ECE 695
Numerical Simulations
Lecture 13: Drift-Diffusion Simulations

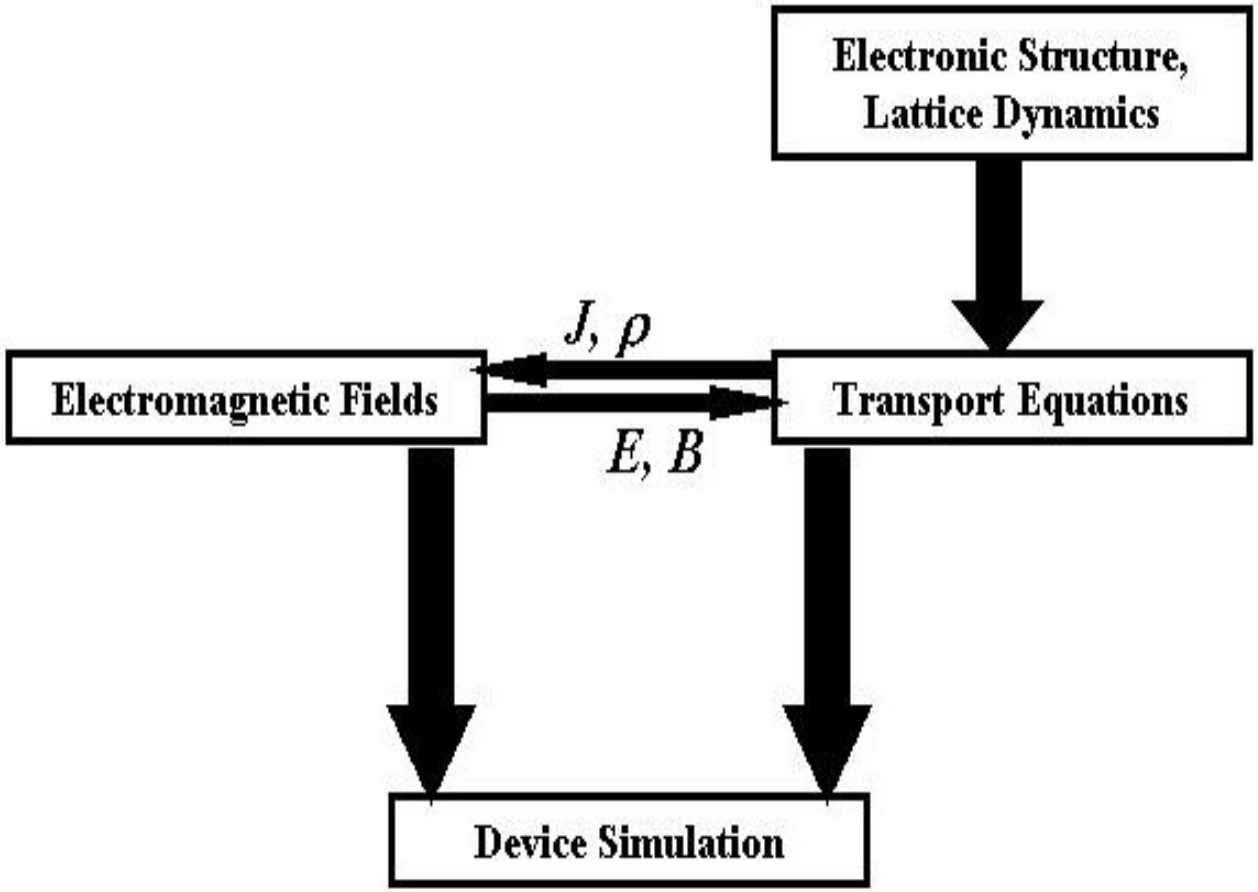
Prof. Peter Bermel

February 8, 2017

Outline

- Drift-Diffusion Model
  - Electronic transport theory
  - Solution scheme
  - Poisson solver
  - Scharfetter-Gummel scheme
  - Newton's method

Physics-Based Device Modeling



D. Vasileska and S.M. Goodnick, *Computational Electronics*, published by Morgan & Claypool , 2006.

Electronic Transport Theory

- Will assume electronic bandstructures known, and take a semiclassical approach
- Electrostatics modeled via Poissson's equation:

$$\nabla \cdot \varepsilon \nabla V = -\left(p - n + N_D^+ - N_A^- \right)$$

- Charge conservation is required:

$$\frac{\partial n}{\partial t} = \frac{1}{q} \nabla \cdot \mathbf{J}_n + U_n$$

$$\frac{\partial p}{\partial t} = -\frac{1}{q} \nabla \cdot \mathbf{J}_p + U_p$$

S. Selberherr: "Analysis and Simulation of Semiconductor Devices", Springer, 1984.

Electronic Transport Theory

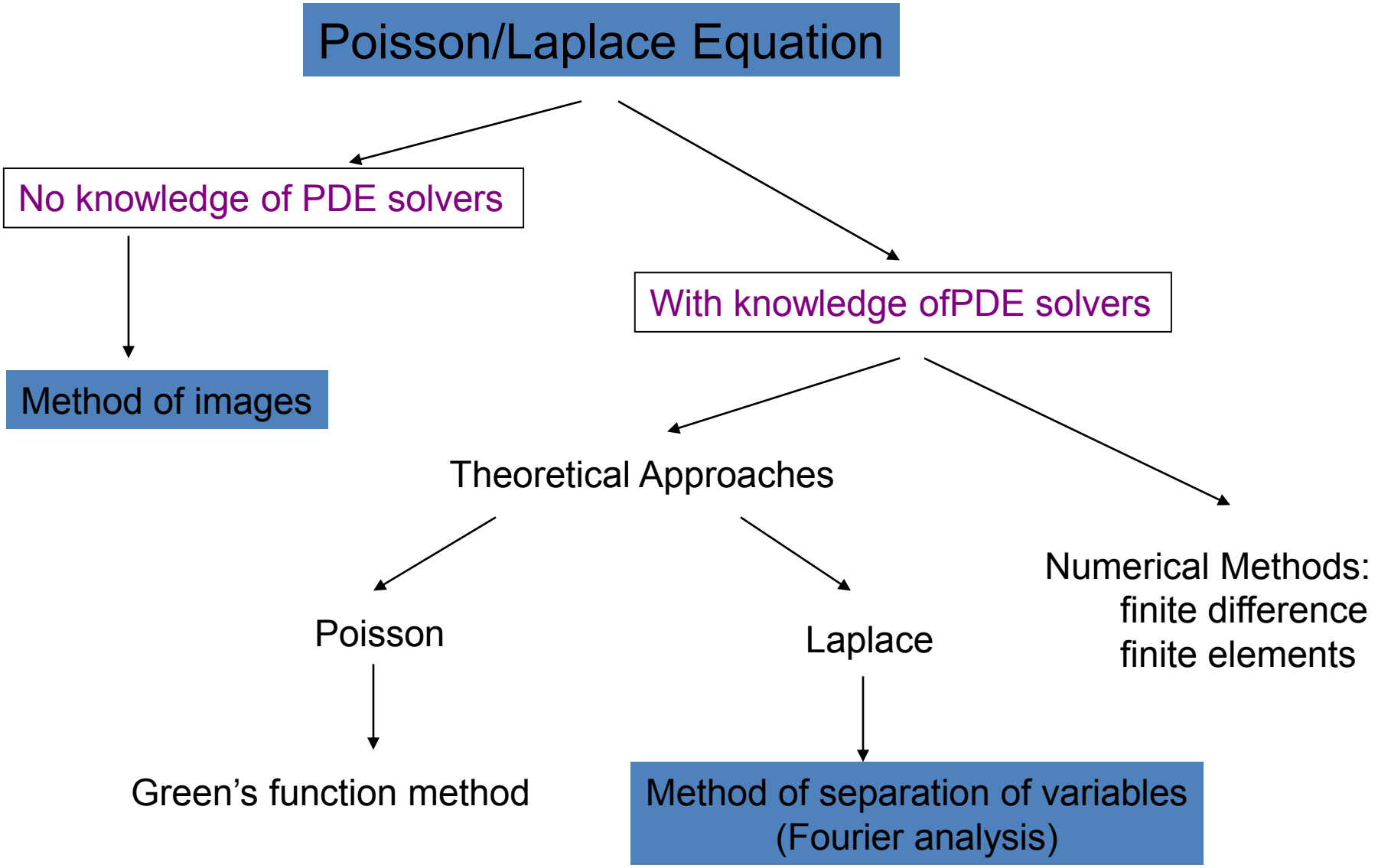
- Both p-type and n-type currents given by a sum of two terms:
  - Drift term, derived from Ohm's law
  - Diffusion term, derived from Second Law of Thermodynamics

$$J_n = qn(x)\mu_n E(x) + qD_n \frac{dn}{dx}$$

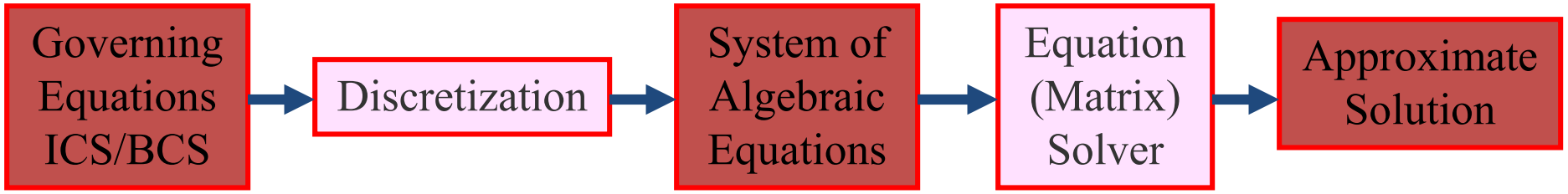
$$J_p = qp(x)\mu_p E(x) - qD_p \frac{dn}{dx}$$

S. Selberherr: "Analysis and Simulation of Semiconductor Devices", Springer, 1984.

Poisson/Laplace Equation Solution



Numerical Solution Details



Continuous Solutions

Finite-Difference

Finite-Volume

Finite-Element

Spectral

Boundary Element

Hybrid

Discrete Nodal Values

Tridiagonal

SOR

Gauss-Seidel

Krylov

Multigrid

$\varphi_i(x,y,z,t)$

$p(x,y,z,t)$

$n(x,y,z,t)$

D. Vasileska, *EEE533 Semiconductor Device and Process Simulation* Lecture Notes, Arizona State University, Tempe, AZ.

FEM Electronic Transport Model

- Much like in earlier work, will employ the following strategy:
  - Specify problem parameters, including bulk and boundary conditions
  - Construct finite-element mesh over spatial domain
  - Generate a linear algebra problem
  - Solve for key field variables: $\varphi_i(x,y,z,t)$

