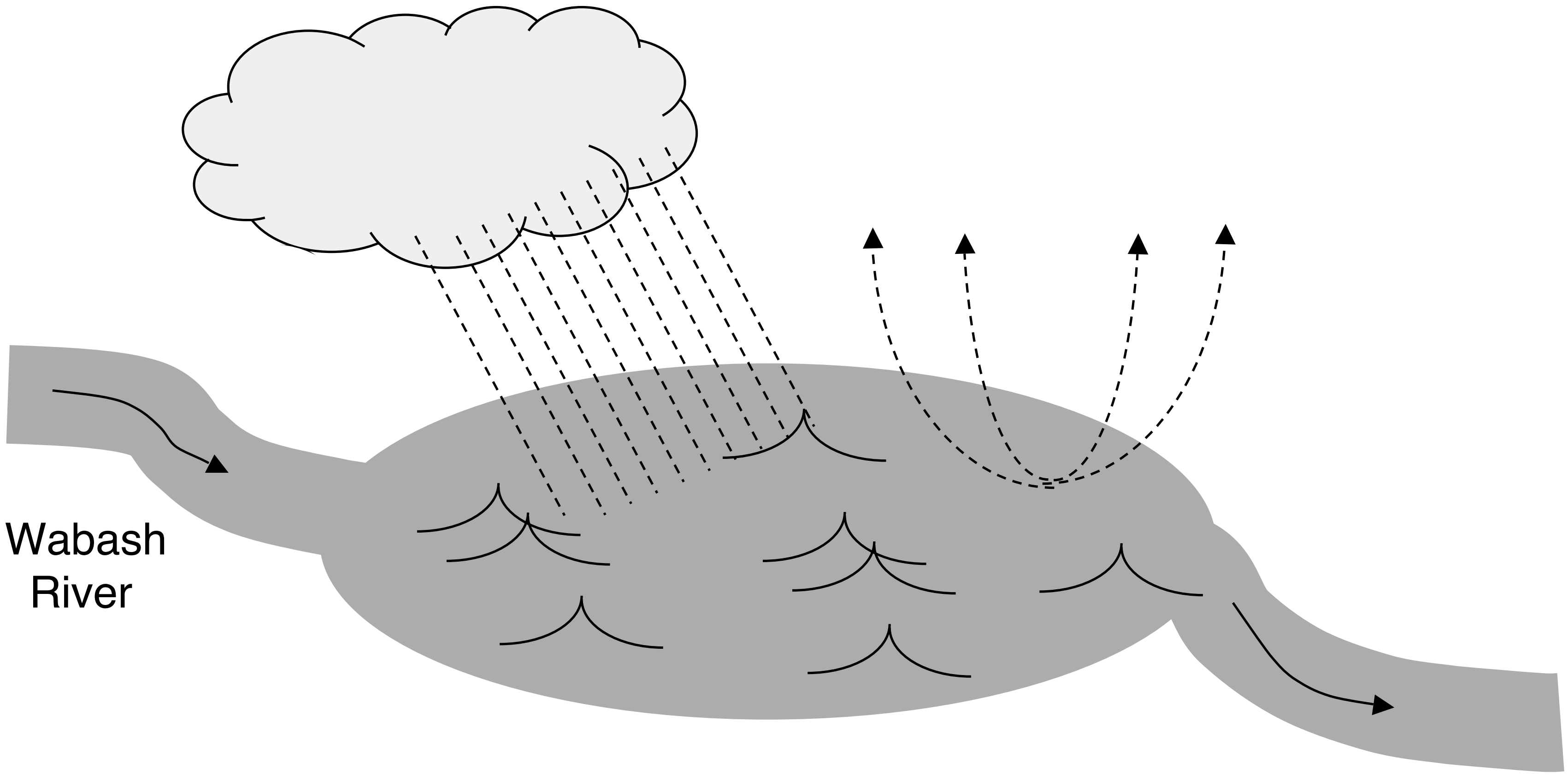
A Primer on Semiconductor Device Simulation

Mark Lundstrom
Purdue University
Network for Computational Nanotechnology

1) The Semiconductor Equations
2) Discretization
3) Numerical Solution
4) Physical Models
5) Examples

1) A Continuity Equation



Rate of increase of water level in lake = (in flow - outflow) + rain - evaporation

$$\frac{\partial p}{\partial t} = -\nabla \bullet \left(\vec{J}_p / q\right) + G - R$$

NCN

1) The Semiconductor Equations

Conservation Laws:

$$\nabla \bullet \vec{D} = \rho$$

$$\nabla \bullet \left(\vec{J}_n / -q \right) = (G - R)$$

$$\nabla \bullet \left(\vec{J}_p / q \right) = (G - R)$$

(steady-state)

Constitutive Relations:

$$\vec{D} = \kappa \varepsilon_0 \vec{E} = -\kappa \varepsilon_0 \vec{\nabla} V$$

$$\rho = q\left(p - n + N_D^+ - N_A^- \right)$$

$$\vec{J}_n = nq\mu_n \vec{E} + qD_n \vec{\nabla} n$$

$$\vec{J}_p = pq\mu_p \vec{E} - qD_p \vec{\nabla} p$$

$$R = f(n, p)$$

etc.

1) The Mathematical Problem

The "Semiconductor Equations"

$$\nabla \bullet \vec{D} = \rho$$

$$\nabla \bullet \left(\vec{J}_n / -q \right) = (G - R)$$

$$\nabla \bullet \left(\vec{J}_p / \ q \right) = (G - R)$$

3 coupled, nonlinear, second order PDE's for the 3 unknowns:

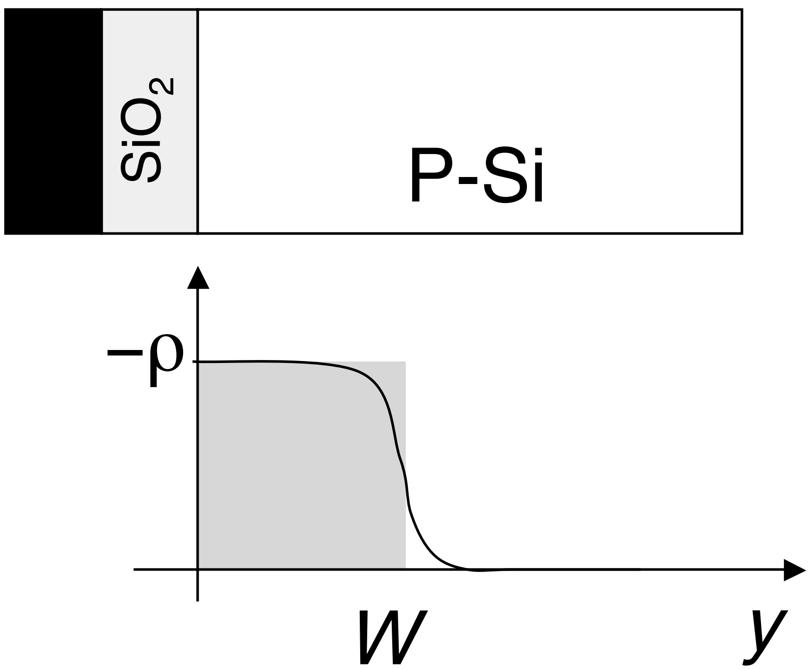
$$V(\vec{r}) \quad n(\vec{r}) \quad p(\vec{r})$$

Conservations laws: **exact**
Transport eqs. (drift-diffusion): **approximate**

1) The Depletion Approximation

(i) analytical solutions (e.g. depletion approximation)

