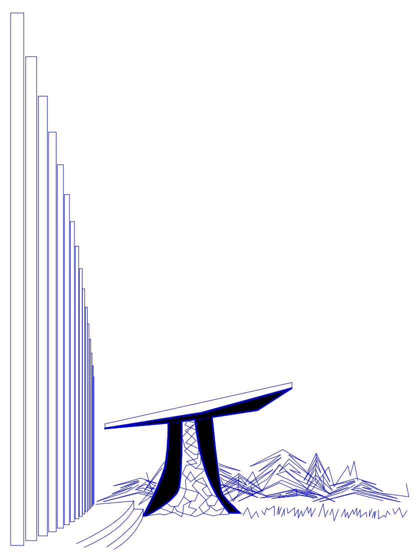
Efficient Time–Domain Simulation and Optimization of Digital FET Circuits

Andrew R. Conn
Ruud A. Haring
Chandu Visweswariah

Thomas J. Watson Research Center
Yorktown Heights, NY 10598

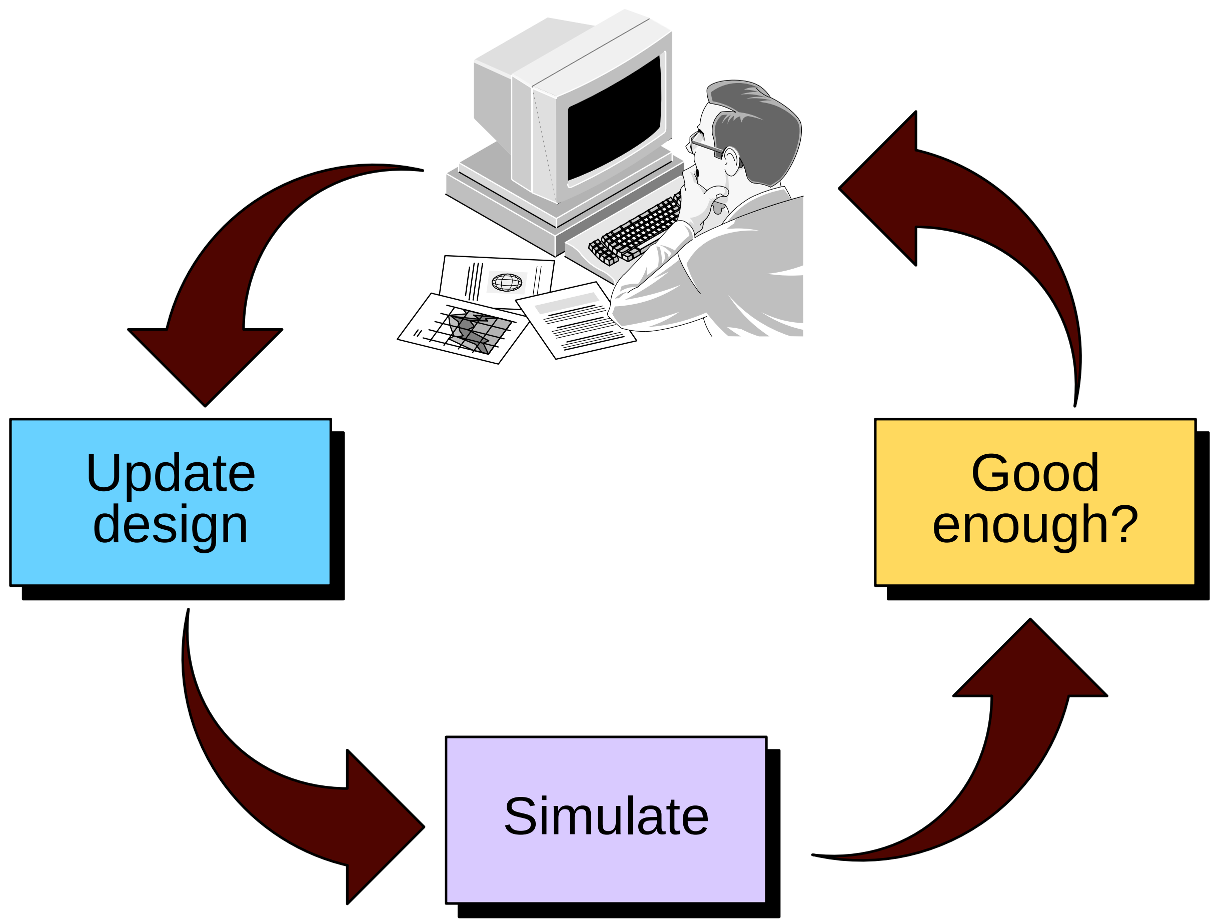


Outline

1. Motivation
2. Circuit analysis (SPECS)
3. Time–domain gradient computation (SPECS)
4. Circuit optimization (JiffyTune with LANCELOT)
5. Conclusions and future work

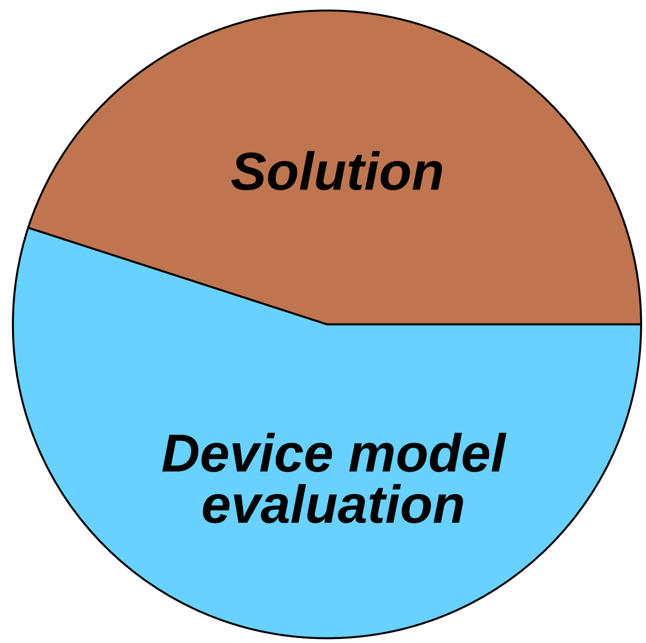
Part 1: Motivation

- *optimization* of custom, high–performance digital circuits is usually a tedious, manual process
- the time spent in closing the loop on delay, area and power requirements is too long