    $p(x,y,z,t)$

    $n(x,y,z,t)$

FEM Electronic Transport Model

- Regarding the grid set-up, there are several points that need to be made:
  - ✓ In critical device regions, where the charge density varies very rapidly, the mesh spacing has to be smaller than the extrinsic Debye length determined from the maximum doping concentration in that location of the device

$$L_D = \sqrt{\frac{\varepsilon k_B T}{N_{\max} e^2}}$$

  - ✓ Cartesian grid is preferred for particle-based simulations
  - ✓ It is always necessary to minimize the number of node points to achieve faster convergence
  - ✓ A regular grid (with small mesh aspect ratios) is needed for faster convergence

D. Vasileska, *EEE533 Semiconductor Device and Process Simulation* Lecture Notes, Arizona State

Example for Meshing

The function below is used to generate non-uniform mesh with constant mesh aspect ratio **r**. Input parameters are initial mesh size (**X0**), total number of mesh points (**N**) and the size of the domain over which we want these mesh points distributed (**XT**).

```
FUNCTION R_COEFF(XO, N, XT)
IMPLICIT REAL*4(A-H, O-Z)
LOGICAL FLAG$CONV
flag$conv = .false.
r = 3.
do while(.not.flag$conv)
    term1 = 1./(r-1.D0)
    term2 = r**float(N)-1.D0
    rnum  = xo*term1*term2 - xt
    term3 = float(N)*r**float(N-1)
    denom = xo*(term1*term3-term1*term1*term2)
    r_new = r - rnum/denom
    error = abs(r-r_new)/abs(r_new)
    if(error.le.1.e-10)then
       flag$conv = .true.
    else
       r = r_new
    endif
enddo
r_coeff = r
RETURN
END
```

This is determined by the maximum doping in the device in a particular region.

Poisson Equation

- The Poisson equation is of the following general form:

$$\nabla^2 \Phi(r) = f(r)$$

  - ✓ It accounts for Coulomb carrier-carrier interactions in the Hartree approximation
  - ✓ It is always coupled with some form of transport simulator except when equilibrium conditions apply
  - ✓ It has to be frequently solved during the simulation procedure to properly account for the fields driving the carriers in the transport part
  - ✓ There are numerous ways to numerically solve this equation that can be categorized into direct and iterative methods

Poisson Solver

- The 1D Poisson equation is of the form:

$$\frac{d^2\varphi}{dx^2} = -\frac{e}{\varepsilon}\left(p - n + N_D - N_A\right)$$

$$n = n_i \exp\left(\frac{E_F - E_i}{k_B T}\right) = n_i \exp(\varphi / V_T)$$

$$p = n_i \exp\left(\frac{E_i - E_F}{k_B T}\right) = n_i \exp(-\varphi / V_T)$$

D. Vasileska, *EEE533 Semiconductor Device and Process Simulation* Lecture Notes, Arizona State

Poisson Solver

- Perturbing potential by δ yields:

$$\frac{d^2\varphi}{dx^2} = -\frac{en_i}{\varepsilon}\left(e^{-\varphi/V_T} - e^{\varphi/V_T} + C/n_i\right) +$$

$$+\frac{en_i}{\varepsilon}\delta\left(e^{-\varphi/V_T} + e^{\varphi/V_T}\right)$$

$$\frac{d^2\varphi}{dx^2} - \frac{en_i}{\varepsilon}\left(e^{-\varphi/V_T} + e^{\varphi/V_T}\right)\varphi^{new} = -\frac{en_i}{\varepsilon}\left(e^{-\varphi/V_T} - e^{\varphi/V_T} + C/n_i\right) -$$

$$-\frac{en_i}{\varepsilon}\left(e^{-\varphi/V_T} + e^{\varphi/V_T}\right)\varphi^{old}$$

$$\delta = \varphi^{new} - \varphi^{old}$$

D. Vasileska, *EEE533 Semiconductor Device and Process Simulation* Lecture Notes, Arizona State

Poisson Solver

- Renormalized form

$$\frac{d^2\varphi}{dx^2} = -\left(p - n + C\right) + \delta\left(p + n\right)$$

$$\frac{d^2\varphi}{dx^2} - \left(p + n\right)\varphi^{new} = -\left(p - n + C\right) - \left(p + n\right)\varphi^{old}$$

$$\delta = \varphi^{new} - \varphi^{old}$$

D. Vasileska, *EEE533 Semiconductor Device and Process Simulation* Lecture Notes, Arizona State

Finite Difference Representation

$$\frac{1}{\Delta^2}\varphi_{i+1}^{n+1} - \left(\frac{2}{\Delta^2} + n_i + p_i\right)\varphi_i^{n+1} + \frac{1}{\Delta^2}\varphi_{i-1}^{n+1} =$$

$$= -(p_i - n_i + C_i) - (p_i + n_i)\varphi_i^n$$

Equilibrium:

$$n_i = \exp(\varphi_i^n), \quad p_i = \exp(-\varphi_i^n)$$

Non-Equilibrium:

n calculated using PM coupling and *p* still calculated as in equilibrium case (quasi-equilibrium approximation)

Criterion for Convergence

There are several criteria for the convergence of the iterative procedure when solving the Poisson equation, but the simplest one is that nowhere on the mesh is the absolute change in potential greater than 10^{-5} V.

This criterion has shown to be sufficient for all device simulations that have been performed within the Computational Electronics community.