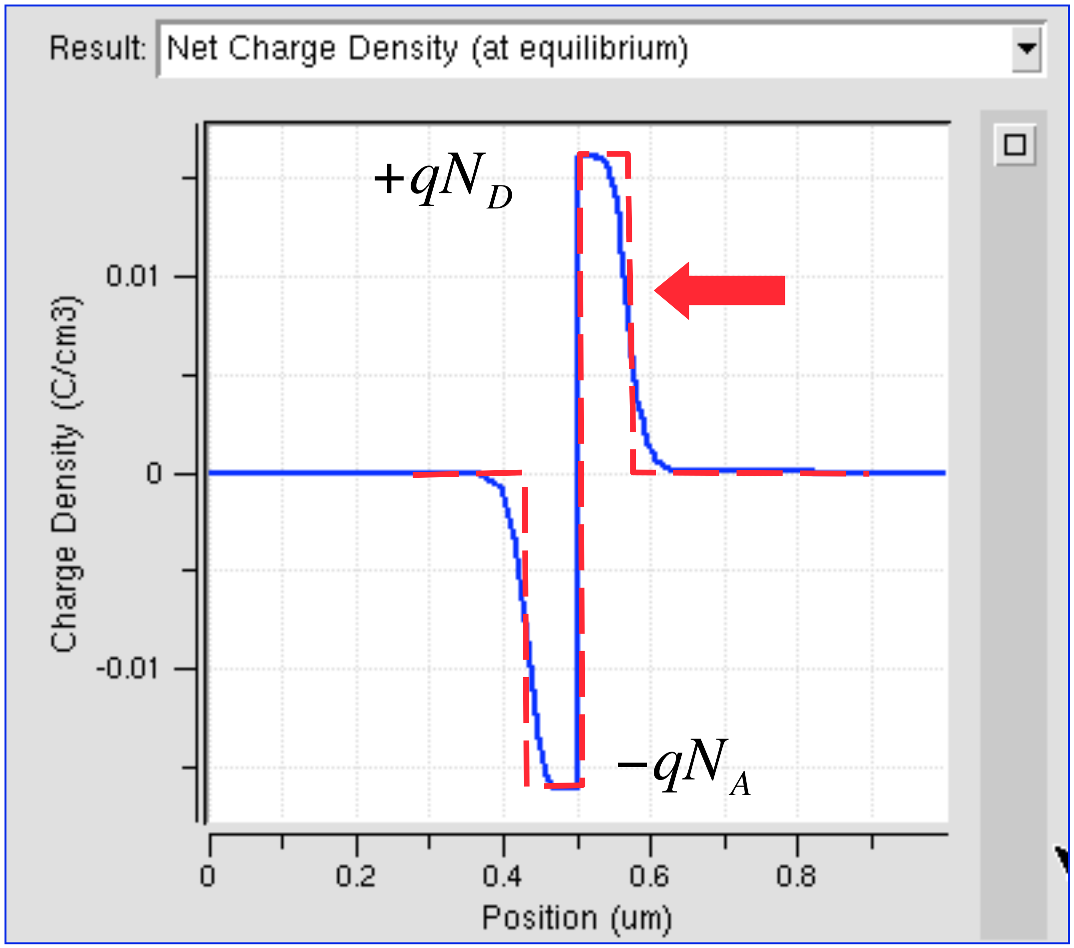
$0 < V_G < V_T$



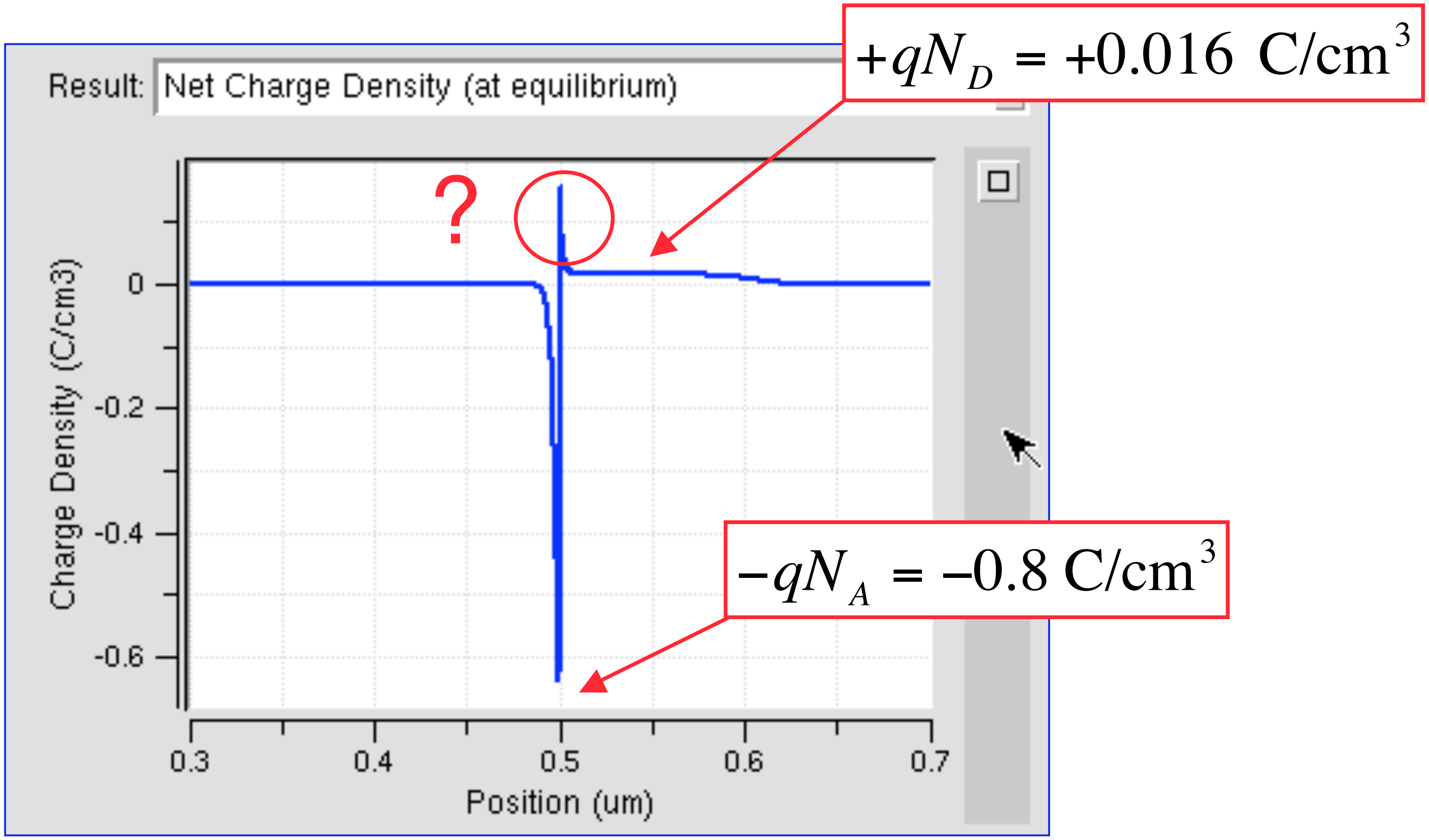
$$\frac{dD}{dx} = q\left(p - n - N_A\right)$$

$$\frac{d^2V}{dx^2} = \frac{qN_A}{\kappa_S \varepsilon_0} \qquad \left(x < W\right)$$