Motivation

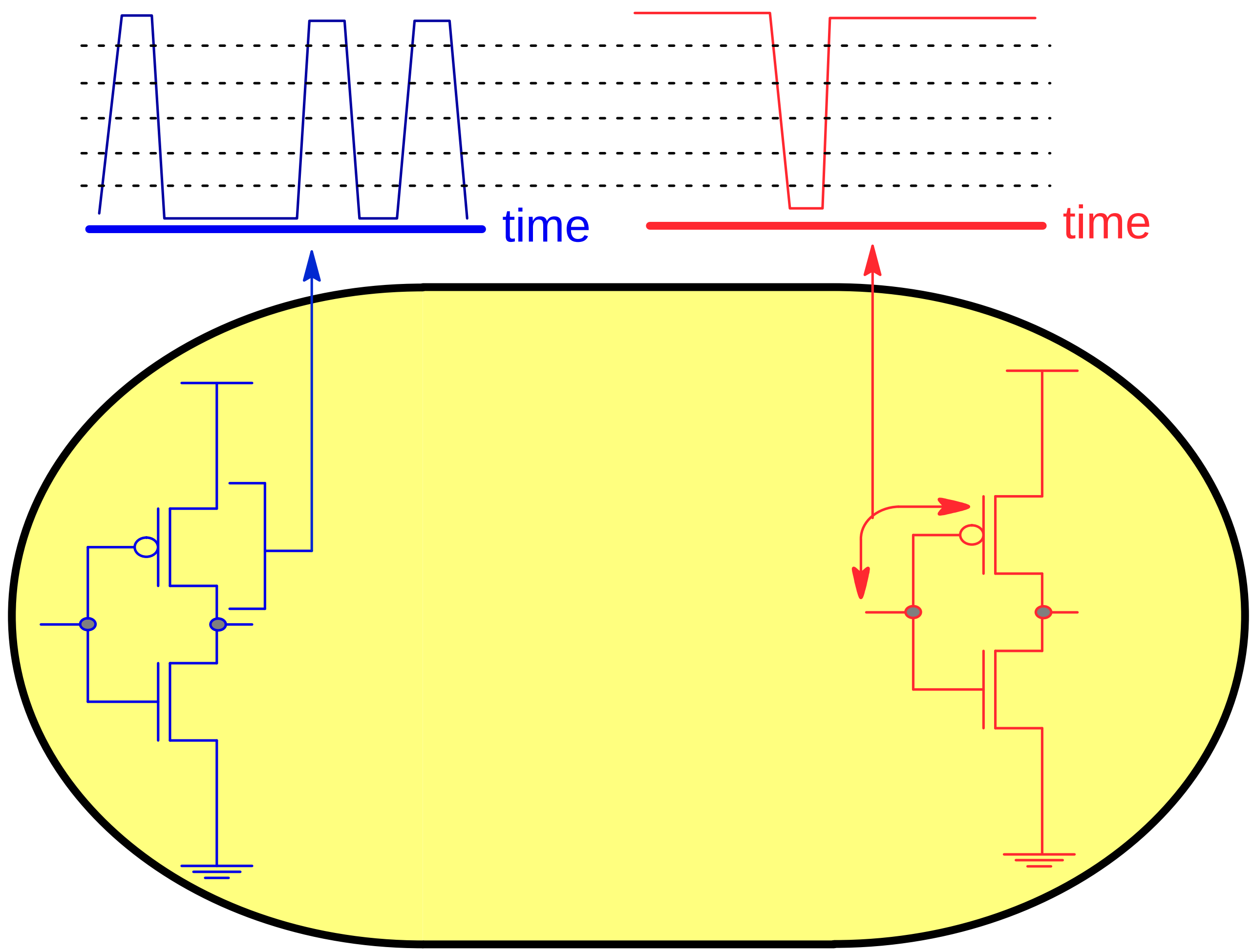
- circuit analysis is computationally expensive
  - evaluation of device modeling equations
  - equation solution



- must attack both segments!

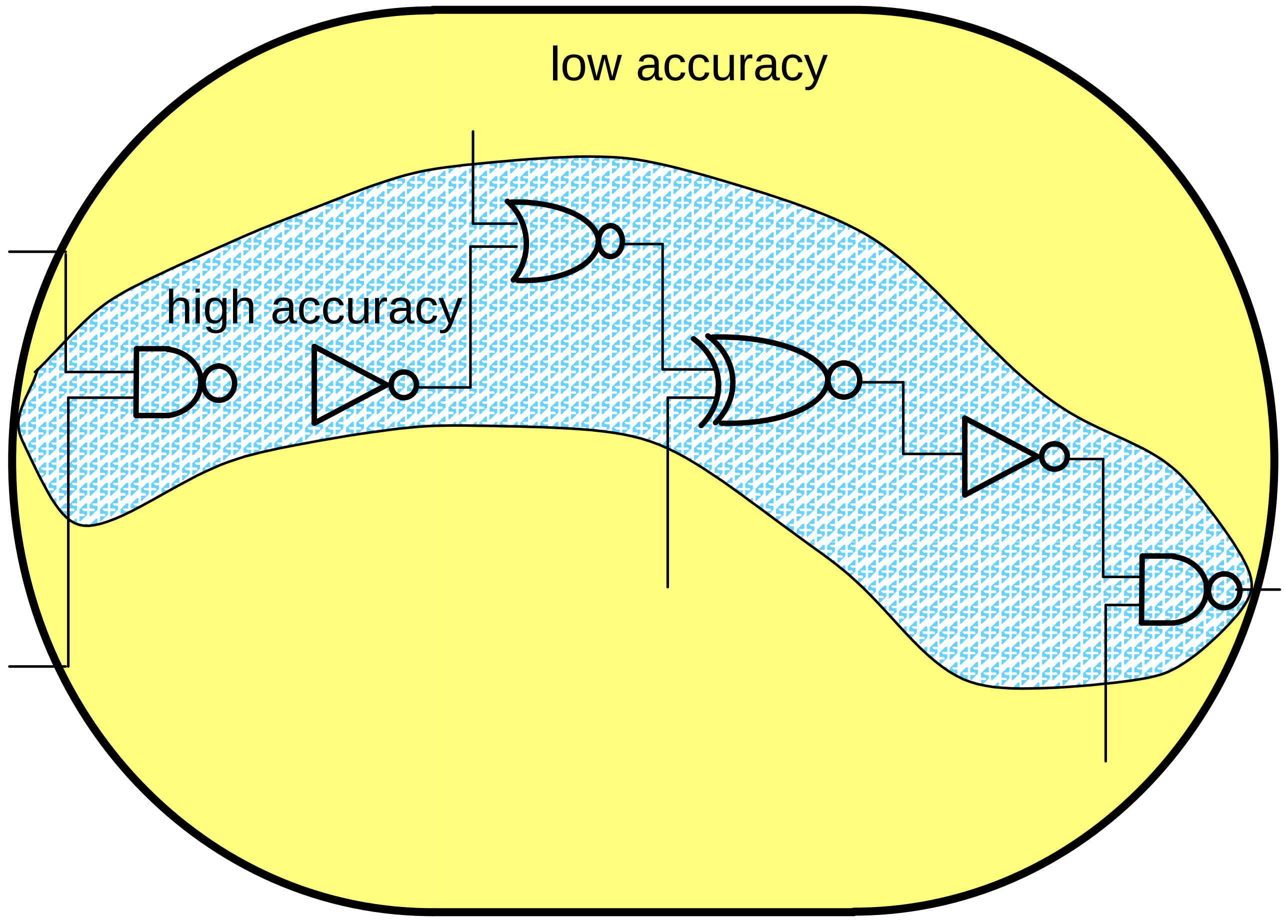
Motivation

- activity factor of a digital circuit is usually low; try to exploit this multi–rate or latency behavior



