resonant peaks. Location of these peaks is necessary for accurate inhomogeneous solution. The homogeneous solutions have been determined successfully here by approximate recursive methods.

Conceptually 2D Schrödinger Poisson simulation is valuable. Application to real world problems is difficult because of generally poor convergence characteristics. The narrow RTD simulation shows lateral interference effects that have not been seen. Simulations of subthreshold MOSFETs have been simulated by similar algorithms[42].