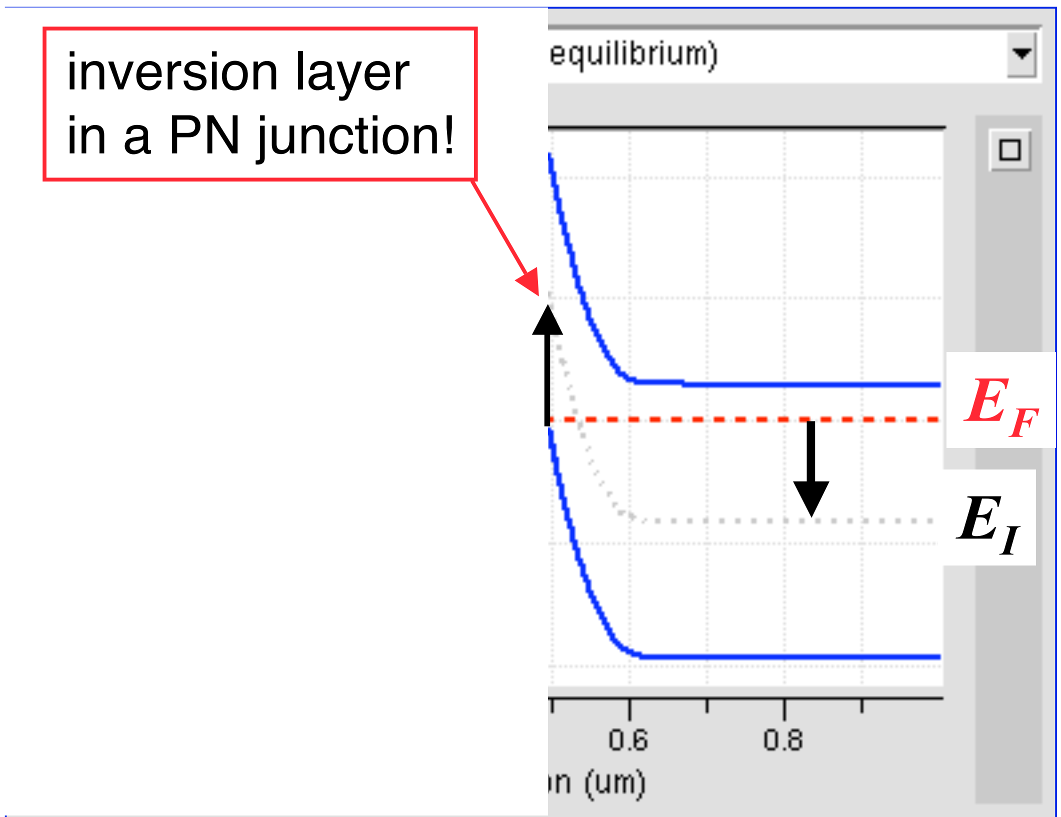
1) The DA vs. Numerical Solution



1) Asymmetric Junction

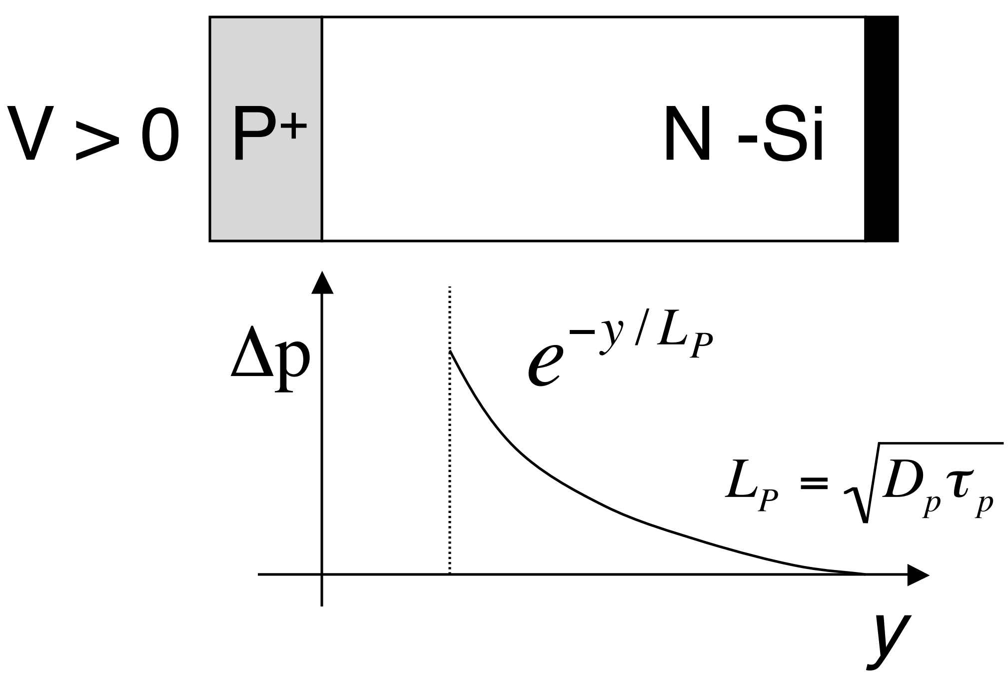


1) Asymmetric Junction



1) The Minority Carrier Diffusion Equation

(i) analytical solutions (e.g. minority carrier diffusion eq)



$$J_p = pq\mu_p E - qD_p \frac{dp}{dy}$$

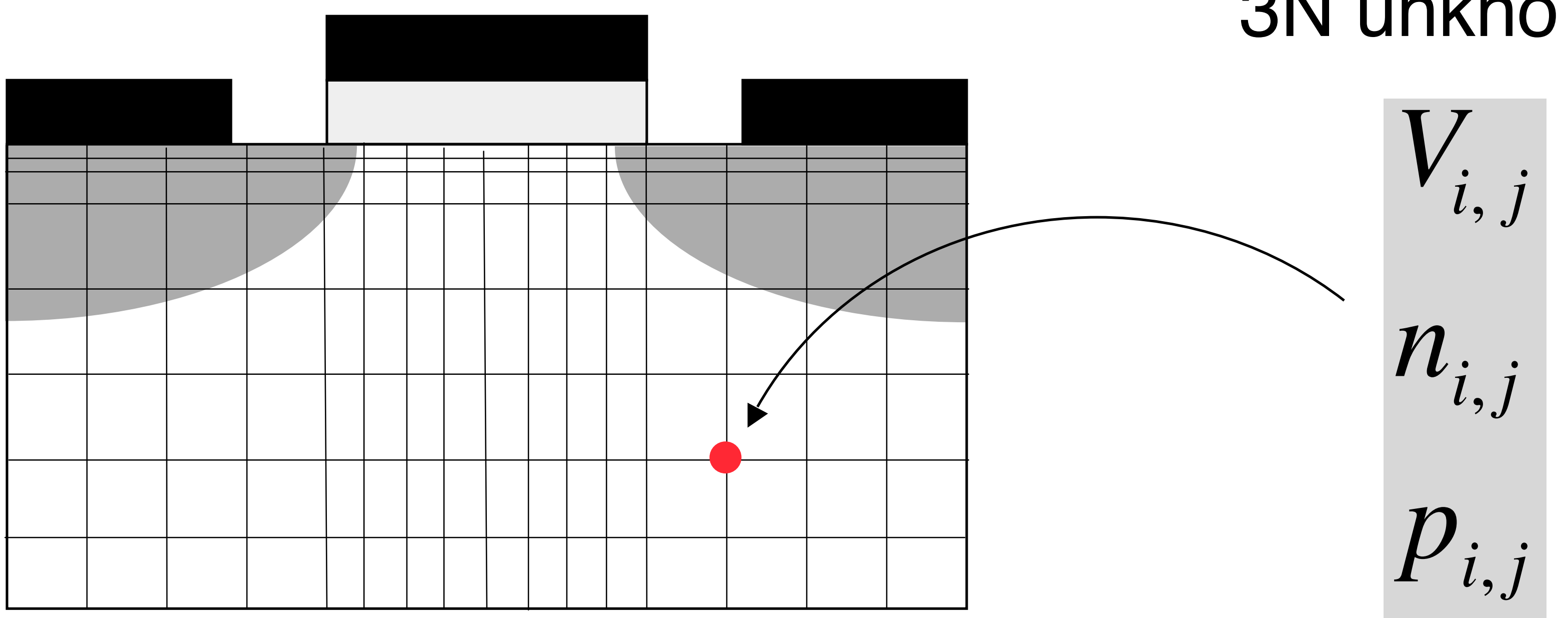
$$\frac{d\left(J_p/q\right)}{dy} = -R \approx -\frac{\Delta p}{\tau_p}$$

$$D_p \frac{d^2 \Delta p}{dy^2} - \frac{\Delta p}{\tau_p} = 0$$

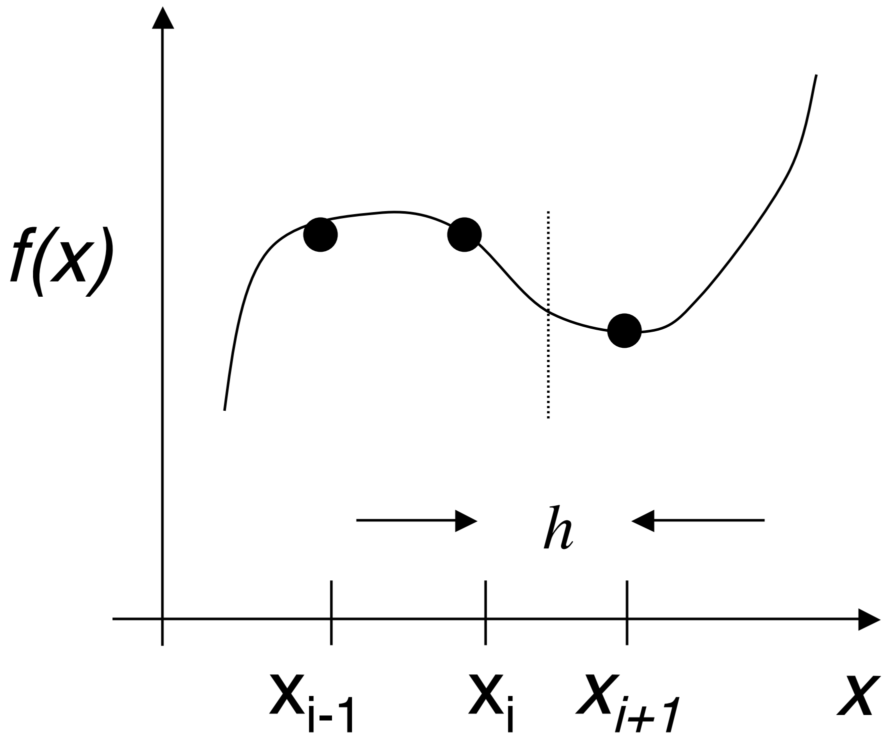
2) The Grid

(ii) "exact" numerical solutions

N nodes
3N unknowns



2) Discretization



$$\left.\frac{df}{dx}\right|_{\left(x_{i+1/2}\right)} = \frac{f_{i+1} - f_i}{h} + O(h^2)$$