Motivation

- avoid the accuracy overkill
- variable accuracy analysis is useful for critical path analysis



Part 2: Circuit Analysis

Classical circuit analysis

Consider a circuit with nonlinear capacitors (C's), nonlinear resistive elements and current sources (L's).

$$\boldsymbol{C}_C \dot{\boldsymbol{v}}_C + \boldsymbol{A}_L \boldsymbol{i}_L = \boldsymbol{0}$$

where $\boldsymbol{C}_C$ = capacitance matrix
$\boldsymbol{i}_C$, $\boldsymbol{v}_C$ = currents, voltages of the C's
$\boldsymbol{i}_L$, $\boldsymbol{v}_L$ = currents, voltages of the L's
$\boldsymbol{A}_L$ = incidence matrix of the L's

Now model $\boldsymbol{i}_L = \boldsymbol{Y}\boldsymbol{v}_L + \boldsymbol{J}$ (admittance formulation).

$$\boldsymbol{C}_C \dot{\boldsymbol{v}}_C + \boldsymbol{A}_L(\boldsymbol{Y}\boldsymbol{v}_L + \boldsymbol{J}) = \boldsymbol{0}$$

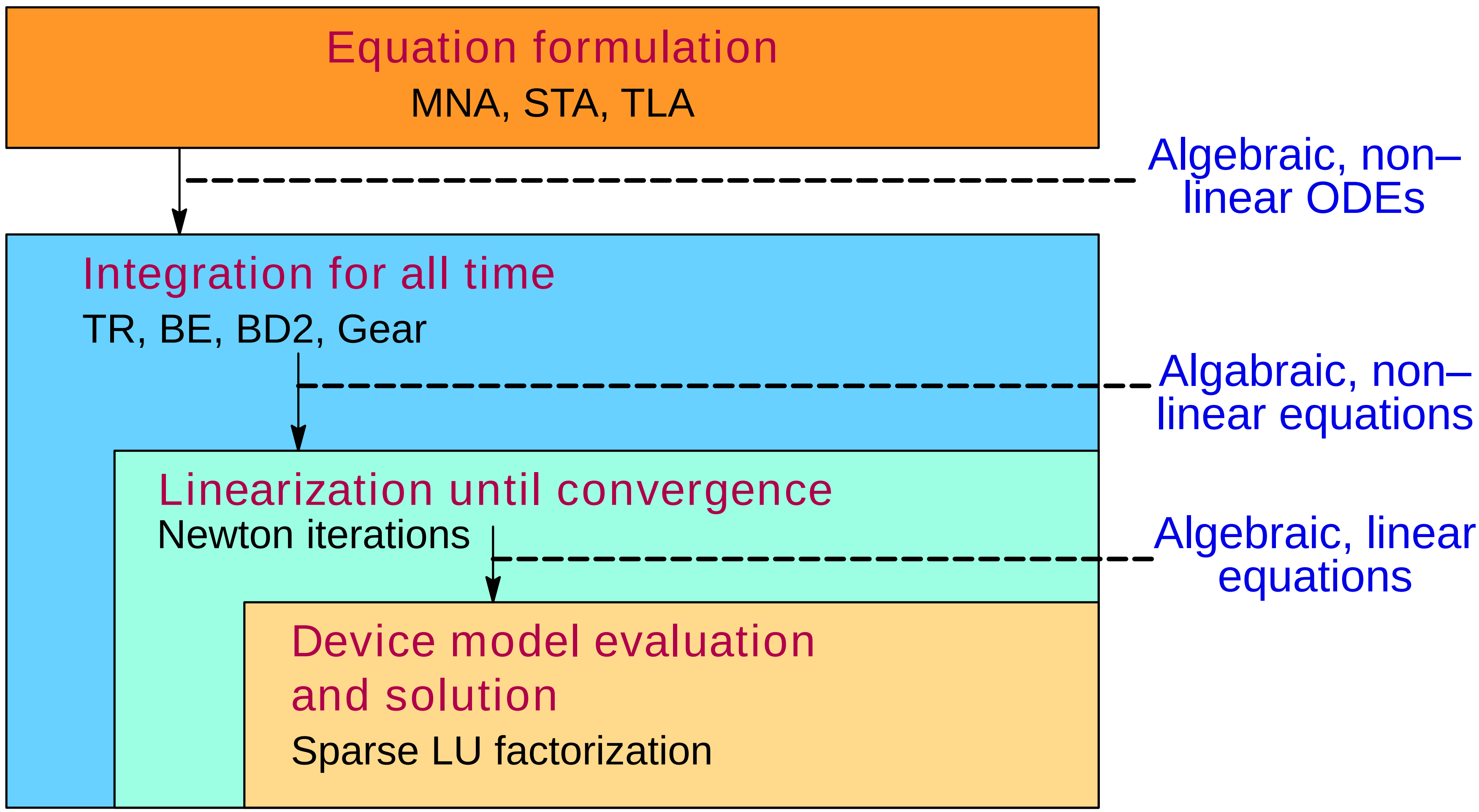
$$\dot{\boldsymbol{v}}_C = -\boldsymbol{C}_C^{-1}\boldsymbol{A}_L\boldsymbol{Y}\boldsymbol{v}_L - \boldsymbol{C}_C^{-1}\boldsymbol{A}_L\boldsymbol{J}$$

But since $\boldsymbol{v}_L = \boldsymbol{A}_L^T \boldsymbol{v}_C$,

$$\boxed{\dot{\boldsymbol{v}}_C = -\boldsymbol{C}_C^{-1}\boldsymbol{A}_L\boldsymbol{Y}\boldsymbol{A}_L^T\boldsymbol{v}_C - \boldsymbol{C}_C^{-1}\boldsymbol{A}_L\boldsymbol{J}}$$

- of the general form $\dot{\boldsymbol{x}} = \hat{\boldsymbol{A}}\boldsymbol{x} + \hat{\boldsymbol{B}}\boldsymbol{u}$
- coupled, nonlinear DAEs

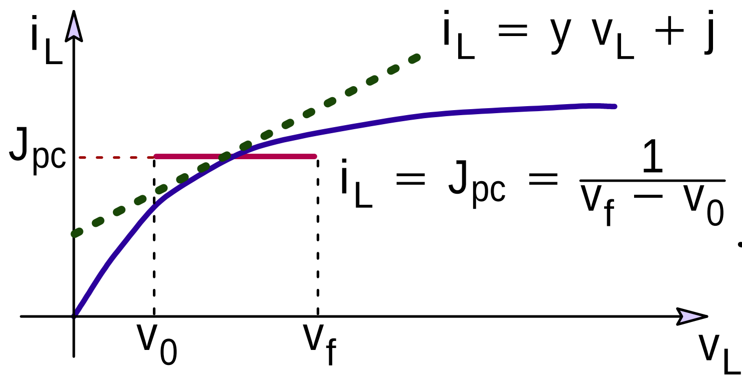
Classical Circuit Analysis



- 1% relative timing accuracy
- $O(n^{1.4})$ growth of run time with circuit size
- size of circuit analyzed is limited to about 10,000 transistors

