The Two-Level Newton Method and its Application to Electronic Simulation.

Eric Keiter, Scott Hutchinson,
Rob Hoekstra, Tom Russo, Eric Rankin and Lon Waters
Sandia National Laboratories
Albuquerque, NM

LLNL Workshop on Solution Methods
for Large-Scale Nonlinear Problems
August 7th, 2003

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy under contract DE-AC04-94AL85000.

Outline

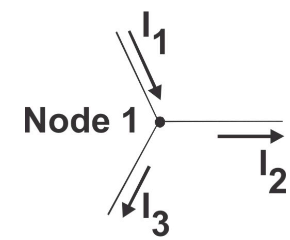
- Circuit Simulation
- PDE Device Simulation
- Motivation
- Multi-Level Newton solver background/history
- Coupling algorithms results
- Future Work/Conclusions

Analog Circuit Simulation

- System of Coupled DAE's
- Kirchoff's Laws, Arbitrary Network
- Formulation: Modified Nodal Analysis. (modified KCL)
  - Most equations are Kirchoff Current Law (KCL) equations.
  - Most solution variables are nodal voltages.
  - Most currents are obtained via an Ohm's law relationship.
- Arbitrary network, so resulting Jacobian matrices are often very ill-conditioned.
- Iterative linear solvers are possible (with right preconditioning), but direct solvers are a lot easier.
- Solving the nonlinear problem, using Newton's method, can often be enhanced with voltage limiting.
- Voltage limiting introduces non-smoothness into the problem.
- Steady state much more difficult to solve than transient.

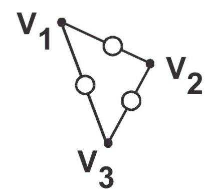
- Most of the Jacobian terms in a (modified KCL) circuit problem are conductances.

$$\sum_{i=0}^{n} (I_n) = 0$$



Kirchoff's Current Law (KCL)

$$\sum_{i=0}^{n} (\Delta V) = 0$$



○ = Device

• = Node

Kirchoff's Voltage Law (KVL)

$$G = \frac{\delta I}{\delta V} = \text{conductance} = \text{Jacobian term}$$

$$\text{Ohm's Law: } I = GV = V/R$$

PDE Device Simulation

- Mesh Based, 1-3D.
- Finite Difference or Finite Element.
- Coupled set of PDE's.
- Drift-Diffusion formulation
- 3 Variables:
  - electron density
  - hole density
  - electrostatic potential.
- Scaling the equations is important.
- Spatial discretization needs some kind of stabilization(SG or SUPG).
- Iterative linear solvers are usually successful.
- Nonlinear solve obtained from Newton's method. Quadratic line search works pretty well.

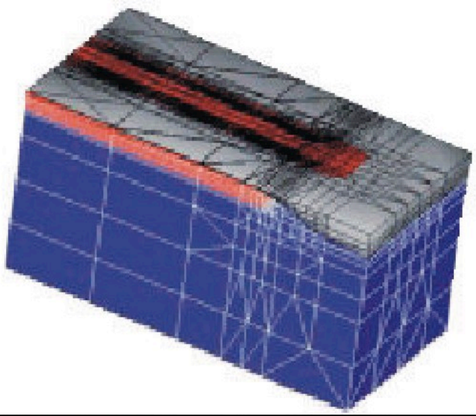
$$-\nabla \cdot (\epsilon \nabla \phi(x)) = \rho(x)$$