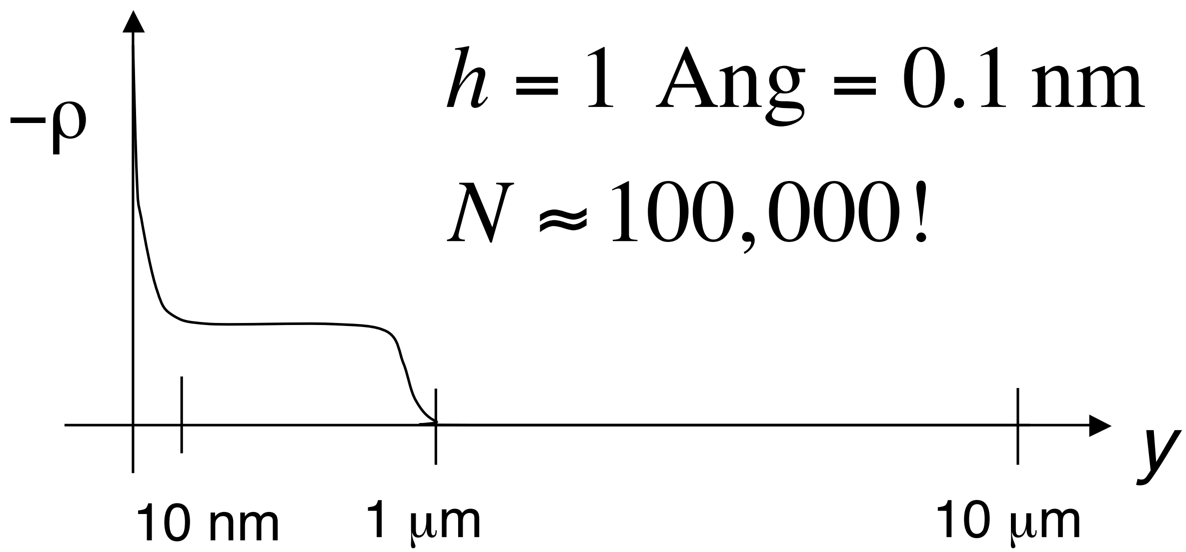
"centered difference"

Local truncation error (LTE)

2) Nonuniform Grid

Example: MOS problem

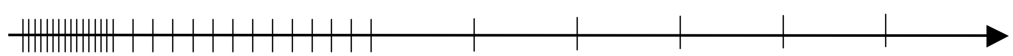
$V_G > V_T$



$$h = 1 \text{ Ang} = 0.1 \text{ nm}$$

$$N \approx 100{,}000!$$

Nonuniform mesh: $N \sim 100$ LTE is O(h)



2) Numerical Errors: finite word length

$$\left.\frac{df}{dx}\right|_{x_{i+1/2}} \approx \frac{f_{i+1} - f_i}{h}$$

$LTE \text{ --> } 0 \text{ as } h \text{ --> } 0$

f_{i+1} ---> f_i as h ---> 0

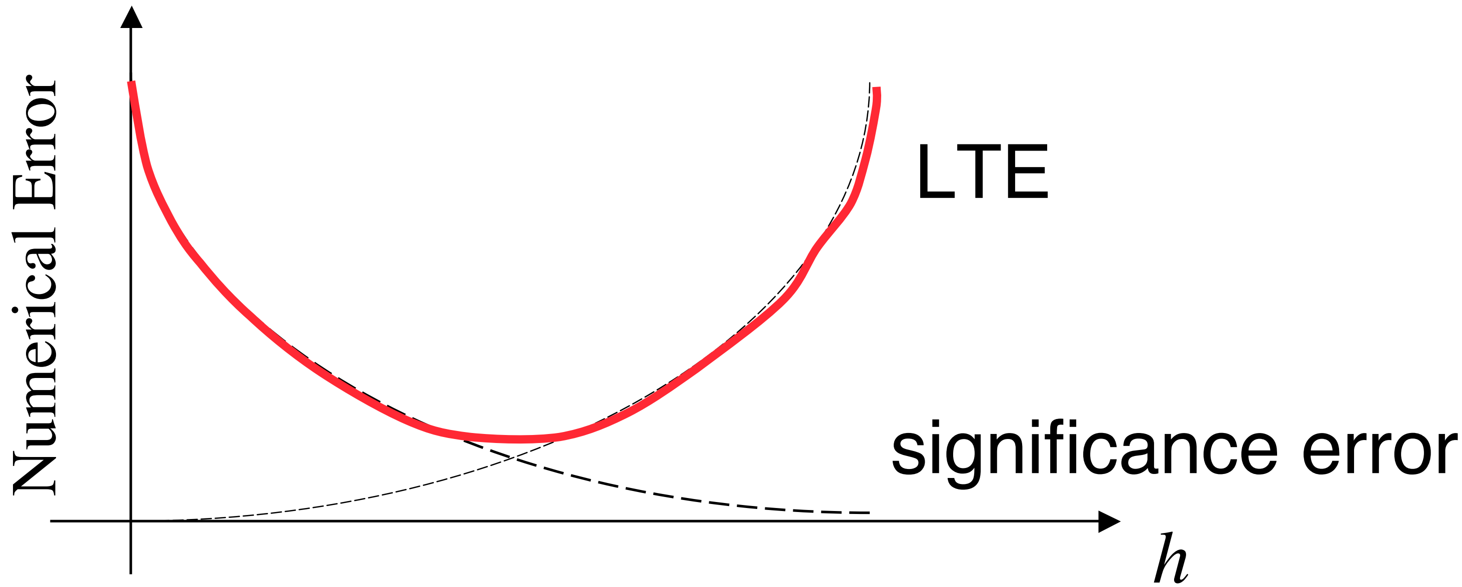
significance errors:

$f_{i+1} = 0.1234567890 \times 10^7$ 10 significant digits

$f_i = 0.1234567889 \times 10^7$ 10 significant digits

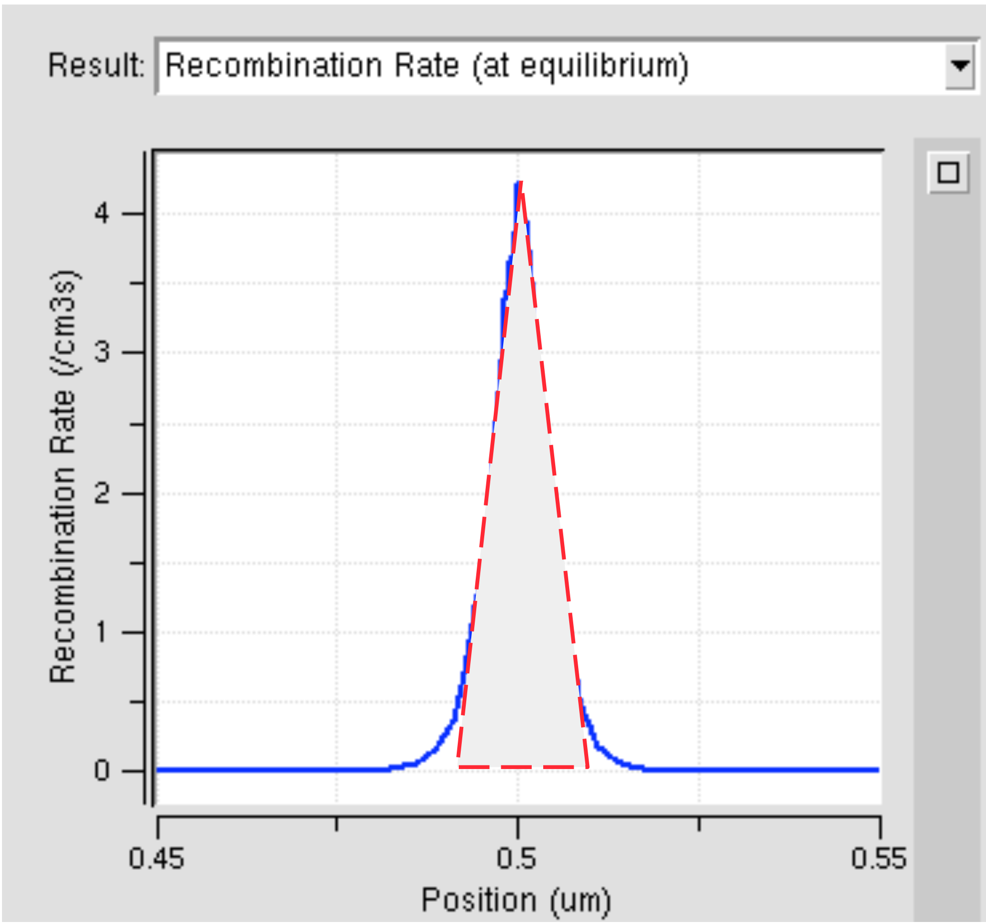
$f_{i+1} - f_i = 0.1 \times 10^{-2}$ 1 significant digit!