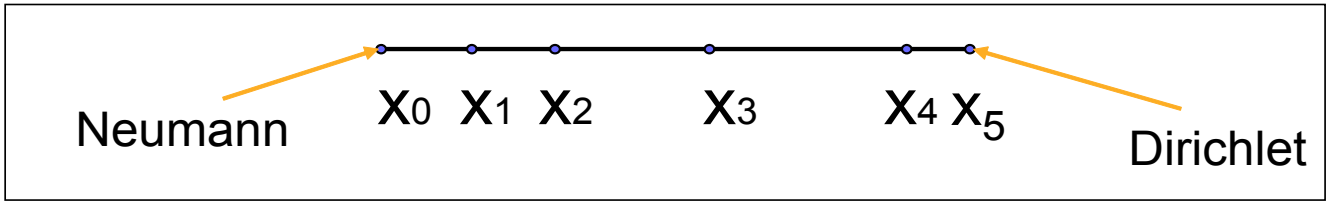
Boundary Conditions

- There are three types of boundary conditions that are specified during the discretization process of the Poisson equation:
  - Dirichlet (this is a boundary condition on the potential)
  - Neumann (this is a boundary condition on the derivative of the potential, i.e. the electric field)
  - Mixed boundary condition (combination of Dirichlet and Neumann boundary conditions)

- Note that when applying the boundary conditions for a particular structure of interest, at least one point MUST have Dirichlet boundary conditions specified on it to get the connection to the real world.

1D Discretization

- The resultant finite difference equations can be represented in a matrix form Au= f, where:



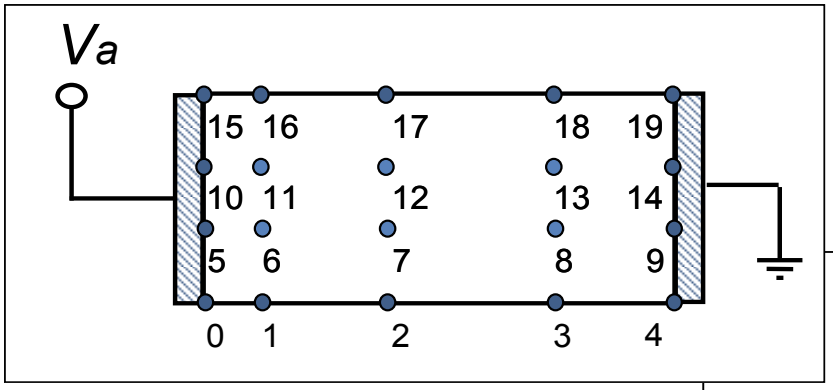
$$\mathbf{u} = (\Phi_0, \Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5)\ , \quad \mathbf{f} = (f_0, f_1, f_2, f_3, f_4, \Phi_5)\ ,$$

$$\mathbf{A} = \begin{pmatrix} b_0 & 2c_0 & & & & \\ a_1 & b_1 & c_1 & & 0 & \\ & a_2 & b_2 & c_2 & & \\ & & a_3 & b_3 & c_3 & \\ & 0 & & a_4 & b_4 & c_4 \\ & & & & 0 & 1 \end{pmatrix} \text{ where}$$

$$a_i = \frac{2}{dx_i^w (dx_i^w + dx_i^e)}\ ;\ b_i = \frac{2}{dx_i^e dx_i^w}\ ;\ c_i = \frac{2}{dx_i^e (dx_i^w + dx_i^e)}$$

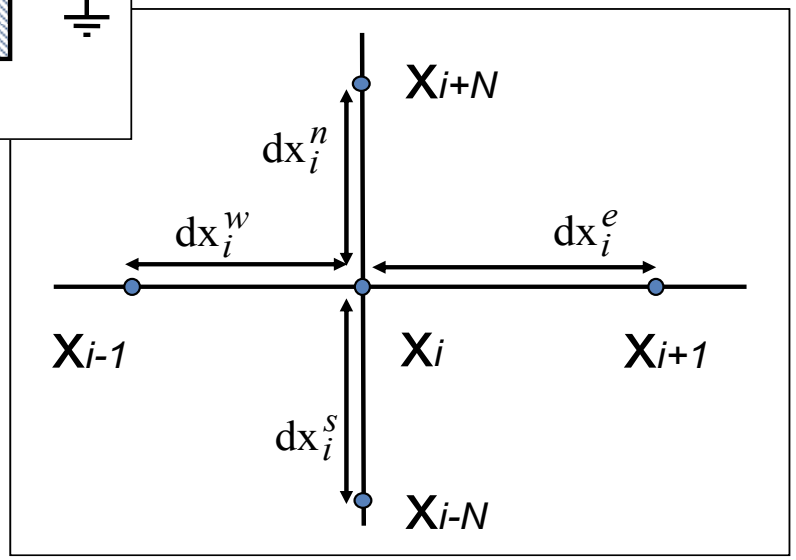
2D Discretization

- In 2D, the finite-difference discretization of the Poisson equation leads to a five point stencil:

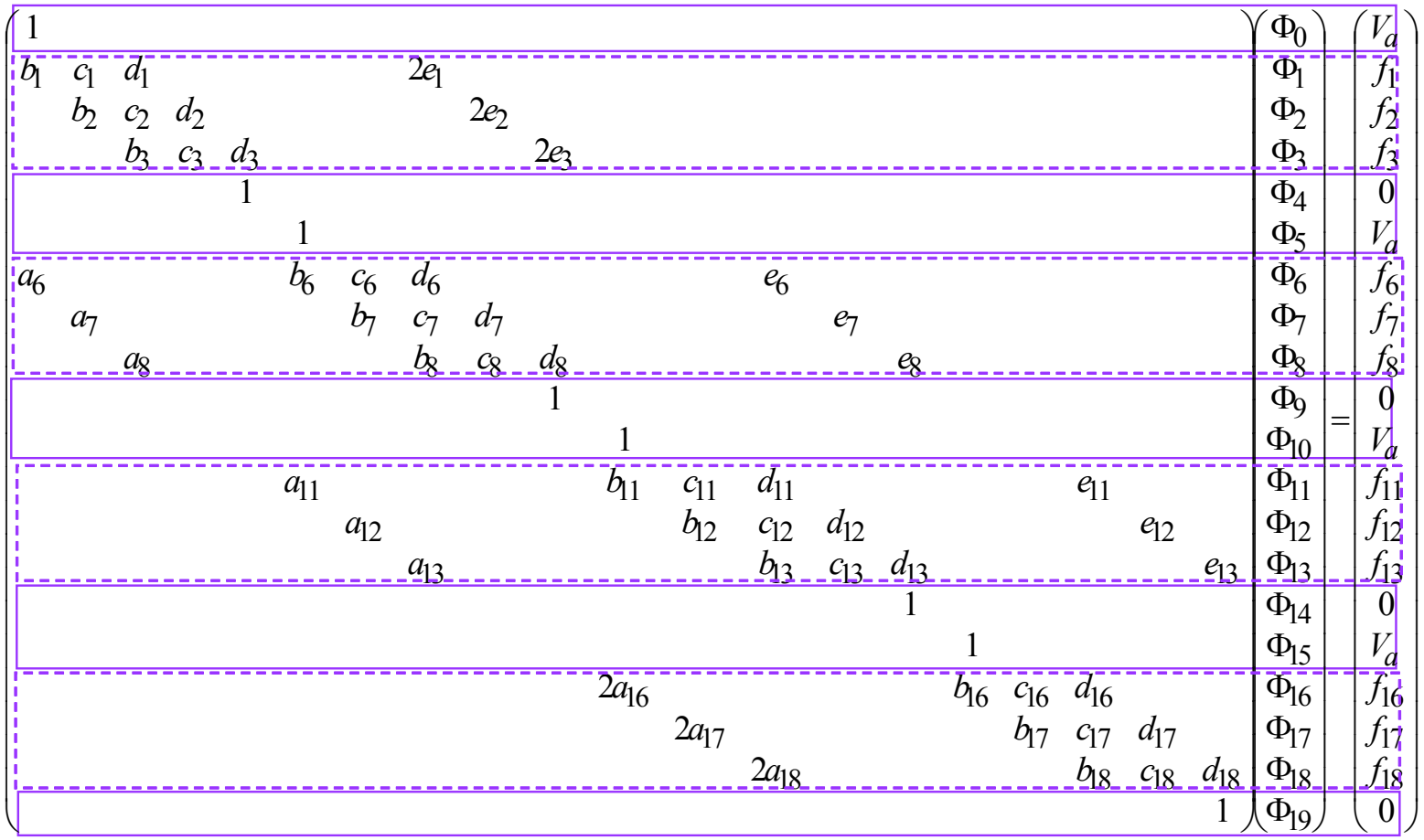


N=5,M=4

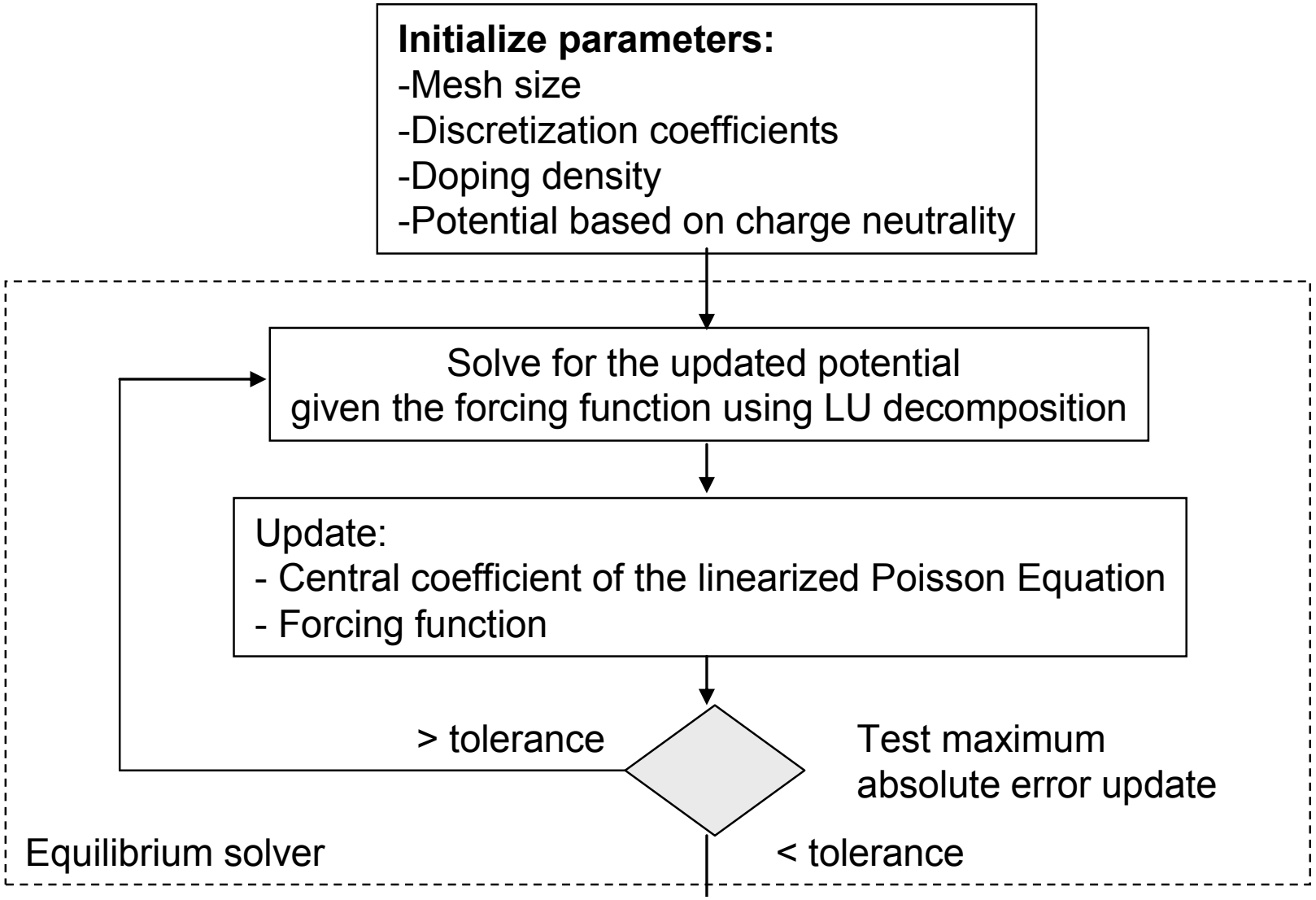
Dirichlet: 0,4,5,9,10,14,15,19
Neuman: 1,2,3,16,17,18