$$\frac{\partial n(x)}{\partial t} + \nabla \cdot \Gamma_n = -R(x)$$

$$\frac{\partial p(x)}{\partial t} + \nabla \cdot \Gamma_p = -R(x)$$

$$\Gamma_n = n(x)\mu_n E(x) + D_n \nabla n(x)$$

$$\Gamma_p = p(x)\mu_p E(x) + D_p \nabla p(x)$$



3D MOSFET mesh from DaVinci

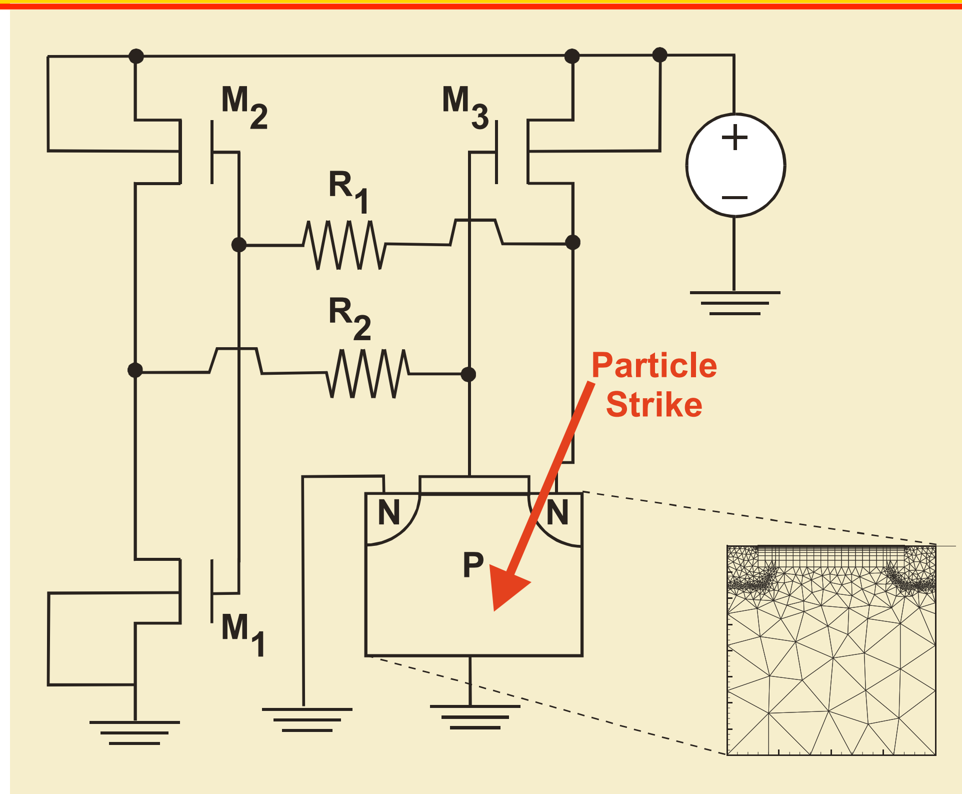
Motivation: PDE Device-Circuit Coupling

- **Some PDE device problems, (see figure), are only meaningful if coupled to an external circuit.**

- **Different optimal solution techniques:**

Simulation Type	Best Linear Solver	Best Nonlinear Solver Enhancement
Circuit	Direct	Voltage Limiting
PDE Device	Iterative	Line Search

- **Ideal solution method:** Two-level Newton?
  - Separate the two calculations:
  - Maintain tight coupling.



Single-Event-Upset (SEU) of an SRAM memory cell

Full Newton Method

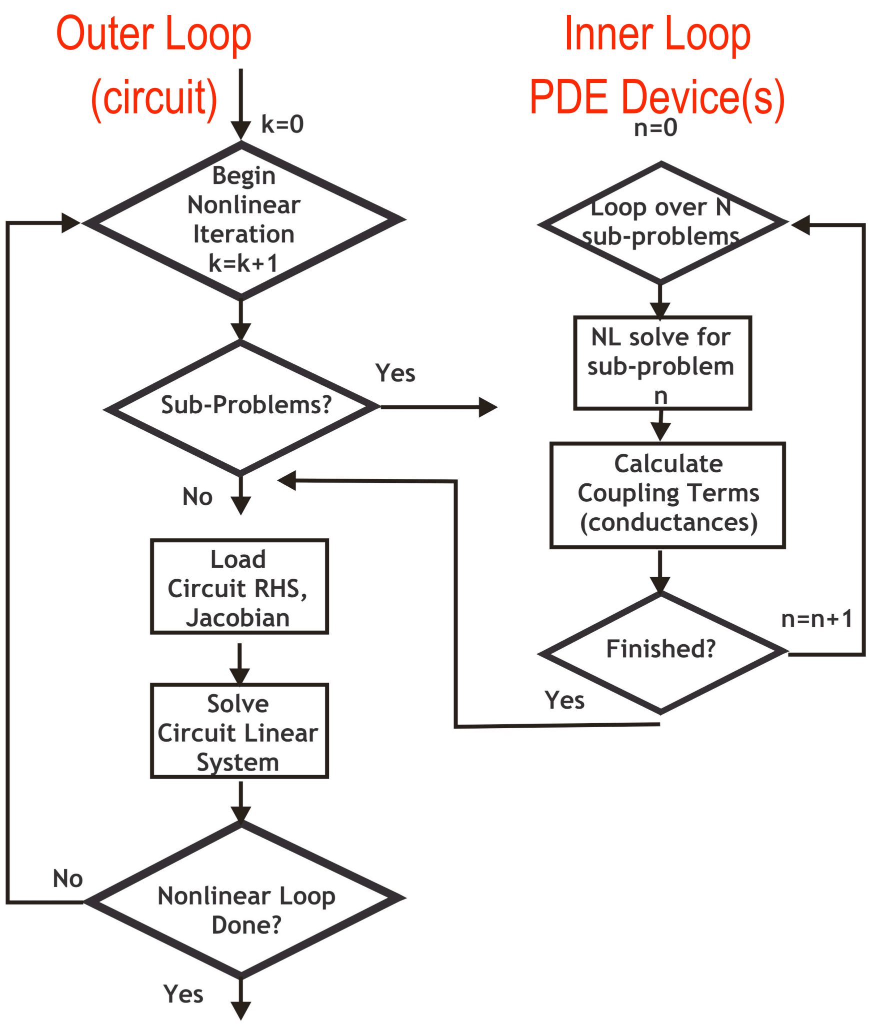
- Want $\mathbf{x}$, the solution, such that $\mathbf{F}(\mathbf{x}) = 0$

- Iteration:

$$\mathbf{J}(\mathbf{x})\Delta\mathbf{x} = -\mathbf{F}(\mathbf{x})$$
$$\mathbf{x}^{k+1} = \mathbf{x}^{k} + \Delta\mathbf{x}$$

- Many variations on this approach.