SPECS: Key Idea

$$\dot{v}_C = -C_C^{-1} A_L Y A_L^T v_C - C_C^{-1} A_L J$$



$$i_L = y\, v_L + j$$

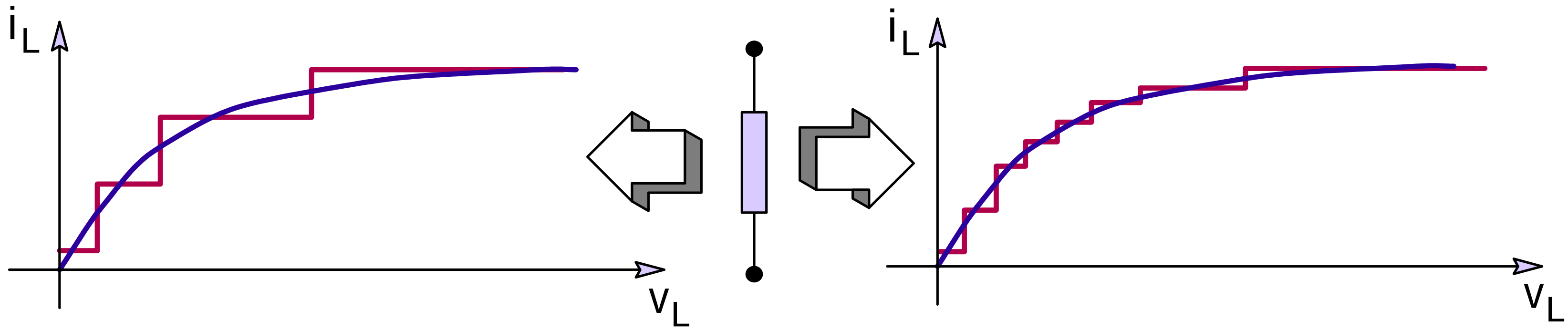
$$i_L = J_{pc} = \frac{1}{v_f - v_0} \int_{v_0}^{v_f} f(v)\, dv$$

$$\dot{v}_C = -C_C^{-1} A_L J_{pc}$$

Implications of this approximation

- nonlinearity of equations eliminated
- if C_C is independent of v_C implicit nature of equations eliminated (i.e., assume that all capacitances are constant)
- if C_C is diagonal, equations are scalar (i.e., all capacitors are grounded)
- all currents are piecewise constant (pwc) in time, voltage slopes are pwc in time, voltages are piecewise linear (pwl) in time
- integration becomes trivial; event–driven analysis possible

SPECS: Modeling and Analysis

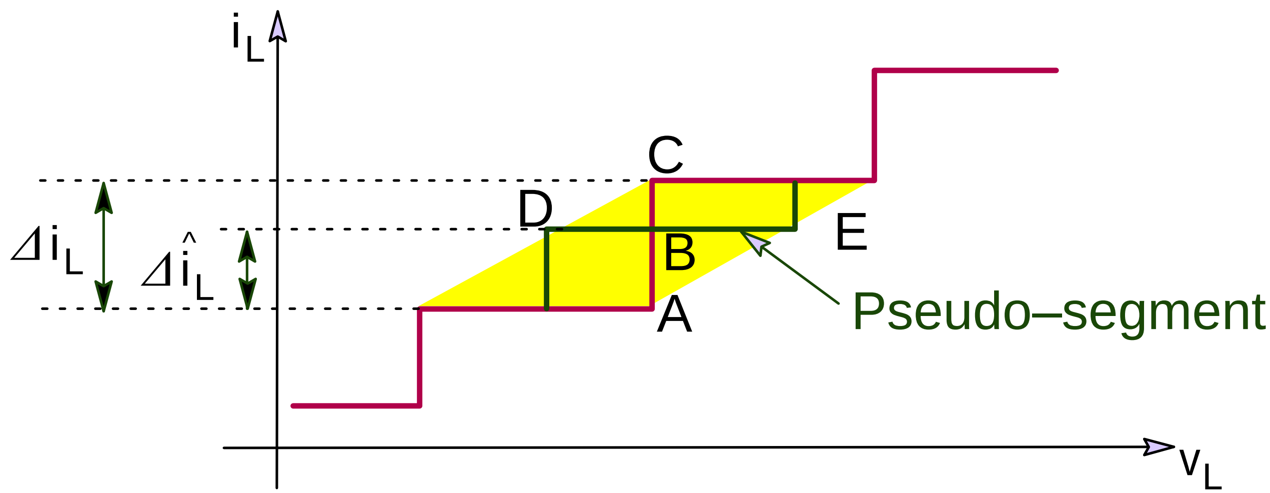


$$\dot{v}_C = -C_C^{-1}A_L J_{pc} \text{ and } \dot{v}_L = -A^T C_C^{-1} A_L J_{pc}$$

1. Every time an element of J_{pc} "jumps," an event occurs; only one element of i_L changes; update $\dot{v}_C$ and $\dot{v}_L$.
2. $t_{target} = \dfrac{v_{target} - v_{present}}{\dot{v}_L}$; update all affected event times.
3. Update the event queue and continue by picking most imminent event on the queue.