2) Numerical Error vs. Grid Spacing



For numerical solution of PDE's, LTE typically dominates, make *h* as small as possible
(but small *h* increases *N*, solution time, and memory!)

2) Numerical Error: Example



$$J = q\int_0^L R(x)dx \ \ \text{A/cm}^2$$

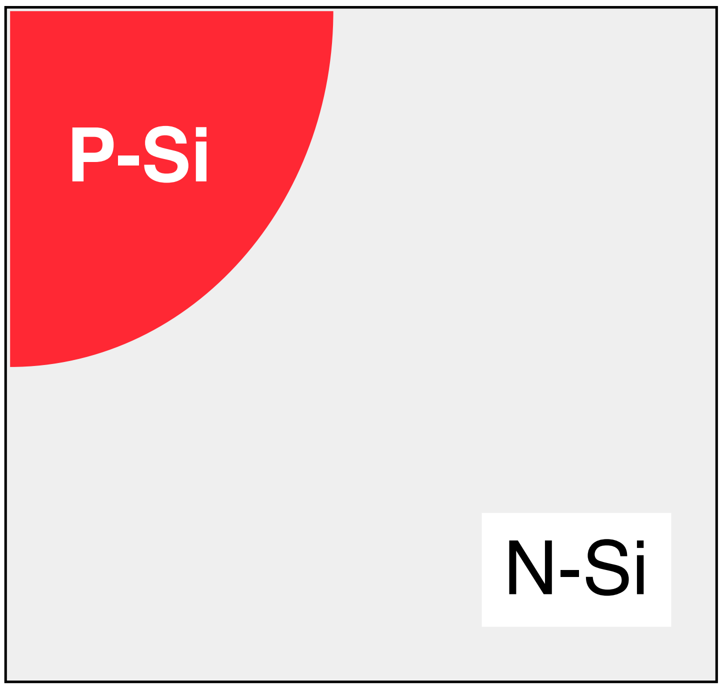
$$J \approx 1.3\times 10^{-24} \ \ \text{A/cm}^2$$

let A = 10μm x 10μm

$$I \approx 1.3\times 10^{-30} \ \ \text{A}$$

1 electron every 15M years

2) Discretization: Example



Gridding:

1) resolve variations in the unknowns

2) minimize LTE

3) minimize *N* (solution time)

(from Mark Pinto)

2) Discretization: Example

Gridding examples

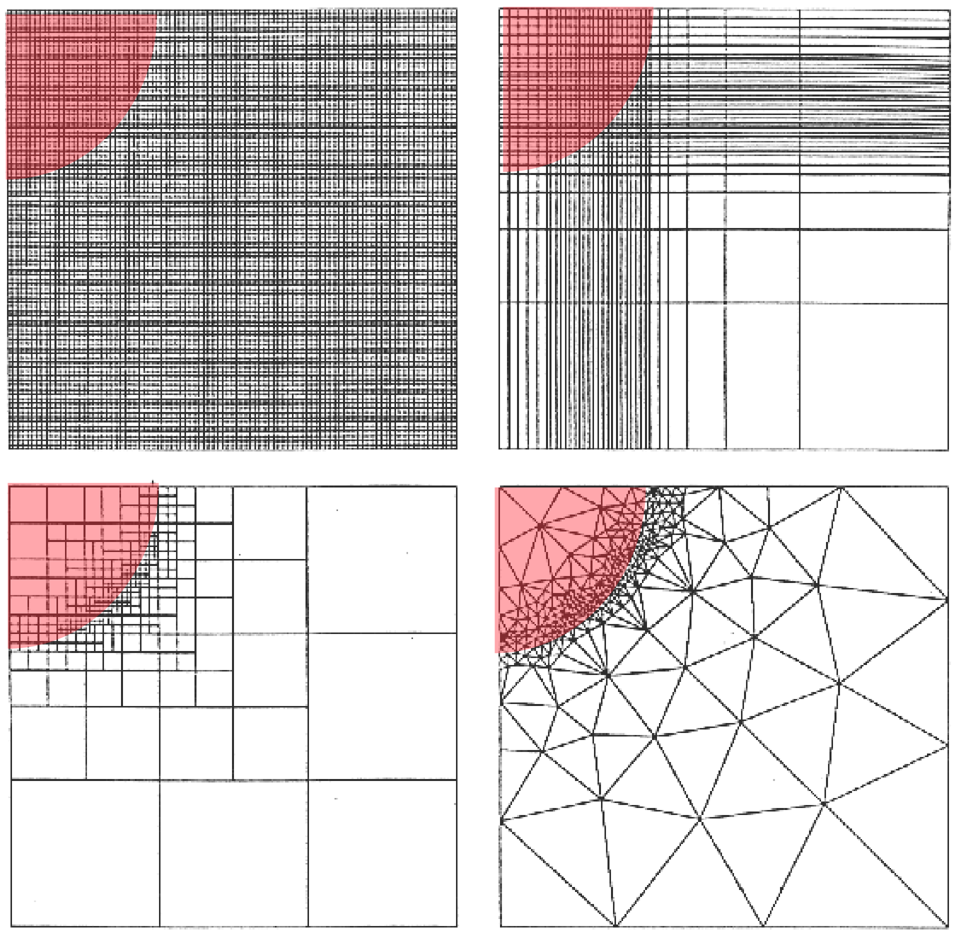
Uniform rectangular grid
9409 points

General tensor product
1156 points

Terminating line- rectangular
387 points

General triangular
264

2) Discretization: Example



Uniform
rectangular grid
9409 points

General
tensor product
1156 points

Terminating line-
rectangular
387 points

General
triangular
264

Gridding tips

- place nodes where V, p, and n are expected to vary
- avoid abrupt changes in *h*
- verify the accuracy of the grid by re-solving with a finer grid

<u>NOTE</u>: for simple MOS geometries, gridding can be automated

e.g. MINIMOS automatically defines a grid and redefines it when the bias changes

2) Discretizing a PDE

Poisson

$$\nabla \bullet \vec{D} = \rho$$

$$\int_{\Omega} \nabla \bullet \vec{D} \, d\Omega = \int_{\Omega} \rho \, d\Omega$$

$$\oint_{S} \vec{D} \bullet d\vec{S} = \int_{\Omega} \rho \, d\Omega$$

Current Continuity

$$\nabla \bullet \vec{J}_n = -q\left(G - R\right)$$

$$\int_{\Omega} \nabla \bullet \vec{J}_n \, d\Omega = \int_{\Omega} -q(G - R) \, d\Omega$$

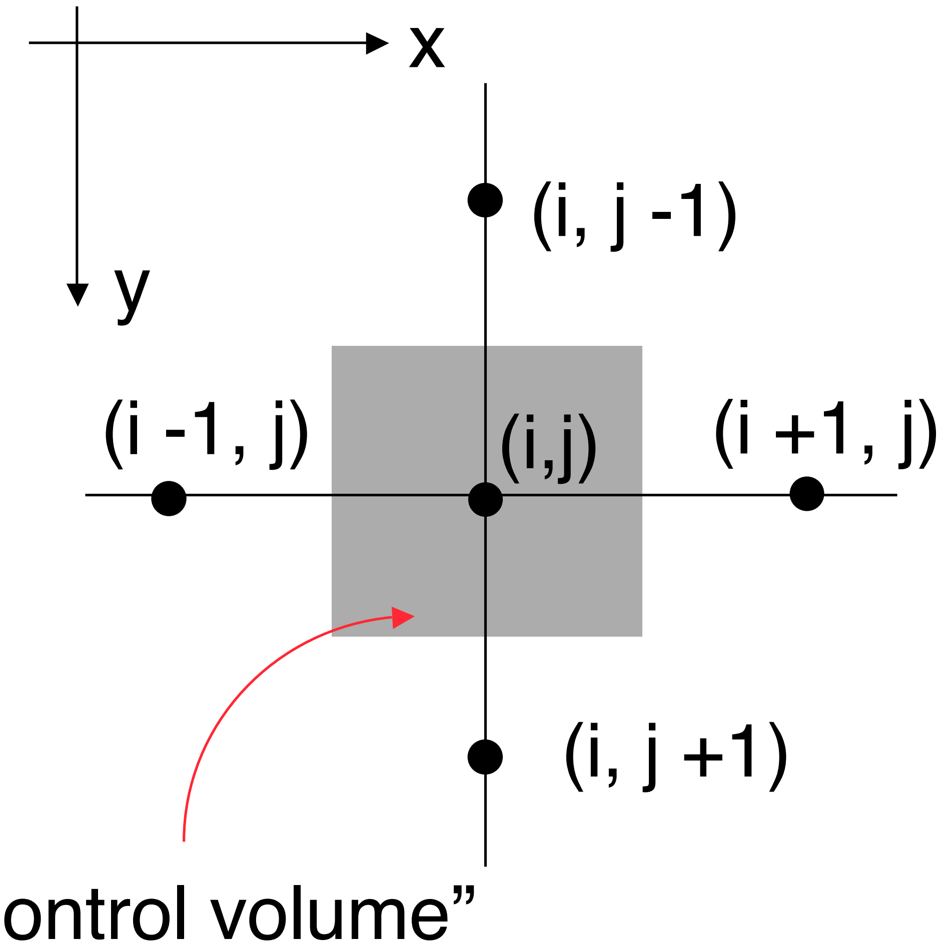
$$\oint_{S} \frac{\vec{J}_n}{-q} \bullet d\vec{S} = \int_{\Omega} (G - R) \, d\Omega$$