Two-Level Newton



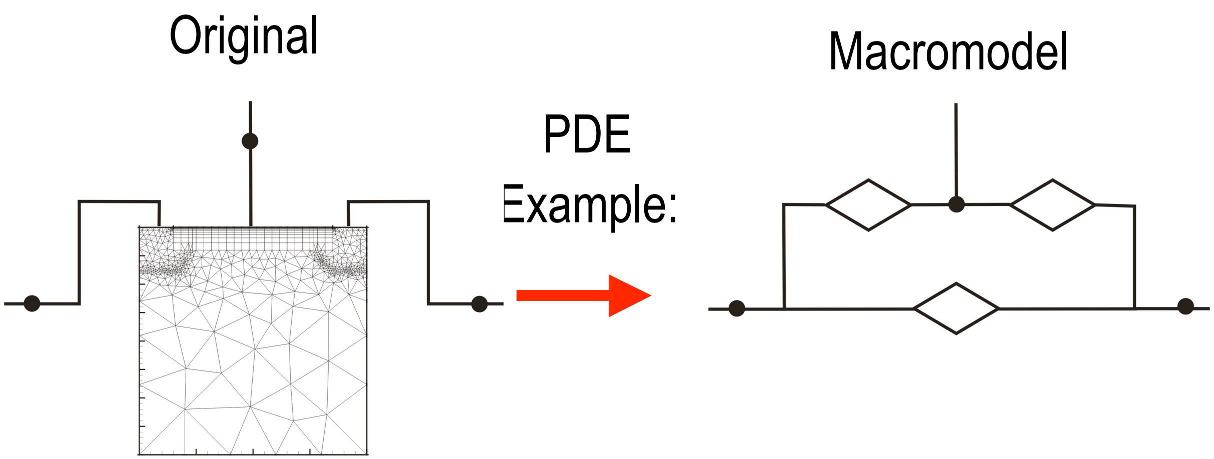
- Two-Level consists of 2 nested Newton loops.
- Outer Loop = circuit problem
- Inner Loop = PDE Device problem.
- Choice of NL algorithm at one level is largely independent of the choice at another level.
- Inner problem(s) solved to convergence at each outer loop iteration.
- Problem is converged when both inner and outer problems are converged, consistent.
- Easier (possible?) to implement if both levels have a Jacobian.
- As with any nonlinear algorithm, you want to get $\mathbf{F}(\mathbf{x}) = 0$

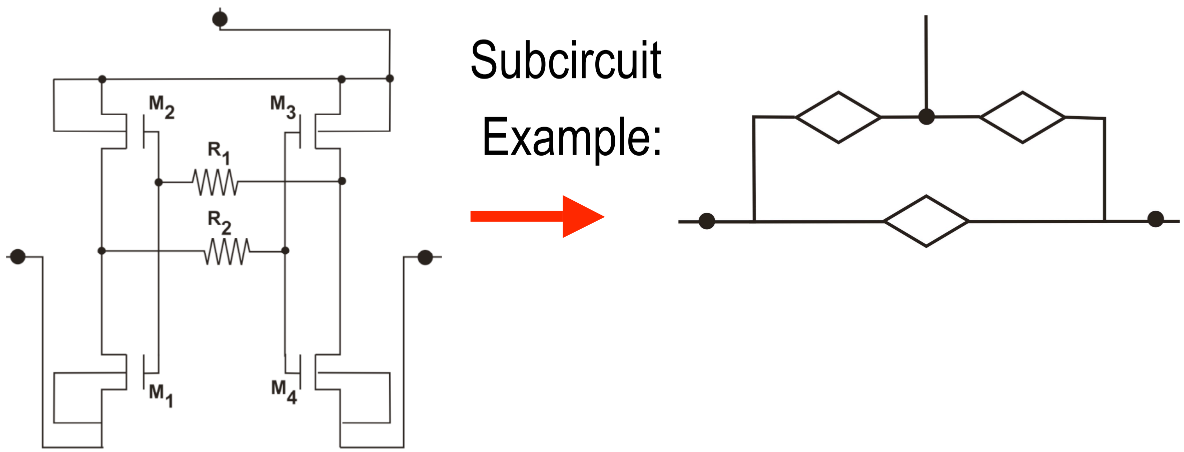
Two-Level Newton Background

- Proposed by Rabbat in 1979, as an enhancement to circuit simulation, in which the repetitive nature of digital circuits could be exploited.
- Subcircuits, of which many identical copies may exist, are replaced by equivalent, much smaller, macromodels.
- Each subcircuit is simulated independently, by itself, and the results used to create macromodel.
- Algorithm depends upon the connections between a subcircuit and the main circuit to exist only through a small number of terminals.
- Local quadratic convergence properties.
- Mayaram (1988) demonstrated that this method could also be used for coupled circuit-device simulation.
- Not much published since the late 80's, but both circuit simulation and device simulation papers have moved towards using Natural Parameter Homotopy

Macromodel Concept

- Macromodel replaces original entity (PDE device or subcircuit) at the nonlinear solver level.
- Inner loop solves the original entity to convergence.
- Outer loop uses the macromodel.
- Small number of connecting variables/nodes.





Solution, Residual Terms

$\mathbf{x}$ = Full Problem Solution Variables $\mathbf{v}$ = Outer Problem Solution Variables $\mathbf{w}$ = Inner Problem Solution Variables	$\mathbf{x} = \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{v_{nc}} \\ \mathbf{v_c} \\ \mathbf{w_{nc}} \\ \mathbf{w_c} \end{bmatrix}$
$\mathbf{v_c}, \mathbf{w_c}$ = Coupling Variables $\mathbf{v_{nc}}, \mathbf{w_{nc}}$ = Non-Coupling Variables	
$\mathbf{F}(\mathbf{x})$ = Full Problem Residual $\mathbf{F_{out}}(\mathbf{v}, \mathbf{w_c})$ = Outer Problem Residual $\mathbf{F_{in}}(\mathbf{w}, \mathbf{v_c})$ = Inner Problem Residual	$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} \mathbf{F_{out}}(\mathbf{v}, \mathbf{w_c}) \\ \mathbf{F_{in}}(\mathbf{w}, \mathbf{v_c}) \end{bmatrix}$

- For sake of simplicity, assuming one sub-problem! (the inner problem)
- If $\mathbf{F_{out}}(\mathbf{v}, \mathbf{w_c})$ and $\mathbf{F_{in}}(\mathbf{w}, \mathbf{v_c})$ are converged, and they are consistent, then (in some sense) $\mathbf{F}(\mathbf{x})$ is also converged.
- Circuit-device coupling: the inner problem is a PDE device, $\mathbf{v_c}$ are connected circuit node voltages, applied as boundary conditions, and $\mathbf{w_c}$ are currents coming from the PDE device which are applied to the circuit.