2D Discretization (cont'd)



Poisson Solver



D. Vasileska, *EEE533 Semiconductor Device and Process Simulation* Lecture Notes, Arizona State

Current Discretization

- The discretization of the continuity equation in conservative form requires the knowledge of the current densities

$$J_n(x) = en(x)\mu_n E + eD_n \nabla n$$
$$J_p(x) = ep(x)\mu_p E - eD_p \nabla p$$

  on the mid-points of the mesh lines connecting neighboring grid nodes. Since solutions are available only on the grid nodes, interpolation schemes are needed to determine the solutions.

- There are two schemes that one can use:
  - (a) Linearized scheme: *V, n, p*, μ and *D* vary linearly between neighboring mesh points
  - (b) Scharfetter-Gummel scheme: electron and hole densities follow exponential variation between mesh points

D. Vasileska, *EEE533 Semiconductor Device and Process Simulation* Lecture Notes, Arizona State

Naïve Linearization Scheme

- Within the linearized scheme, one has that

$$J_{i+1/2} = -en_{i+1/2}\mu_{i+1/2}\frac{V_{i+1}-V_i}{a_i} + eD_{i+1/2}\nabla n\big|_{i+1/2}$$

$$J_{i+1/2} = n_{i+1}\left[-\frac{e\mu_{i+1/2}}{2}\frac{V_{i+1}-V_i}{a_i} + \frac{eD_{i+1/2}}{a_i}\right] - n_i\left[\frac{e\mu_{i+1/2}}{2}\frac{V_{i+1}-V_i}{a_i} + \frac{eD_{i+1/2}}{a_i}\right]$$

- This scheme can lead to substantial errors in regions of high electric fields and highly doped devices.

D. Vasileska, *EEE533 Semiconductor Device and Process Simulation* Lecture Notes, Arizona State

Scharfetter-Gummel Scheme

- One solves the electron current density equation

$$J_{i+1/2} = -en\mu_{i+1/2}\frac{V_{i+1}-V_i}{a_i} + eD_{i+1/2}\frac{\partial n}{\partial x}$$
$$= -en\mu_{i+1/2}\frac{V_{i+1}-V_i}{a_i} + eD_{i+1/2}\frac{\partial n}{\partial V}\frac{\partial V}{\partial x}$$

for $n(V)$, subject to the boundary conditions

$$n(V_i) = n_i \quad \text{and} \quad n(V_{i+1}) = n_{i+1}$$

- The solution of this first-order differential equation leads to

$$n(V) = n_i[1-g(V)] + n_{i+1}g(V), \quad g(V) = \frac{e^{(V-V_i)/Vt}-1}{e^{(V_{i+1}-V_i)/Vt}-1}$$

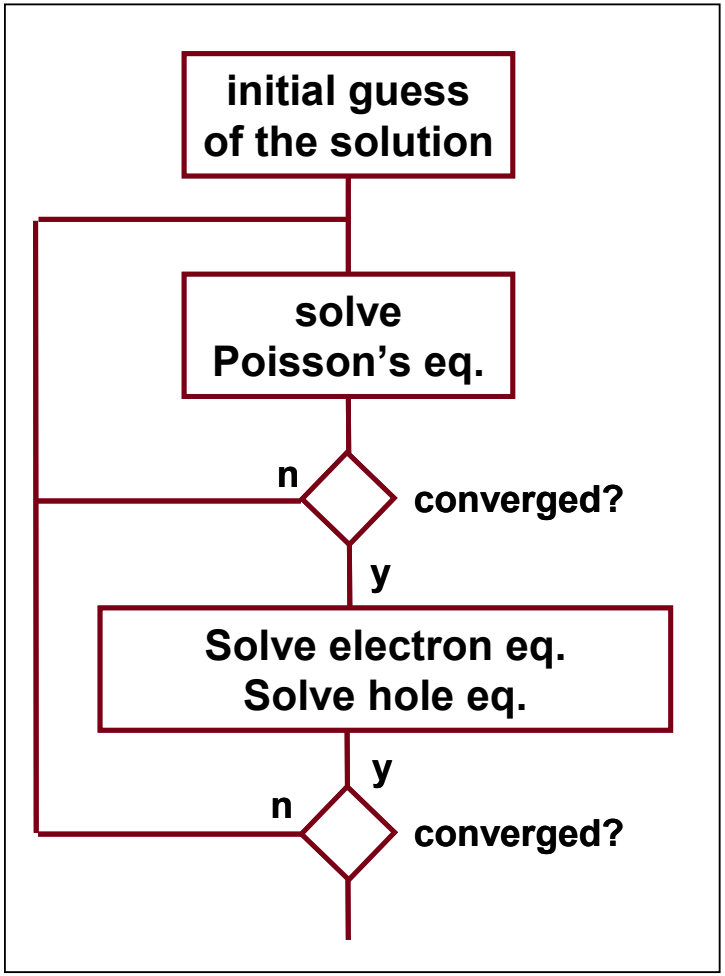
$$J_{i+1/2} = \frac{eD_{i+1/2}}{a_i}\left[n_{i+1}B\left(\frac{V_{i+1}-V_i}{Vt}\right) - n_iB\left(\frac{V_i-V_{i+1}}{Vt}\right)\right]$$