2) Control Volume

3 unknowns at each node:

$$V_{ij}, n_{ij}, p_{ij}$$

Need 3 equations
at each node

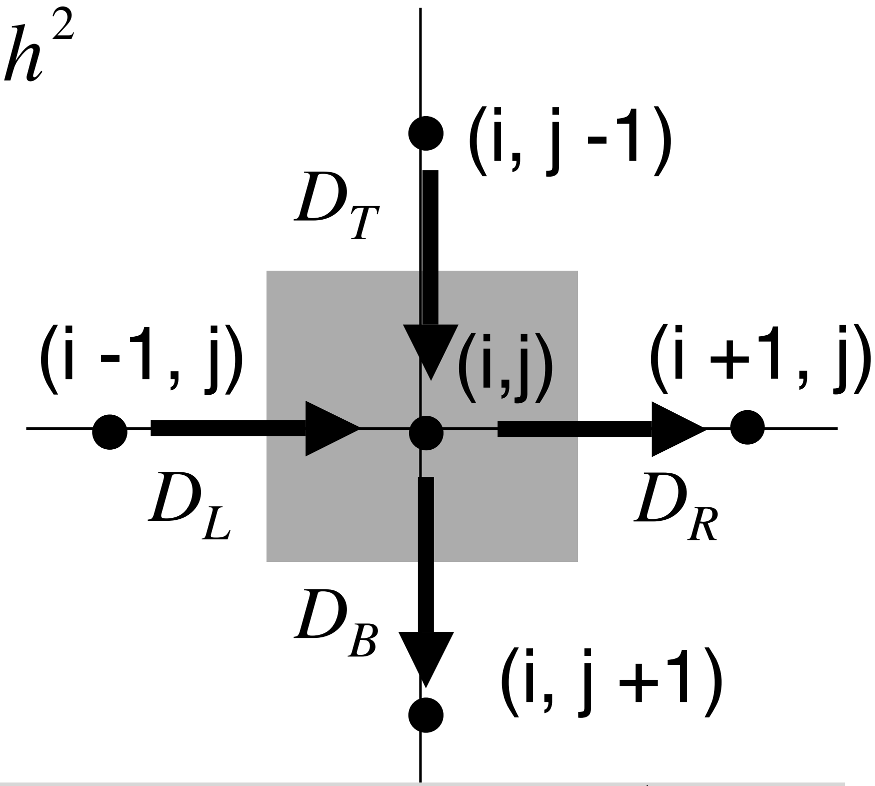


2) Discretizing Poisson's Equation

$$\left(D_R + D_B - D_L - D_T\right)h = \rho_{i,j}h^2$$

$$D_L = \kappa_S\,\varepsilon_0 E_L$$

$$D_L \approx \frac{\kappa_S\,\varepsilon_0}{h}\left(V_{i-1,j} - V_{i,j}\right)$$



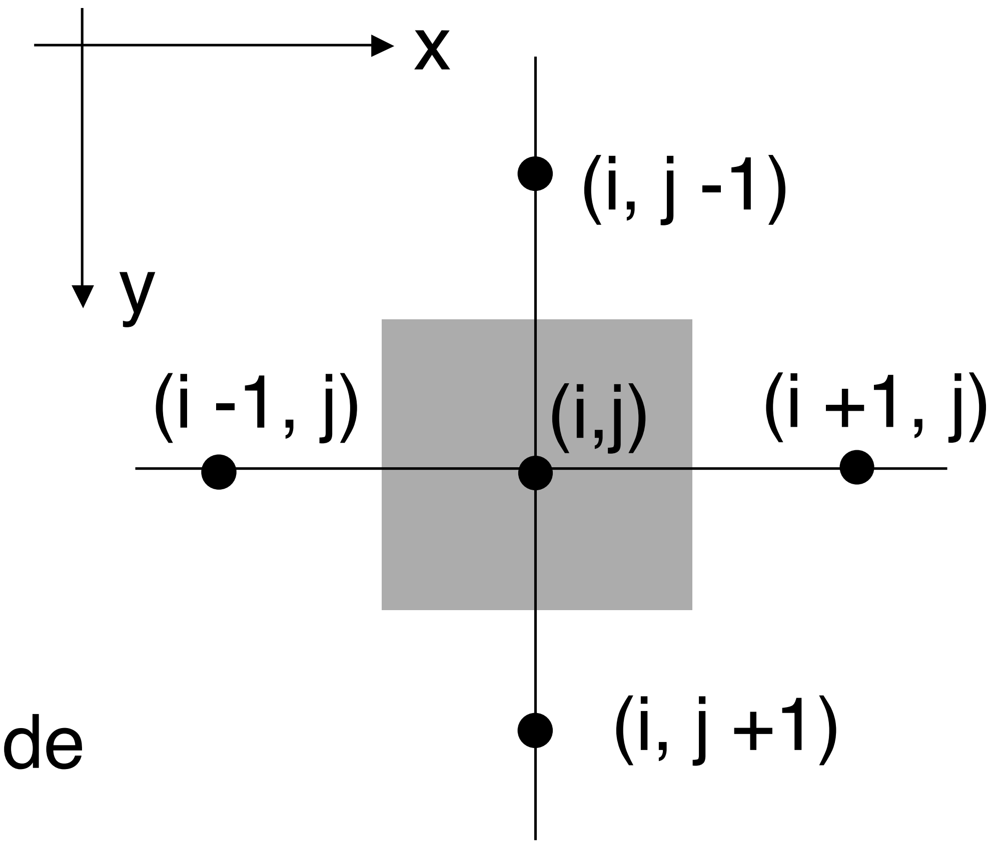
$$F_V^{i,j}\left(V_{i,j-1}, V_{i-1,j}, V_{i,j}, V_{i+1,j}, V_{i,j+1}, n_{i,j}, p_{i,j}\right) = 0$$

2) The 3 Discretized Equations

$$F_V^{i,j} = 0$$

$$F_n^{i,j} = 0$$

$$F_p^{i,j} = 0$$

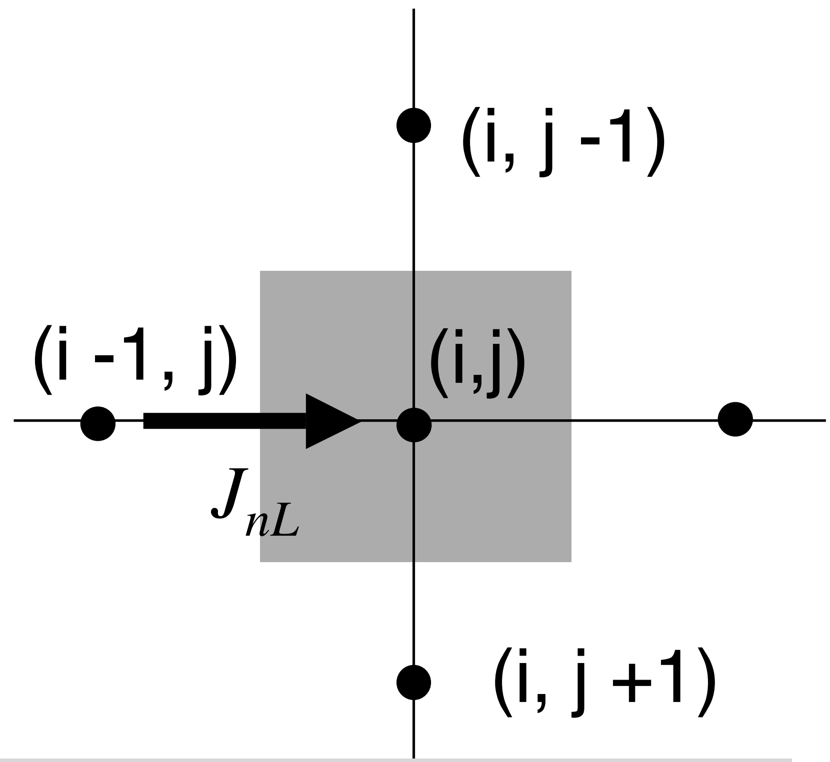


3 unknowns at each node
N nodes
3N unknowns and 3N equations (nonlinear!)