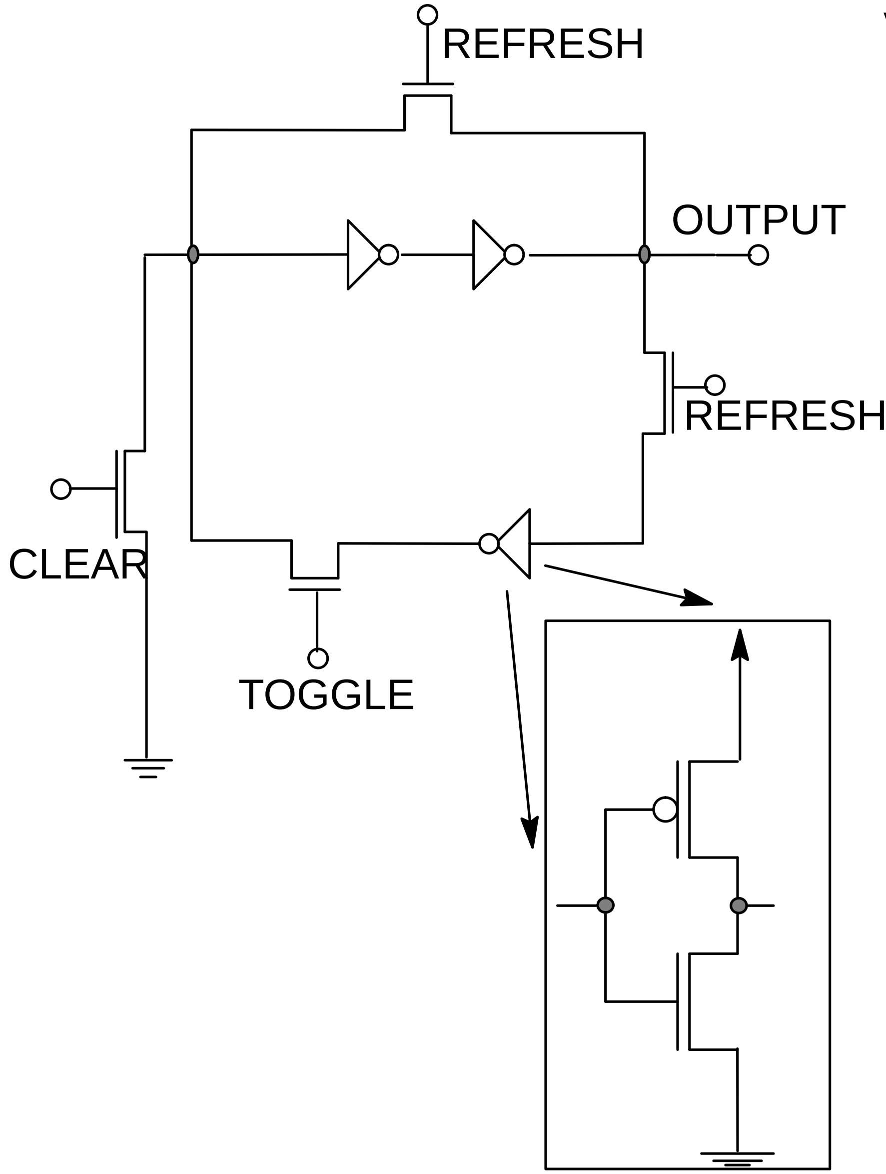
- Variable accuracy analysis achieved
- Relative timing error due to approximation is proportional to $\left[\dfrac{\Delta i}{\dot{v}_L}\right]^2$ which is very advantageous for digital circuits.

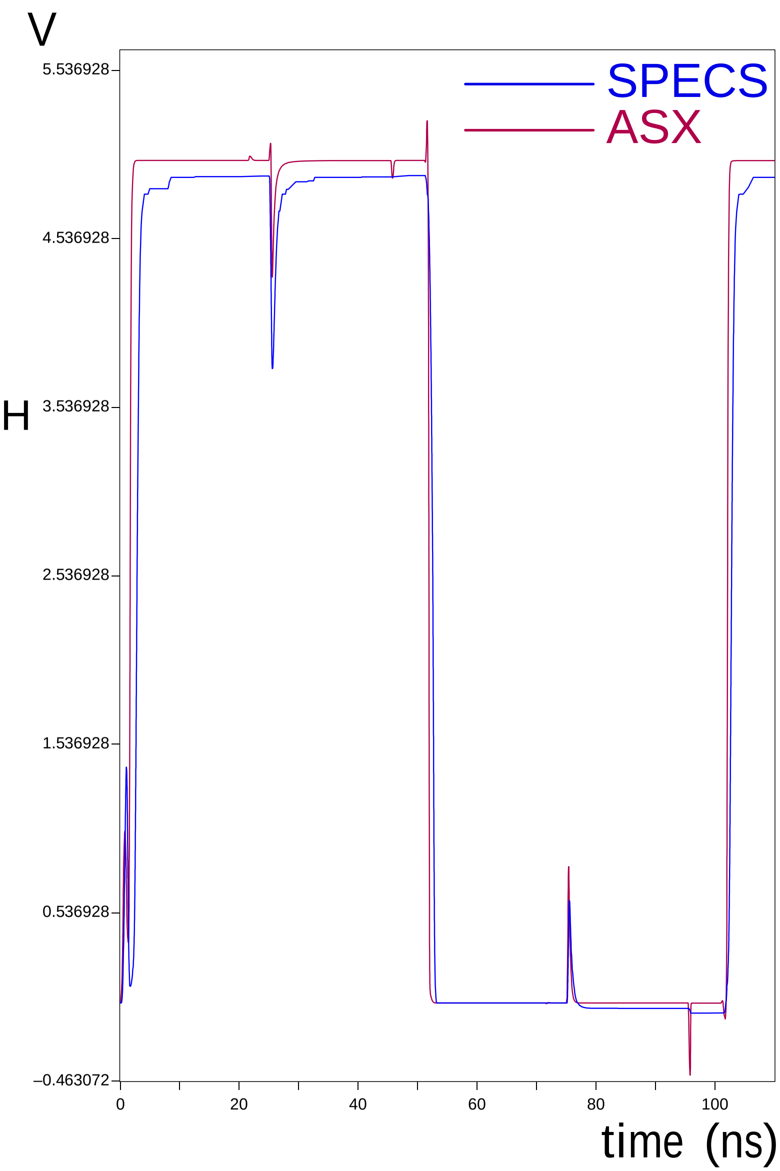
SPECS: The Steady–State Problem



- if $\dot{v}_L^A > 0$, we have an “increasing event”
- then if $\dot{v}_L^C = \dot{v}_L^A - \frac{\Delta i_L}{C_{eq}} < 0$
- find $\Delta \hat{i}_L = \dot{v}_L^A \, C_{eq}$ so that $\dot{v}_L^B = 0$
- interpolate later if necessary to create the pseudo–segment DBE; schedule the element to traverse the segment and reach either D or E
- extend this scheme to multiple dimensions