Jacobian Terms

$$
\begin{aligned}
\mathbf{J}(\mathbf{x}) &= \text{Full Problem Jacobian} \\
\mathbf{J}_{\mathbf{out}}(\mathbf{v}, \mathbf{w}_{\mathbf{c}}) &= \text{Outer Problem Jacobian} \\
\mathbf{J}_{\mathbf{in}}(\mathbf{w}, \mathbf{v}_{\mathbf{c}}) &= \text{Inner Problem Jacobian}
\end{aligned}
$$

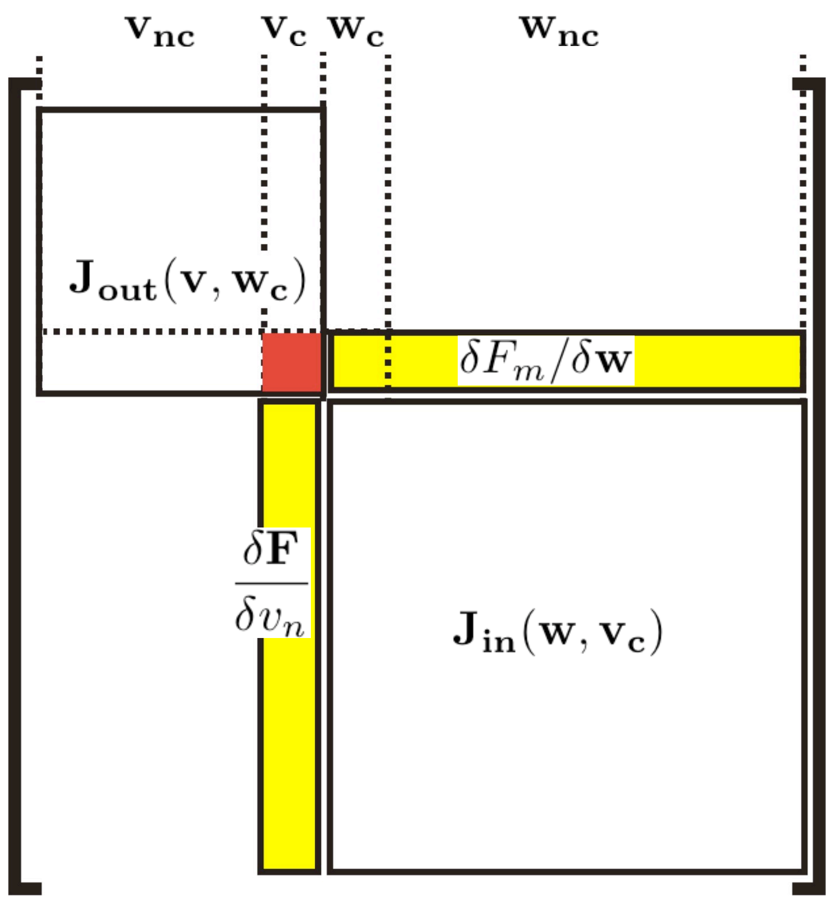
- $\mathbf{J}_{\mathbf{in}}(\mathbf{w}, \mathbf{v}_{\mathbf{c}})$ is a sub-block of $\mathbf{J}(\mathbf{x})$
- $\mathbf{J}_{\mathbf{out}}(\mathbf{v}, \mathbf{w}_{\mathbf{c}})$ is also, a sub-block, but it needs some modification in the red part.
- For each pair of variables in $\mathbf{v}_{\mathbf{c}}$, compact conductance is given by:

$$
\frac{\delta F_m}{\delta x_n} = \frac{\delta F_m}{\delta v_n} + \frac{\delta F_m}{\delta \mathbf{w}} \frac{\delta \mathbf{w}}{\delta v_n}
$$

$$
\frac{\delta \mathbf{w}}{\delta v_n} = -\mathbf{J}_{\mathbf{in}}^{-1} \frac{\delta \mathbf{F}}{\delta v_n}
$$

- So, for each variable in $\mathbf{v}_{\mathbf{c}}$, a linear solve is required.