2) Discretization: pitfalls

$$\nabla \bullet \vec{J}_n = -q(G - R)$$

$$J_{nL} = -nq\mu_n \frac{dV}{dx} + kT\mu_n \frac{dn}{dx}$$



The simplest approach…..

$$\frac{J_{nL}}{kT\mu_n} = -\left(\frac{n_{i-1,j} + n_{i,j}}{2}\right)\left(\frac{V_{i,j} - V_{i-1,j}}{h(kT/q)}\right) + \left(\frac{n_{i,j} - n_{i-1,j}}{h}\right)$$

2) Discretization: pitfalls

$$J_{nL} = 0$$

(equilibrium)

$$\frac{n_{i,j}}{n_{i-1,j}} = \frac{2(kT/q) - \Delta V}{2(kT/q) + \Delta V}$$

fails when:

$$\Delta V = V_{i,j} - V_{i-1,j} > 2(kT/q)$$

(use Scharfetter-Gummel discretization instead!)

3) Numerical Solution

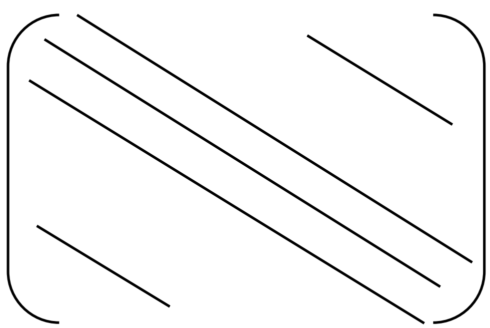
- have a system of 3N nonlinear equations to solve

- recall Poisson's equation at node (i,j):

$$F_V^{i,j}\left(V_{i,j-1,}V_{i-1,j},V_{i,j},V_{i+1,j},V_{i,j+1},n_{i,j},p_{i,j}\right)=0$$

<u>linear</u> if n_{ij} and p_{ij} are known $\quad \left[A\right]\vec{V}=\vec{b}$

[A]:



$$\vec{V}=\begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_N \end{bmatrix}$$

3) Curse of Dimensionality

Linear systems:

1D	N ~ 100 nodes	[A]: 100 x 100
2D	N ~ 10,000	[A]: 10,000 x 10,000
3D	N ~ 100,000	[A]: huge!

Sparseness = # of non-zero elements / total number
(~ 5 / *N* for 2D)

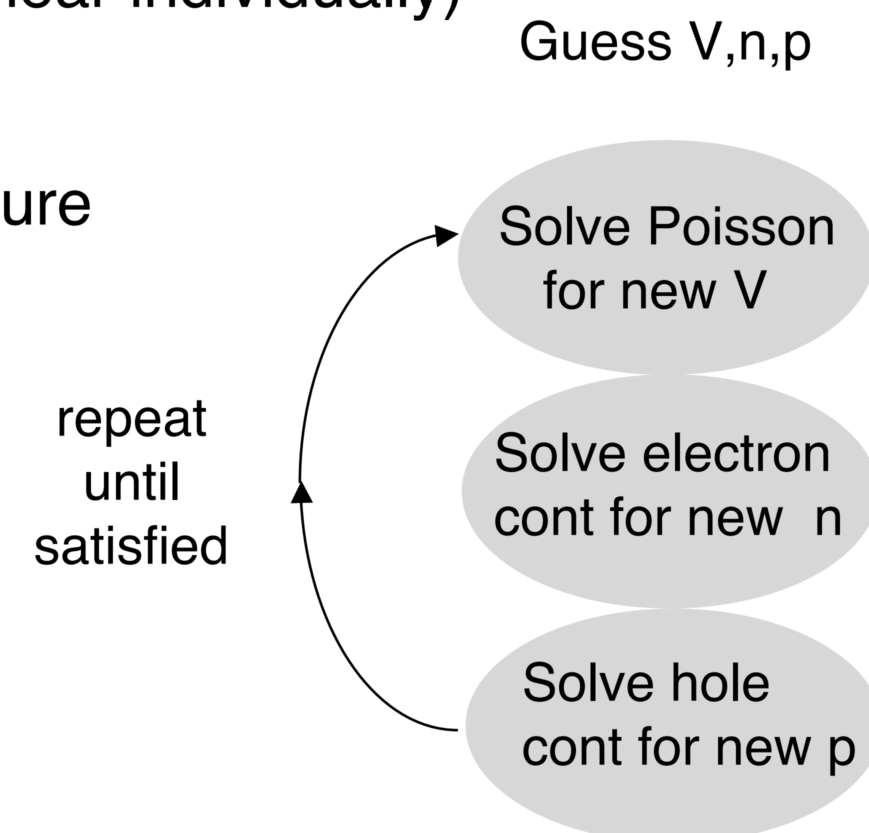
Linear system solution methods:

direct
iterative

3) Uncoupled Numerical Solution

The semiconductor equations are nonlinear!
(but they are linear individually)

Uncoupled solution procedure



3) Coupled vs. Uncoupled Numerical Solution

1) Uncoupled (sequential) method:

basis of Gummel's method
memory efficient
may converge rapidly at low bias; slowly at high bias

2) Coupled method:

a generalization of Newton's method
requires more memory
converges more quickly
may require a careful initial guess (e.g. from a sequential method)

3) Numerical Solution: Stopping

How do we know when we're done?

1)

$$\begin{pmatrix} \vec{F}_V \\ \vec{F}_n \\ \vec{F}_p \end{pmatrix} = \vec{0} \qquad \begin{pmatrix} \vec{F}_V(\vec{V}^k, \vec{n}^k, \vec{p}^k) \\ \vec{F}_n(\vec{V}^k, \vec{n}^k, \vec{p}^k) \\ \vec{F}_p(\vec{V}^k, \vec{n}^k, \vec{p}^k) \end{pmatrix} = \vec{r}$$

|| r ||

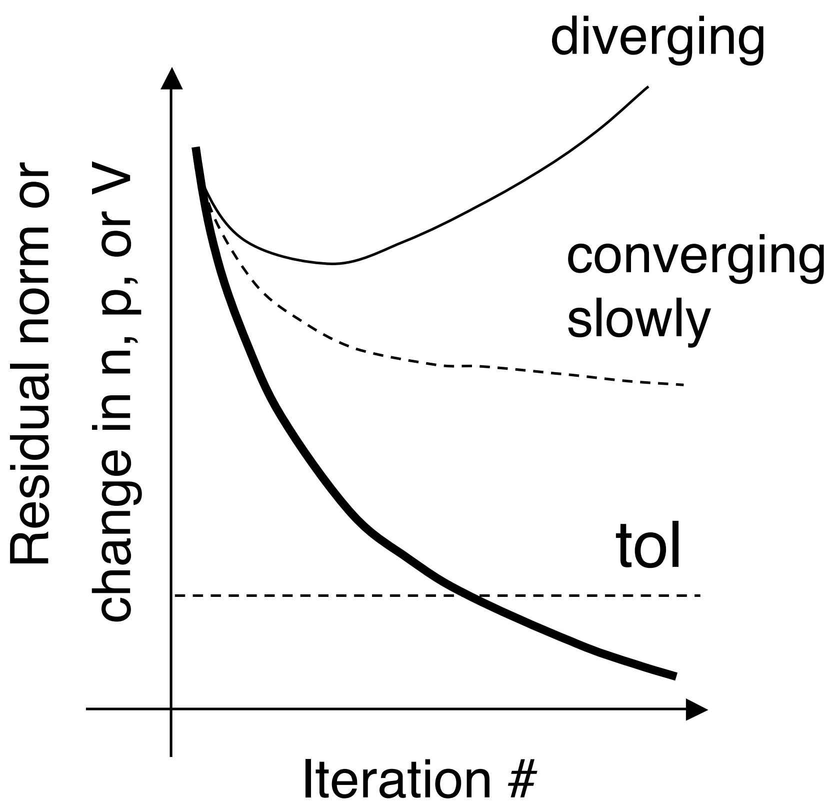
Is a measure of the numerical error

2)

$$\Delta V^k = V^{k+1} - V^k$$

ΔV^k --> 0 as k --> oo

3) Convergence



Convergence tips:

- check problem definition
- take small steps in voltage
- increase k_{max} if converging
- change convergence criterion
- try another method

3) Numerical Solution: Summary

Summary: Solving Partial Differential Equations

1) Begin with a set of equations and boundary conditions

2) Discretize the equations on a grid with N nodes to obtain 3N nonlinear equations in 3N unknowns

3) Solve the system of nonlinear equations by iteration

4) Physical Models

The physical parameters in the semiconductor equations need to be modeled.
e.g.