SPECS: Sample Waveforms





SPECS Summary

Upside

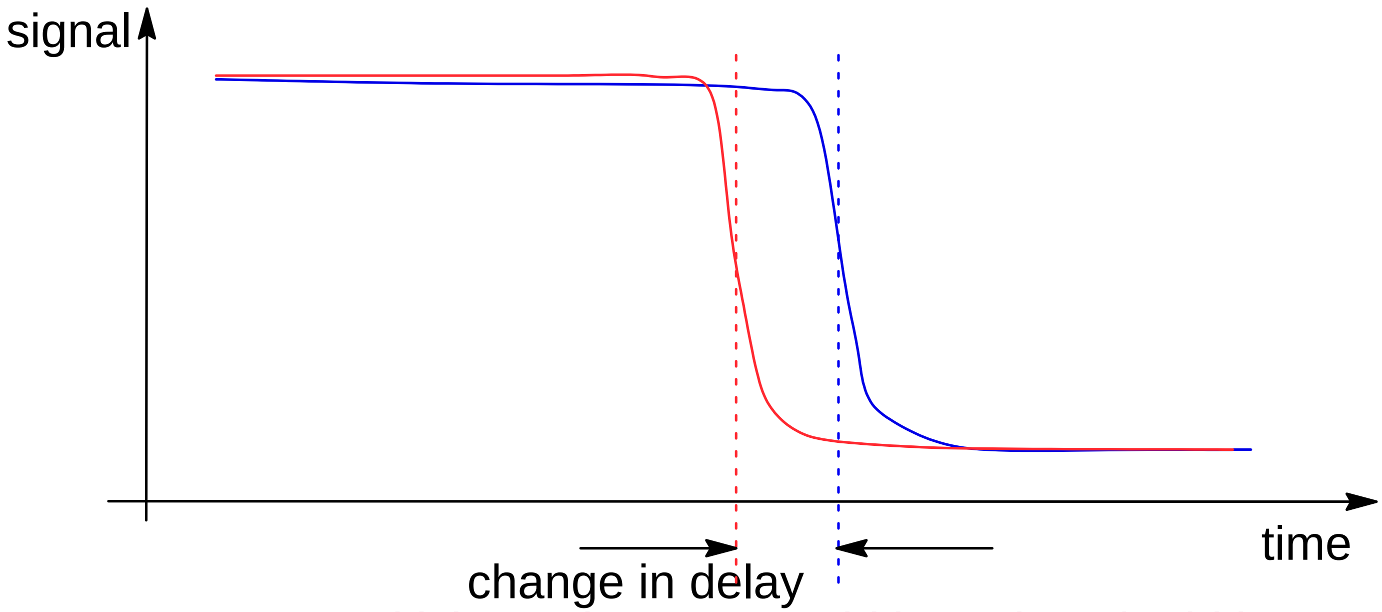
- speed: 70x (6,000 events per second on a workstation)
- capacity: 10x (0.25M transistors)
- accuracy: greater than 1σ of delays with 5% relative timing accuracy; worst case is 10%
- used in 12 IBM sites for
  - timing verification
  - functional verification
  - tuning
  - power estimation

  of digital and memory circuits

Downside

- constant–grounded–capacitance–assumption causes inaccuracy
- stiff circuits a problem
- no inductors
- steady–state level errors incurred

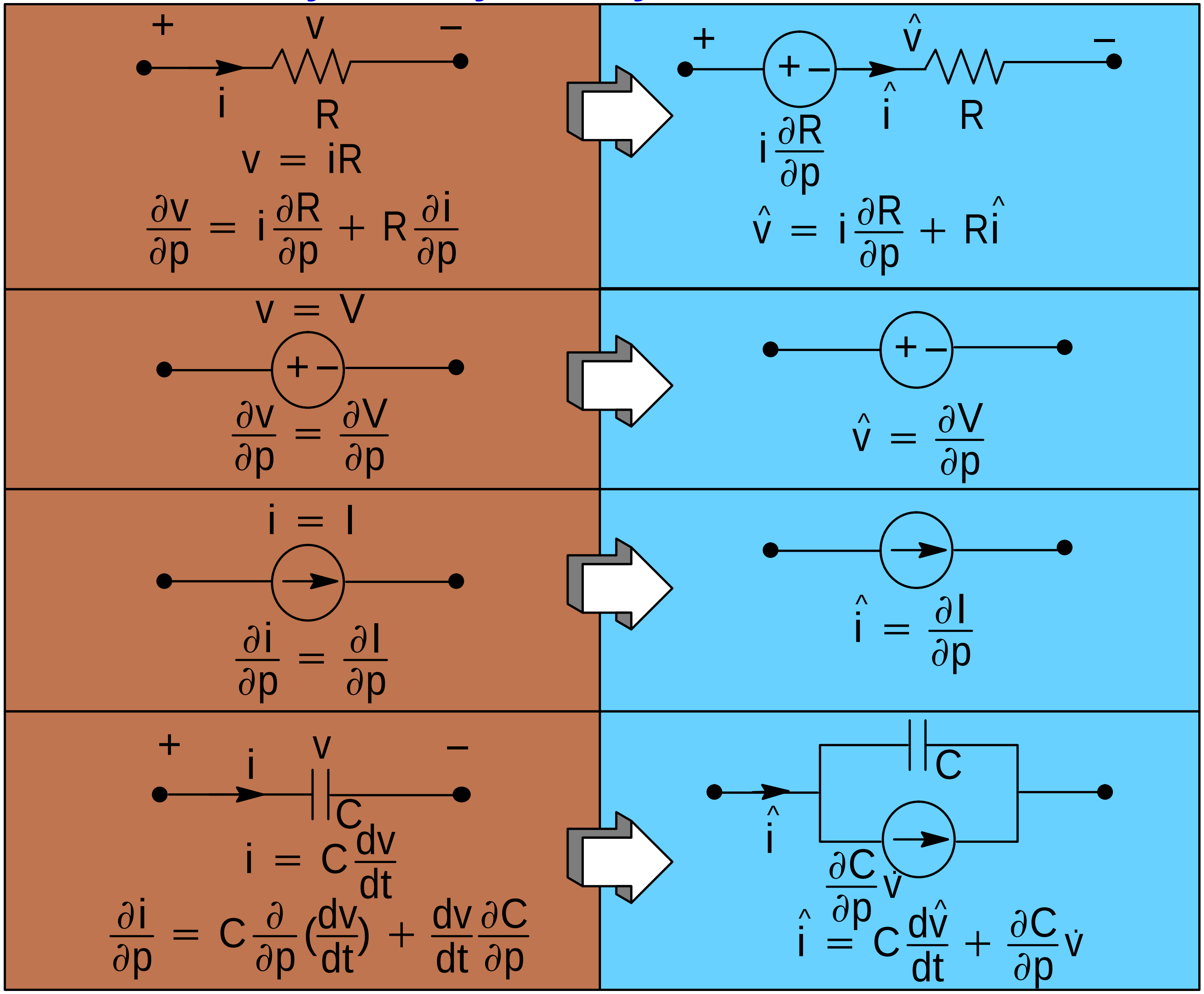
Part 3: Time–Domain Gradients



Sensitivity of delay w.r.t. width = ∂delay/∂width

- time–domain sensitivity computation is a unique feature of SPECS
- sensitivity computation is incremental; not by finite difference or perturbation methods
- both the adjoint and direct methods have been implemented
- sensitivity computation is a small overhead on nominal transient simulation run