$B(x) = \dfrac{x}{e^x-1}$ is the Bernouli function

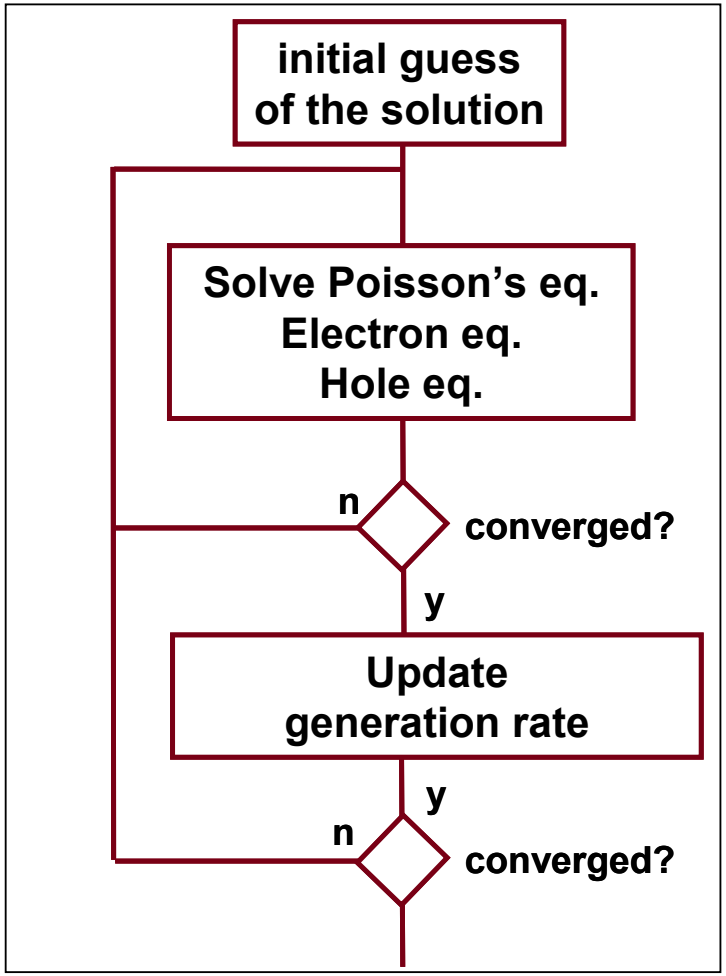
D. Vasileska, *EEE533 Semiconductor Device and Process Simulation* Lecture Notes, Arizona State

Gummel's Method (cont'd)

Original Gummel's scheme



Modified Gummel's scheme



Newton's Method

- The three equations that constitute the DD model, written in residual form are:

$$F_V(v,n,p)=0 \qquad F_n(v,n,p)=0 \qquad F_p(v,n,p)=0$$

- Starting from an initial guess, the corrections are calculated by solving:

$$\begin{bmatrix} \frac{\partial F_V}{\partial v} & \frac{\partial F_V}{\partial n} & \frac{\partial F_V}{\partial p} \\ \frac{\partial F_n}{\partial v} & \frac{\partial F_n}{\partial n} & \frac{\partial F_n}{\partial p} \\ \frac{\partial F_p}{\partial v} & \frac{\partial F_p}{\partial n} & \frac{\partial F_p}{\partial p} \end{bmatrix} \cdot \begin{bmatrix} \Delta v \\ \Delta n \\ \Delta p \end{bmatrix} = \begin{bmatrix} F_V \\ F_n \\ F_p \end{bmatrix} \Rightarrow \begin{aligned} V^{k+1} &= V^k + \Delta V^k \\ n^{k+1} &= n^k + \Delta n^k \\ p^{k+1} &= p^k + \Delta p^k \end{aligned}$$

Newton's Method (cont'd)

- The method can be simplified by the following iterative scheme:

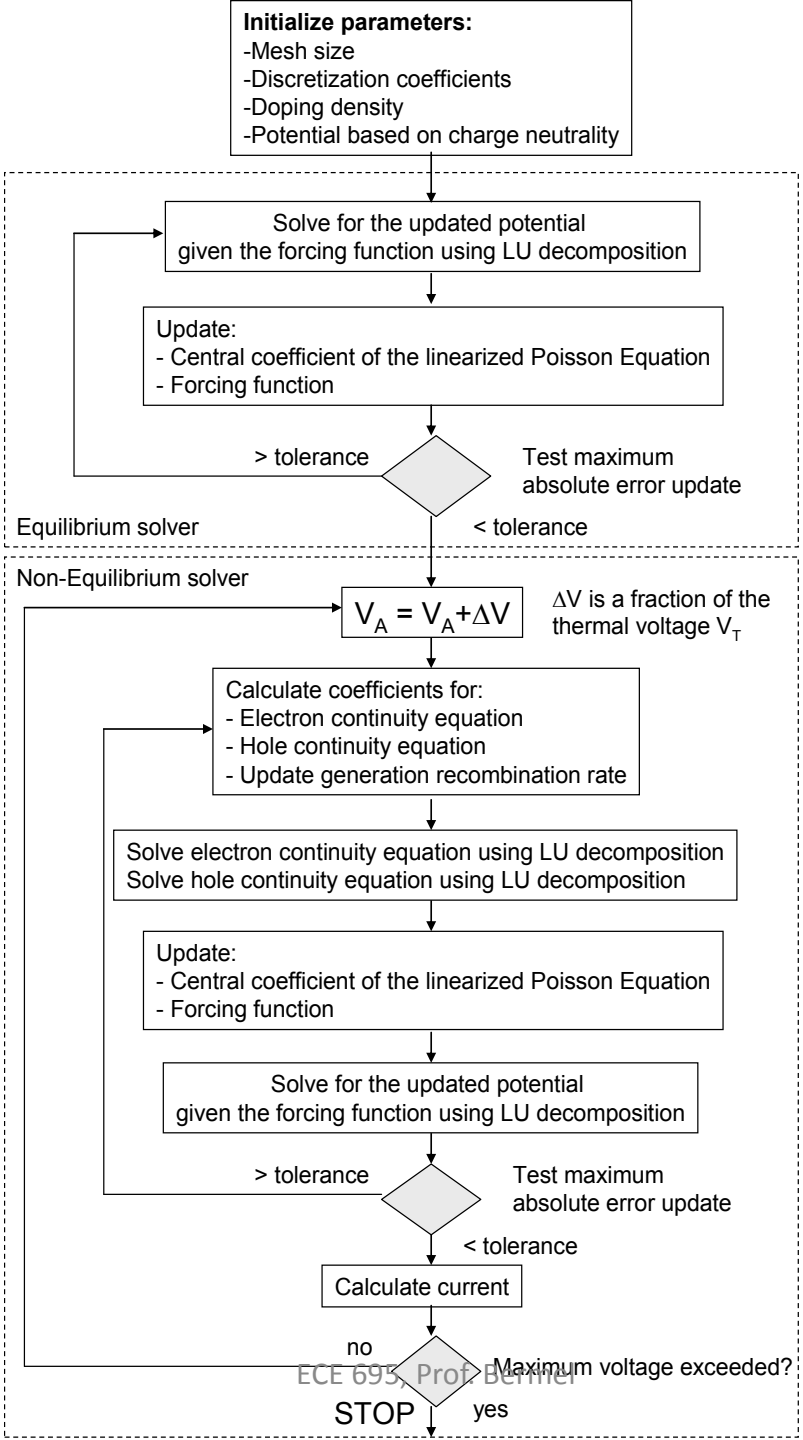
$$\begin{bmatrix} \frac{\partial F_V}{\partial V} & 0 & 0 \\ \frac{\partial F_n}{\partial V} & \frac{\partial F_n}{\partial n} & 0 \\ \frac{\partial F_p}{\partial V} & \frac{\partial F_p}{\partial n} & \frac{\partial F_p}{\partial p} \end{bmatrix} \cdot \begin{bmatrix} \Delta V \\ \Delta n \\ \Delta p \end{bmatrix} = -\begin{bmatrix} F_V \\ F_n \\ F_p \end{bmatrix} - \begin{bmatrix} 0 & \frac{\partial F_V}{\partial n} & \frac{\partial F_V}{\partial p} \\ 0 & 0 & \frac{\partial F_n}{\partial n} \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta V \\ \Delta n \\ \Delta p \end{bmatrix}$$

k+1 k

$$\frac{\partial F_V}{\partial V}\Delta V^{k+1} = -F_V - \frac{\partial F_V}{\partial n}\Delta n^k - \frac{\partial F_V}{\partial p}\Delta p^k$$

$$\frac{\partial F_V}{\partial V}\Delta n^{k+1} = -F_n - \frac{\partial F_n}{\partial V}\Delta V^{k+1} - \frac{\partial Fn}{\partial p}\Delta p^k$$

$$\frac{\partial F_p}{\partial p}\Delta p^{k+1} = -F_p - \frac{\partial F_p}{\partial V}\Delta V^{k+1} - \frac{\partial F_p}{\partial n}\Delta n^{k+1}$$


Initialize parameters:
-Mesh size
-Discretization coefficients
-Doping density
-Potential based on charge neutrality
Solve for the updated potential
given the forcing function using LU decomposition
Update:
- Central coefficient of the linearized Poisson Equation
- Forcing function
> tolerance
Test maximum
absolute error update
Equilibrium solver
< tolerance
Non-Equilibrium solver
$V_A = V_A + \Delta V$
ΔV is a fraction of the
thermal voltage V_T
Calculate coefficients for:
- Electron continuity equation
- Hole continuity equation
- Update generation recombination rate
Solve electron continuity equation using LU decomposition
Solve hole continuity equation using LU decomposition
Update:
- Central coefficient of the linearized Poisson Equation
- Forcing function
Solve for the updated potential
given the forcing function using LU decomposition
> tolerance
Test maximum
absolute error update
< tolerance
Calculate current
no
Maximum voltage exceeded?
STOP
yes

Next Class

- We will continue with electronic band structure modeling