1) doping dependent mobility

$$\mu = \frac{\mu_i}{1 + N_D / N^{\alpha}}$$

2) field dependent mobility

$$\mu = \frac{\mu_o}{\sqrt{1 + E / E_{cr}}}$$

3) recombination

$$R = \frac{np - n_i^2}{(n + n_1)\tau_{po} + (p + p_1)\tau_{no}}$$

4) etc.

4) Physical Models: Example

MINIMOS physical parameters (see Ch. 2 of manual)

1) doping, field, and temperature dependent mobility
2) SRH recombination
3) impact ionization
4) band-to-band tunneling
5) interface and traps
6) intrinsic carrier concentration
7) hot carrier transport model parameters
8) Monte Carlo transport model parameters

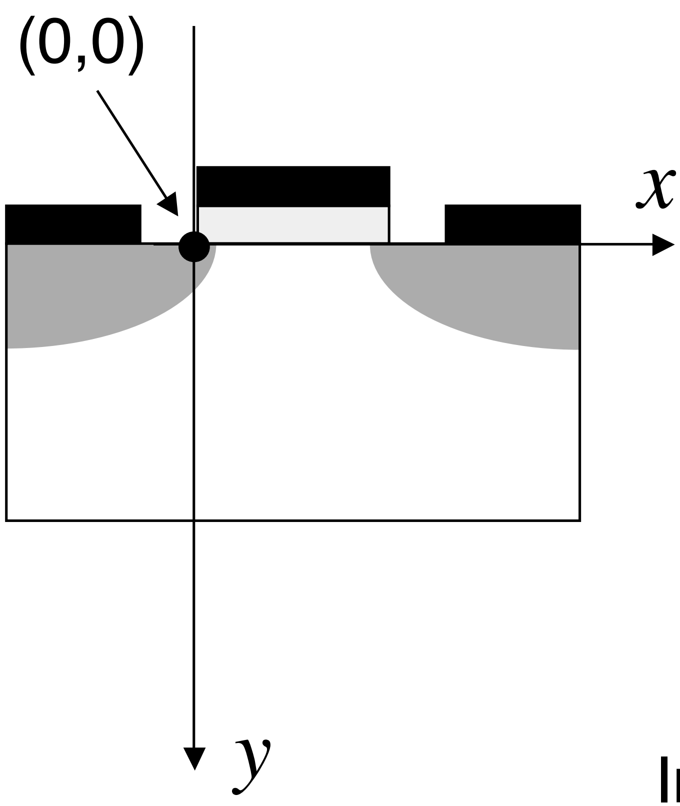
4) Physical Models: Tips

Tips for dealing with physical models

- **understand** the models available in the tool
- **understand** the parameters in the model you select
- **know** the default models and their parameters
- **check** for conflicts between various models
  (i.e. if model A is selected, model B can't be used)

Proper selection and specification of physical models is critical!

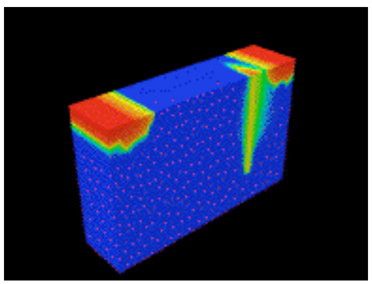
5) Example: The MINIMOS program




```
* EXAMPLE  MINIMOS  6.0 SIMULATION
DEVICE     CHANNEL=N  GATE=NPOLY
+           TOX=150.E-8 W=1.E-4  L=0.85E-4
BIAS        UD=4. UG=1.5
PROFILE     NB=5.2E16  ELEM=AS DOSE=2.E15
+           TOX=500.E-8  AKEV=160.
+           TEMP=1050.  TIME=2700
IMPLANT     ELEM=B  DOSE=1.E12  AKEV=12
+           TEMP=940  TIME=1000
OPTION      MODEL=2-D
OUTPUT      ALL=YES
END
```

Input directives are described in Ch. 3 of the MINIMOS 6.0 User's Guide

5) Example: The PADRE program



M. Pinto, R.K. Smith, M.A. Alam, Bell Labs

```
title      MOSFET - NMOS
MESH RECT NX=51NY=51
X. M  N=1 LOC=0
X. M  N=15 LOC=0.05 RATIO=0.8
X. M  N=26 LOC=0.0625 RATIO=1.25
X. M  N=36 LOC=0.075 RATIO=0.8
X. M  N=51 LOC=0.125 RATIO=1.25

Y.M N=1 LOC=0
Y.M N=25 LOC=0.068 RATIO=0.8
Y.M N=36 LOC=0.0805 RATIO=1.25
Y.M N=46 LOC=0.093 RATIO=0.8
Y.M N=51 LOC=0.0942 RATIO=1.25

# Substrate
REGION NUM=1 ix.l=1 ix.h=51 iy.l=1 iy.h=25 silicon
# Source
REGION NUM=2 ix.l=1 ix.h=15 iy.l=25 iy.h=46 silicon
# Drain
REGION NUM=3 ix.l=36 ix.h=51 iy.l=25 iy.h=46 silicon
# Channel
REGION NUM=4 ix.l=15 ix.h=36 iy.l=25 iy.h=46 silicon
# Gate
REGION NUM=5 ix.l=15 ix.h=36 iy.l=46 iy.h=51
…
```

Where to get more information

1) "MINIMOS - A Two-Dimensional MOS Transistor Analyzer," by S. Selberherr, A. Schutz, and H.W. Potzl, *IEEE Transactions on Electron Devices*, Vol. ED-27, pp. 1540-1550, 1980

2) MINIMOS 6.0 User's Guide, October, 1994 (available from the MINIMOS page of the the nanoHUB: www.nanohub.org)

3) Analysis and Simulation of Semiconductor Devices, S. Selberherr, Springer-Verlag, New York, 1984. (discusses numerical methods)

4) Padre User's Guide (available from the Padre page of the nanoHUB)

Tips on using a new simulation tool

- Understand what the tool does
  - -what equations are being solved?
  - -what numerical methods are used?
  - -what physical models are implemented?
- Try a simple problem first to be sure you get the correct answer
- Look for example files - close to the problem you're interested in.
- Know what the default settings are
- Ask an experienced user for help

some thoughts on modeling and simulation

Some views on modeling and simulation

Many members of the Spice generation merely hack away at design. They *guess* at circuit values, run a simulation, and then *guess* at changes before they run the simulation again…..and again…..and again. Designers need an ability to create a simple and correct model to describe a complicated situation - designing on the back of an envelope. The back of the envelope has become the back of a cathode ray tube, and intuition has *gone* on vacation.

Paraphrased from:

Ronald A. Rohrer, "Taking Circuits Seriously," *IEEE Circuits and Devices*, July, 1990.

another view on modeling and simulation

"All software begins with some fundamental assumptions that translate into fundamental limitations, but these are not always displayed prominently in advertisements. Indeed, some of the limitations may be equally unknown to the vendor and to the customer. Perhaps the most damaging limitation is that software can be misused or used inappropriately by an inexperienced or overconfident engineer."

Henry Petroski, "Failed Promises," *American Scientist*, 82(1), 6-9 (1994)

stand up to a computer!

The use of sophisticated computer simulation tools is a growing component of modern engineering practice. These tools are unavoidably based on numerous assumptions and approximations, many of which are not apparent to the user and may not be fully understood by the software developer. But even in the face of these inherent uncertainties, computer simulation tools can be a powerful aid to the engineer. Engineers need to develop an ability to derive insight and understanding from simulations. They must be able to "**stand up to a computer**"and reject or modify the results of a computer-design when dictated to do so by *engineering judgement*.

Paraphrased from:
Eugene S. Fergusson, *Engineering in the Mind's Eye*, MIT Press (1993)

My Compute Lied To Me

Bob Pease
analog circuit designer
National Semiconductor

(after his computer "lied"
To him)

how to use a simulation program

"The basic difference between an ordinary TCAD user and an true technology designer is that the former is relaxed, accepting on faith the program's results, the latter is concerned and busy checking them in sufficient depth to satisfy himself that the software developer did not make dangerous assumptions. It takes years of training in good schools, followed by hands-on design practice to develop this capability. It cannot be acquired with short courses, or with miracle push-button simulation tools that absolve the engineer of understanding in detail what he is doing."

Paraphrased from:

Constantin Bulucea, "Process and Device Simulation in the Era of Multi-Million-Transistor VLSI - A Technology Developer's View," IEEE Workshop on Simulation and Characterization, Mexico City, Sept. 7-8, 1998.

Final thought on modeling and simulation

"The purpose of computing is insight, not numbers."

R. W. Hamming

Thank you!

Please rate this tutorial on nanohub.org and send comments.