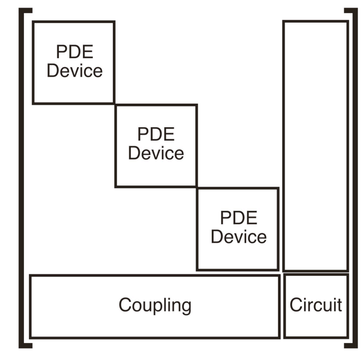
$\mathbf{J}(\mathbf{x}) =$



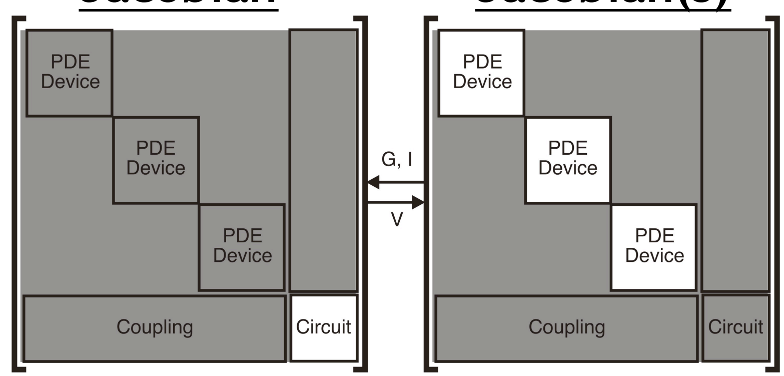
Prototype Implementation in Xyce™

- Xyce is a Sandia-developed analog circuit simulator: distributed or serial, Object-oriented(OO), C++.
- Devices sit behind a common, base class, OO interface.
- 1D and 2D PDE devices sit behind this interface as well, so to the rest of the code, they don't appear any different than an analog circuit device, like a capacitor, or diode.
- Prototype for Charon, a 3D device simulator.
- This made setting up the full Newton algorithm very straightforward.
- For 2-level, same linear system Petra objects are used (Jacobian, RHS vector, solution vector, etc.), but different "views" of the problem are solved, depending on the phase of the solution.

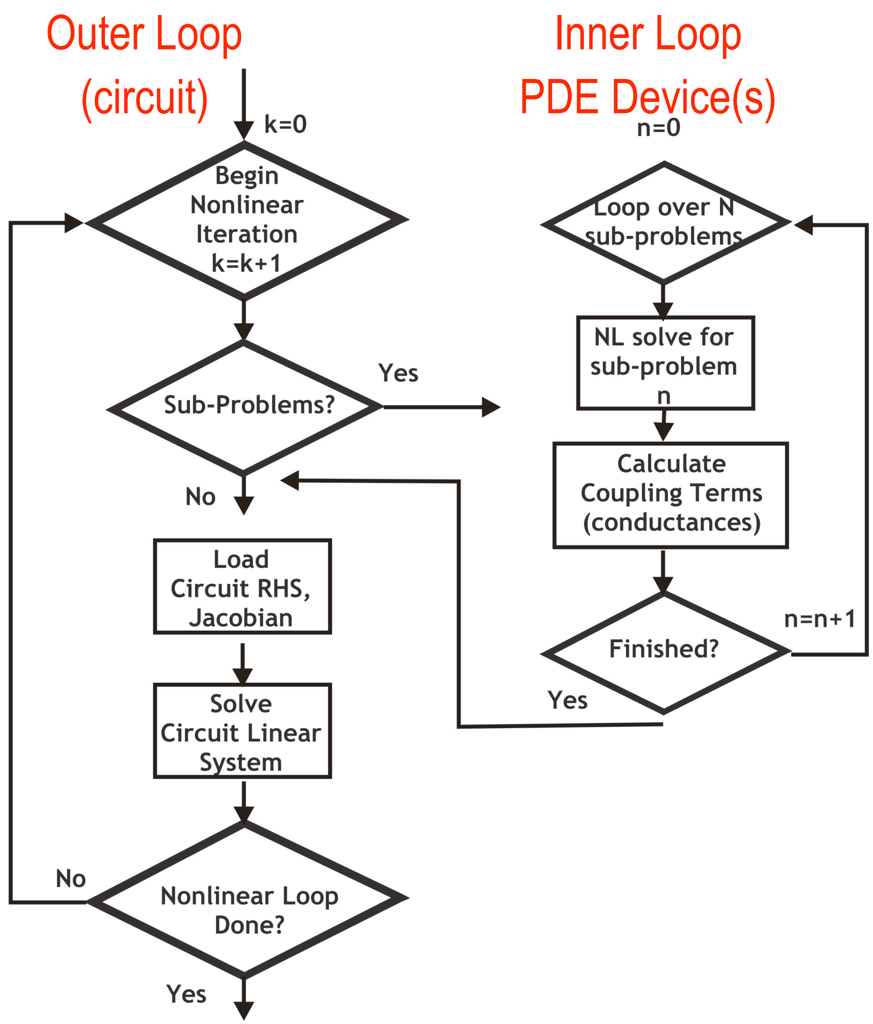
Full Newton Jacobian



Circuit Jacobian **PDE Jacobian(s)**



Two-Level Newton



- Two-Level consists of 2 nested Newton loops.
- Outer Loop = circuit problem
- Inner Loop = PDE Device problem.
- Choice of NL algorithm at one level is largely independent of the choice at another level.
- Inner problem(s) solved to convergence at each outer loop iteration.
- Problem is converged when both inner and outer problems are converged, consistent.
- Easier (possible?) to implement if both levels have a Jacobian.
- As with any nonlinear algorithm, you want to get $\mathbf{F}(\mathbf{x}) = 0$

Two-Level Newton Variations

- At each outer loop iteration (especially early ones) the outer loop variables (voltages) may change a great deal, possibly enough that the inner loop solve has difficulty converging.

- Modified 2-level Newton (Mayaram). Use forward-Euler formula to predict $\mathbf{w}^{k+1}$

$$\mathbf{w}^{k+1} = \mathbf{w}^{k} + \frac{\delta \mathbf{w}}{\delta v_n} \Delta v_n$$

- Natural parameter continuation (Xyce). Do continuation from $\mathbf{w}^{k}$ to $\mathbf{w}^{k+1}$
  – For the inner loop PDE simulation, this effectively means slowly varying the dirichlet boundary conditions on the electrostatic potential
  – Generally, the potential BCs need to change by an amount between the thermal voltage (0.025V) and the bandgap (1.1V). Much more than that, and the PDE device has a lot of trouble converging.