Sensitivity Analysis by the Direct Method

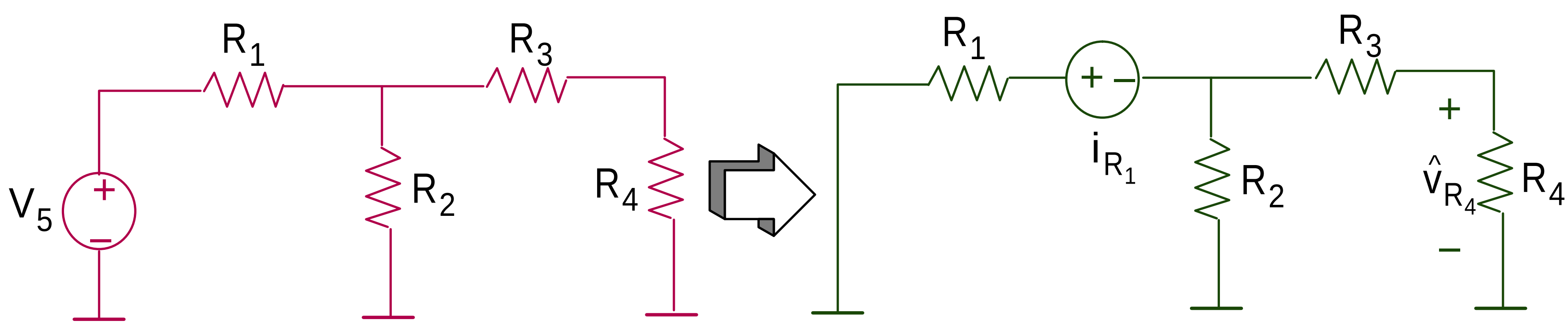


Key: directly differentiate BCRs

Direct Method: Procedure

- replace elements by directly differentiating BCRs to get sensitivity circuit
- reuse LU factors of original solution
- any number of functions, only one parameter

Example:



- problem is to find $\frac{\partial v_{R_4}}{\partial R_1}$, so $p = R_1$
- replace V_5 by a short; add a voltage source in series with R_1; solve for $\hat{v}_{R_4}$; that is the required answer!

Sensitivity Analysis by the Adjoint Method

Matrix formulation of adjoint method

$$Ax = b$$

$$A\frac{\partial x}{\partial p} + \frac{\partial A}{\partial p}x = \frac{\partial b}{\partial p}$$

$$\text{Given a scalar function } f(x),\ \frac{\partial f}{\partial p} = [\frac{\partial f}{\partial x}]^T[\frac{\partial x}{\partial p}]$$

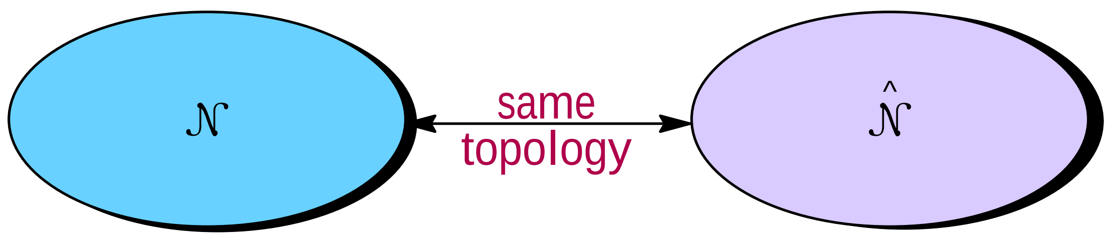
$$\text{Postulate } A^T\zeta = [\frac{\partial f}{\partial x}]$$

$$\text{Then } \zeta^T A = [\frac{\partial f}{\partial x}]^T$$

$$\text{So } \frac{\partial f}{\partial p} = \zeta^T A[\frac{\partial x}{\partial p}] = \zeta^T[\frac{\partial b}{\partial p} - \frac{\partial A}{\partial p}x]$$

- re–use LU factors of original solution; LU factors of A^T are U^T and L^T
- any number of parameters, only one function
- electrical formulation based on Tellegen's theorem

Adjoint Sensitivity via Tellegen's Theorem



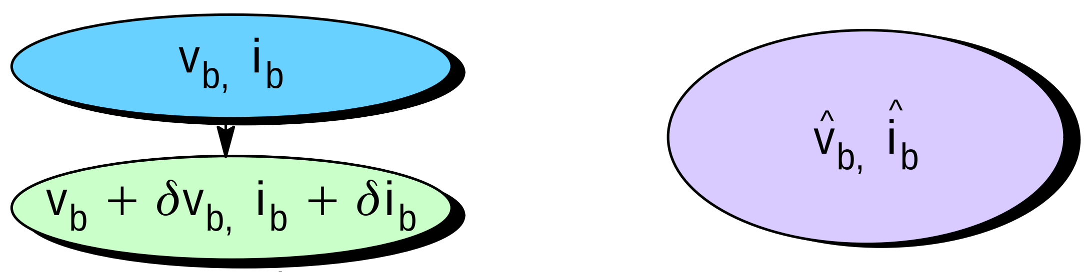
$$A i_b = 0$$

$$v_b = A^T v_n$$

$$\sum_i v_{b_i} \hat{i}_{b_i} = v_b^T \hat{i}_b = [A^T v_n]^T \hat{i}_b = v_n^T A \hat{i}_b = 0$$

$$\sum_i i_{b_i} \hat{v}_{b_i} = i_b^T \hat{v}_b = i_b^T A^T \hat{v}_n = [A\, i_b]^T \hat{v}_n = 0$$

Application of Tellegen's theorem to a perturbed circuit



$$\sum_i v_{b_i} \hat{i}_{b_i} = \sum_i i_{b_i} \hat{v}_{b_i} = 0$$

$$\sum_i (v_{b_i} + \delta v_{b_i}) \hat{i}_{b_i} = \sum_i (i_{b_i} + \delta i_{b_i}) \hat{v}_{b_i} = 0$$

$$\sum_i (\delta v_{b_i} \hat{i}_{b_i} - \delta i_{b_i} \hat{v}_{b_i}) = 0$$

Key relation!

Adjoint Sensitivity Computation

Consider a circuit with only resistors and current sources:

- Current source: $i_J = J; \qquad \delta i_J = \delta J$

$$\text{Typical term} = (\delta v_J \hat{i}_J - \delta J \hat{v}_J)$$

- Resistor: $v_R = i_R R; \qquad \delta v_R = i_R \delta R + R \delta i_R + \cancel{\delta R \delta i_R}$

  By choosing $\hat{v}_R = R \hat{i}_R$, we get $\text{typical term} = i_R \hat{i}_R \delta R$.

- Total circuit: $\sum_{Js} - \delta v_J \hat{i}_J = \sum_{Js} - \delta J \hat{v}_J + \sum_{Rs} i_R \hat{i}_R \delta R$

- Given a scalar performance function $f(v_J\text{'s})$, $\delta f = \sum \frac{\partial f}{\partial v_J} \delta v_J$.