- These two approaches can be done together - they aren't incompatible.

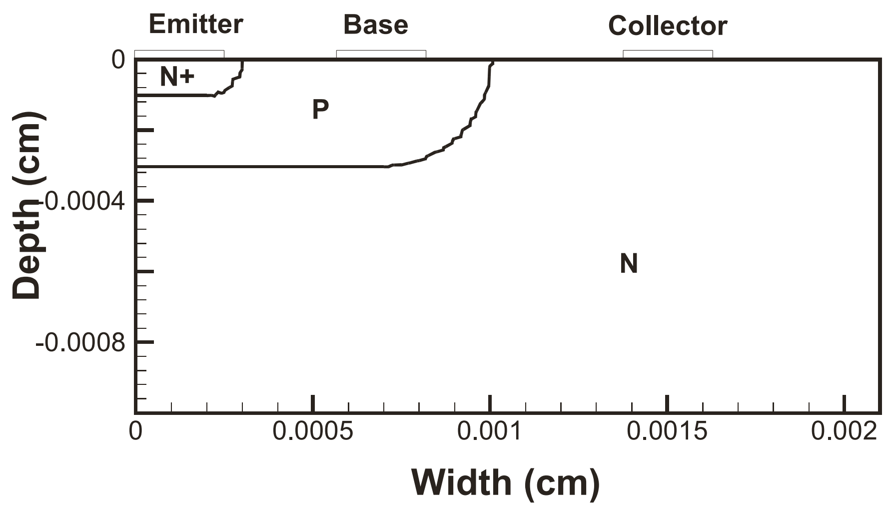
Two-Level Newton Costs/Benefits

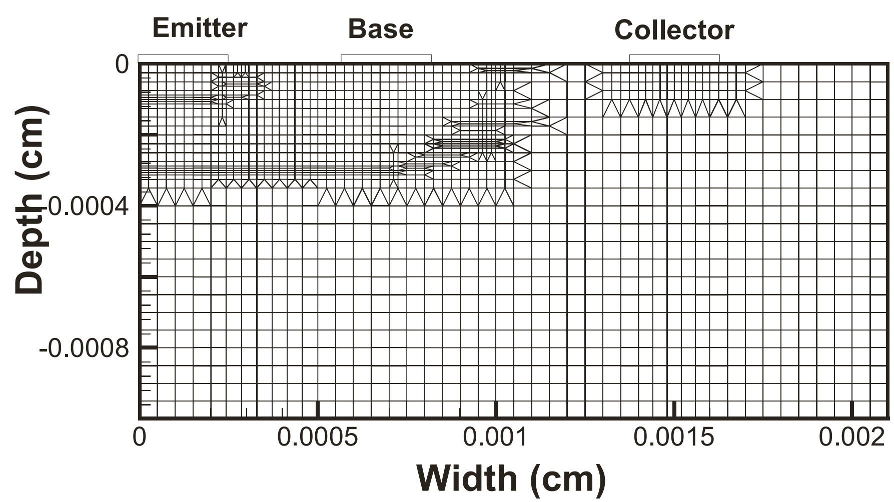
- Costs:
  - Relatively slow. Complete inner NL solve at each outer step.
  - Extra linear solves, to calculate coupling terms - means you need a small number of coupling variables/terms.

- Benefits
  - Very robust.
  - Very flexible.
  - Well suited to multi-physics simulations which use two separate simulation codes. (one for circuit, one for device, for example)

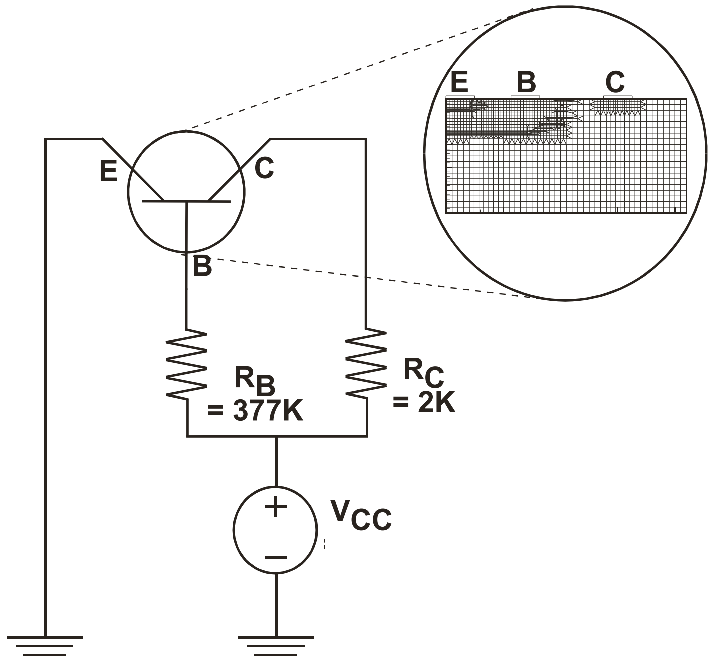
Example Problem: NPN BJT

- Mesh created/refined based on doping profile and electrode contacts.
- Doping (per ccm)
  - N+ region: 1e19
  - P region: -1e16
  - N region: 1e14
- This device (and its inverse, a PNP) is used in most of the examples to follow.





Full Newton vs. 2-Level, Example 1: Easy circuit, Easy PDE Device



Method	Outer Newton Steps	Inner Newton Steps	Continuation Steps	Linear Solves
Full	15	NA	NA	15
Two-Level	2	16	NA	24
Full, IG	1	NA	NA	1
Two-Level,IG	1	1	NA	4

- One PDE Device: NPN BJT in equilibrium.
- Easy circuit, Easy device
- Steady state
- Circuit is linear, BJT unbiased: Vcc=0.
- Full Newton, and Inner Newton use quadratic line search.
- Outer loop is straight Newton.

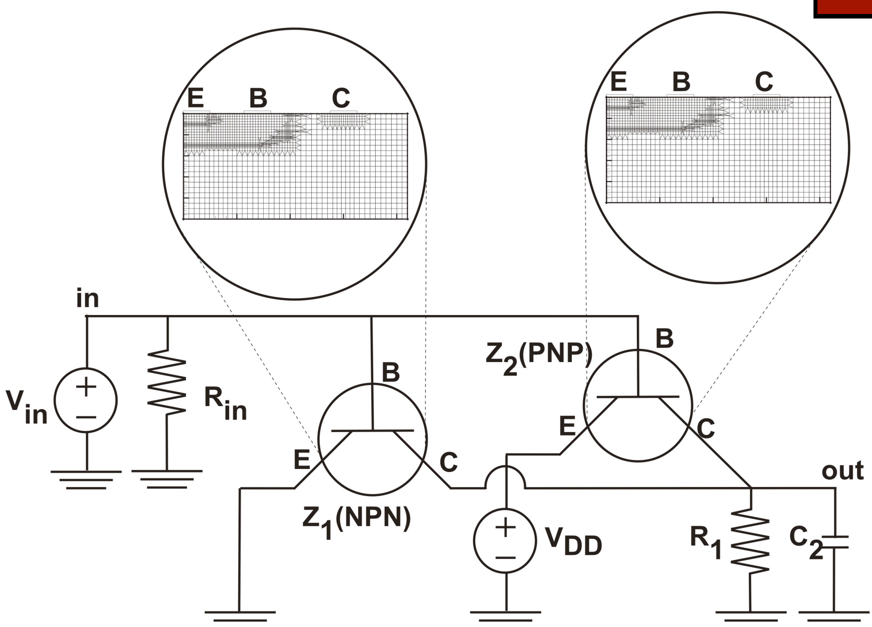
IG = initial guess (nonlinear Poisson)

C = continuation

VL = voltage limiting

Full Newton vs. 2-Level, Example 2: Easy circuit, Difficult PDE Device

Method	Outer Newton Steps	Inner Newton Steps	Continuation Steps	Linear Solves
Full, IG	-	NA	NA	NA
Two-Level, IG	-	NA	NA	NA
Two-Level, IG, C	16	358	83	422
Two-Level, IG, C, VL	7	276	49	304



- BJT Inverter
- 2 PDE Devices: NPN BJT, PNP BJT
- Easy circuit, Hard Device b/c one BJT always on
- Steady state
- Full Newton, and Inner Newton use quadratic line search.
- For success, Inner solve needs continuation.
- Outer loop is straight Newton w/voltage limiting.

IG = initial guess (nonlinear Poisson)
C = continuation
VL = voltage limiting

Full Newton vs. 2-Level, Example 3: Difficult Circuit, Difficult PDE Device

Method	Outer Newton Steps	Inner Newton Steps	Continuation Steps	Linear Solves
Full, IG	-	NA	NA	NA
Two-Level, IG	-	NA	NA	NA
Two-Level, IG, C	-	-	-	-
Two-Level, IG, C, VL	15	86	28	144

(NO PICTURE)
11 SPICE BJT's
1 PDE BJT.

- RCA3040 Amplifier
- PDE Device: one NPN BJT
- Hard circuit, Hard Device
- Steady state.
- (Circuit contains many nonlinear SPICEModels, and BJT is on)
- Full Newton, and Inner Newton use quadratic line search.
- For success, Inner solve needs continuation.
- Outer loop is straight Newton w/voltage limiting.

IG = initial guess (nonlinear Poisson)
C = continuation
VL = voltage limiting

Comparison to Full Problem Continuation (Example 3, redux)

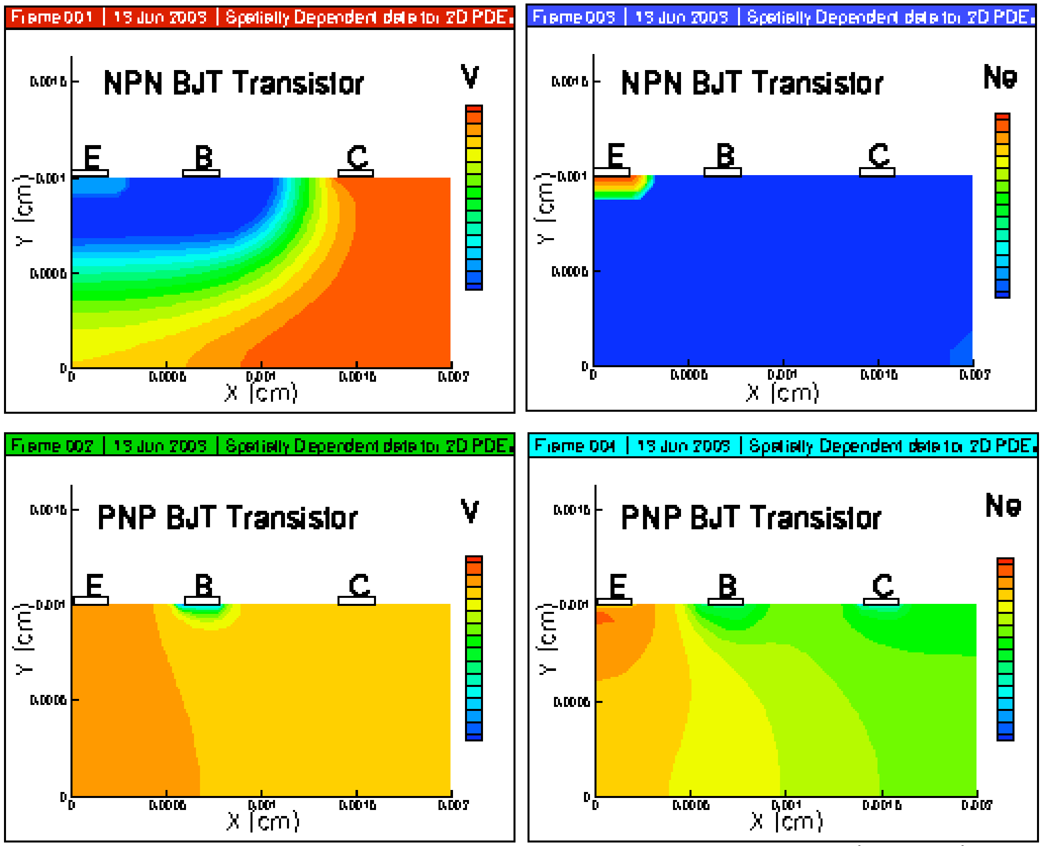
- Circuit: RCA3040 Amplifier
  - Hard circuit (same as previous slide)
  - Hard Device
  - (Circuit contains many nonlinear SPICE Models, and BJT is on)

(NO PICTURE)
11 SPICE BJT's
1 PDE BJT.

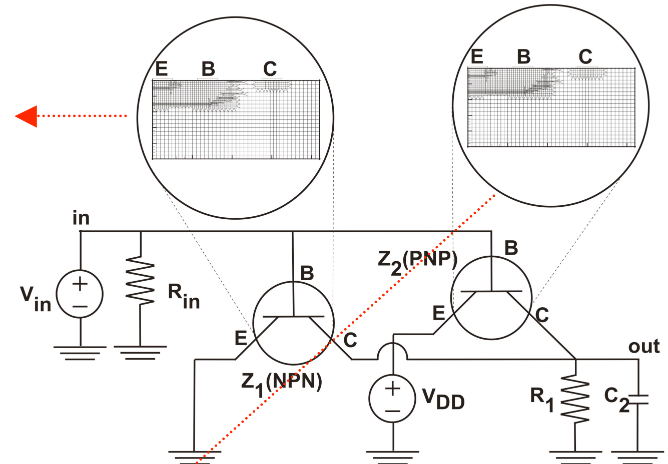
- Two-Level Newton result:
  – Successful convergence, 144 linear solves, 110 seconds total time.

- Natural parameter continuation: (very preliminary!)
  – Continuation parameters: voltage source magnitudes: 0 - max.
  – To get this to work, had to reduce the size of the voltage sources.
  – Successful convergence, 247 linear solves, 176 seconds total time.

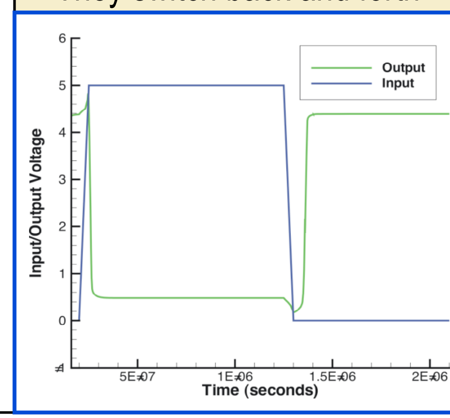
Transient Result: Inverter Circuit



Electrostatic Potential	Electron Density (log plot)
Max V = 5.5 Volts	Max Ne = 1.0e+19 per ccm
Min V = 0.0 Volts	Min Ne = 1.0e+14 per ccm



One BJT is on, the other off.
They switch back and forth



Conclusions

- Multi-level Newton method is an effective method for coupled circuit-device simulation.
- Allows different phases of a problem to be handled using methods that are well-suited for each phase.
- Could be applied to a variety of multi-physics problems, assuming:
  - Each sub-problem has a Jacobian
  - Coupling between sub-problems is limited to a small number of variables
- Seems to compare favorably to natural parameter continuation on the full problem, but this is a very preliminary result.
- Natural parameter continuation is very helpful when applied to the inner problem, will probably also be helpful to the outer problem.

Future Work

- Generalized version of 2-level Newton to be put into NOX.
- More comparison to natural parameter homotopy (via LOCA).
- Application to problems involving two separate application codes (Xyce and Charon)
- Application to hierarchical circuit simulation and/or waveform relaxation.
- Application to non-electrical problems:
  - Bio-Xyce (inter-cellular reaction/intra-cellular diffusion)
  - Charon (reacting flow/surface chemistry)
  - Adagio (MEMs contact simulation)

Acknowledgements

- The entire Xyce™ circuit simulator team:
  - Scott Hutchinson, Rob Hoekstra, Tom Russo, Eric Rankin, Lon Waters, Richard Sheik, Et. Al.
- NOX people: Roger Pawlowski, Tammy Kolda.
- CHARON device simulator: Gary Hennigan