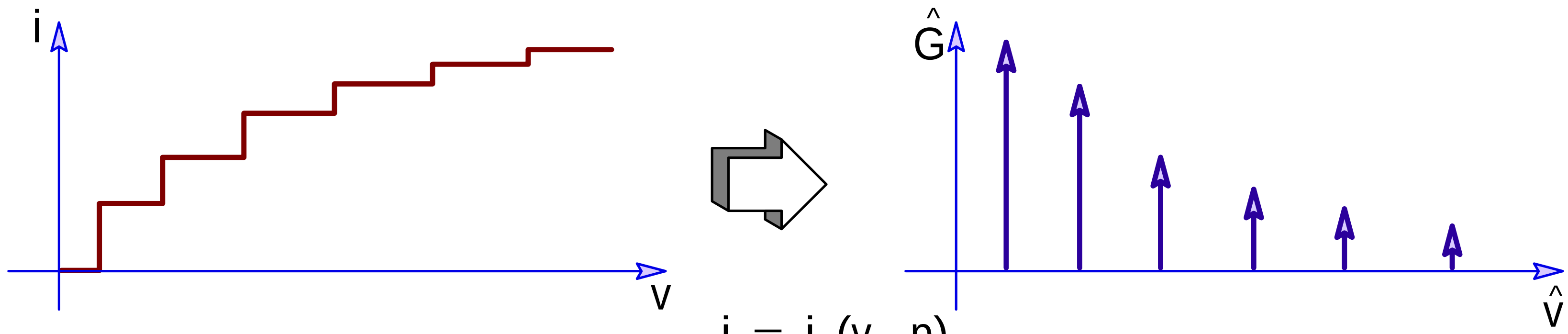
- Choose $\hat{i}_J = -\frac{\partial f}{\partial v_J}$

- Pick off the required sensitivities:

  Thus $\boxed{\frac{\partial f}{\partial J} = -\hat{v}_J}$ and $\boxed{\frac{\partial f}{\partial R} = i_R \hat{i}_R}$.

- Any number of parameters can be accommodated at once!

Sensitivity Analysis in SPECS



$$i = i\,(v,\; p)$$

$$\frac{di}{dp} = \frac{\partial i}{\partial v}\frac{dv}{dp} + \frac{\partial i}{\partial p} \qquad \text{or} \qquad \hat{i} = \hat{G}\hat{v} + \hat{I}_x$$

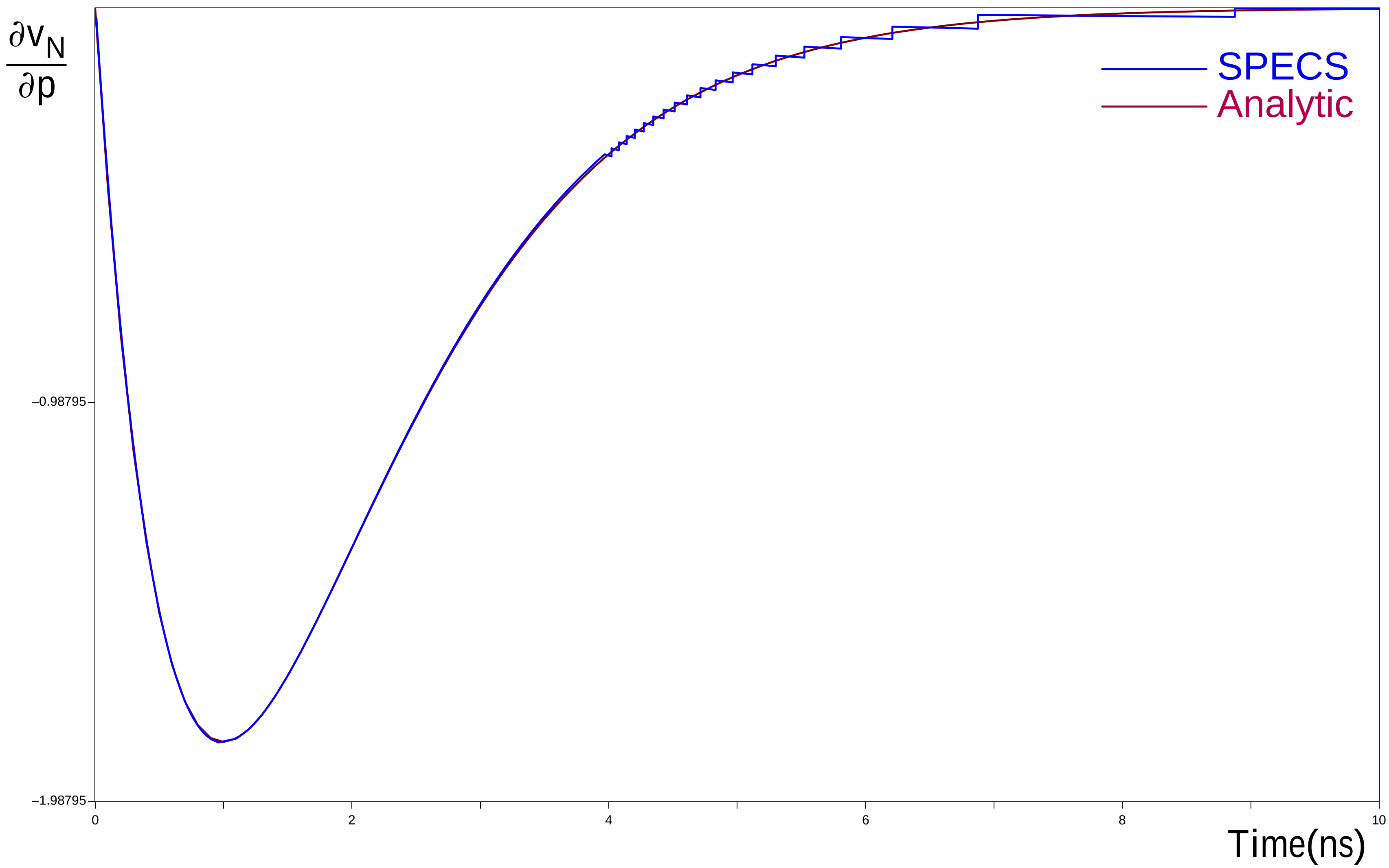
- sensitivity or adjoint circuit consists of mostly disconnected capacitors
- little "spurts" of charge sharing occur at the times of the events of the original nominal circuit
- results of sensitivity or adjoint circuit yield required sensitivities; convolution is required in the case of adjoint sensitivity
- compute delay sensitivities as follows:

$$v_{cross} = v_N \Big|_{t = t_{cross}}$$

$$\frac{dv_{cross}}{dt} = 0 = \frac{d}{dp} v_N(t, p) = \frac{\partial v_N}{\partial t}\Bigg|_{t = t_{cross}} \frac{dt_{cross}}{dp} + \frac{\partial v_N}{\partial p}\Bigg|_{t = t_{cross}}$$

$$\frac{dt_{cross}}{dp} = -\frac{\hat{v}_N}{\dot{v}_N}\Bigg|_{t = t_{cross}}$$

Sensitivity Results on Simple Circuit



Part 4: Circuit Optimization

- designers spend OODLES of time manually tuning their (custom) circuits by adjusting transistor widths
- slow, tedious, error–prone process with circuit analysis in the inner loop

<u>Given</u>

- a circuit schematic with initial transistor sizes
- input signals or stimuli
- precise statement of circuit performance requirements

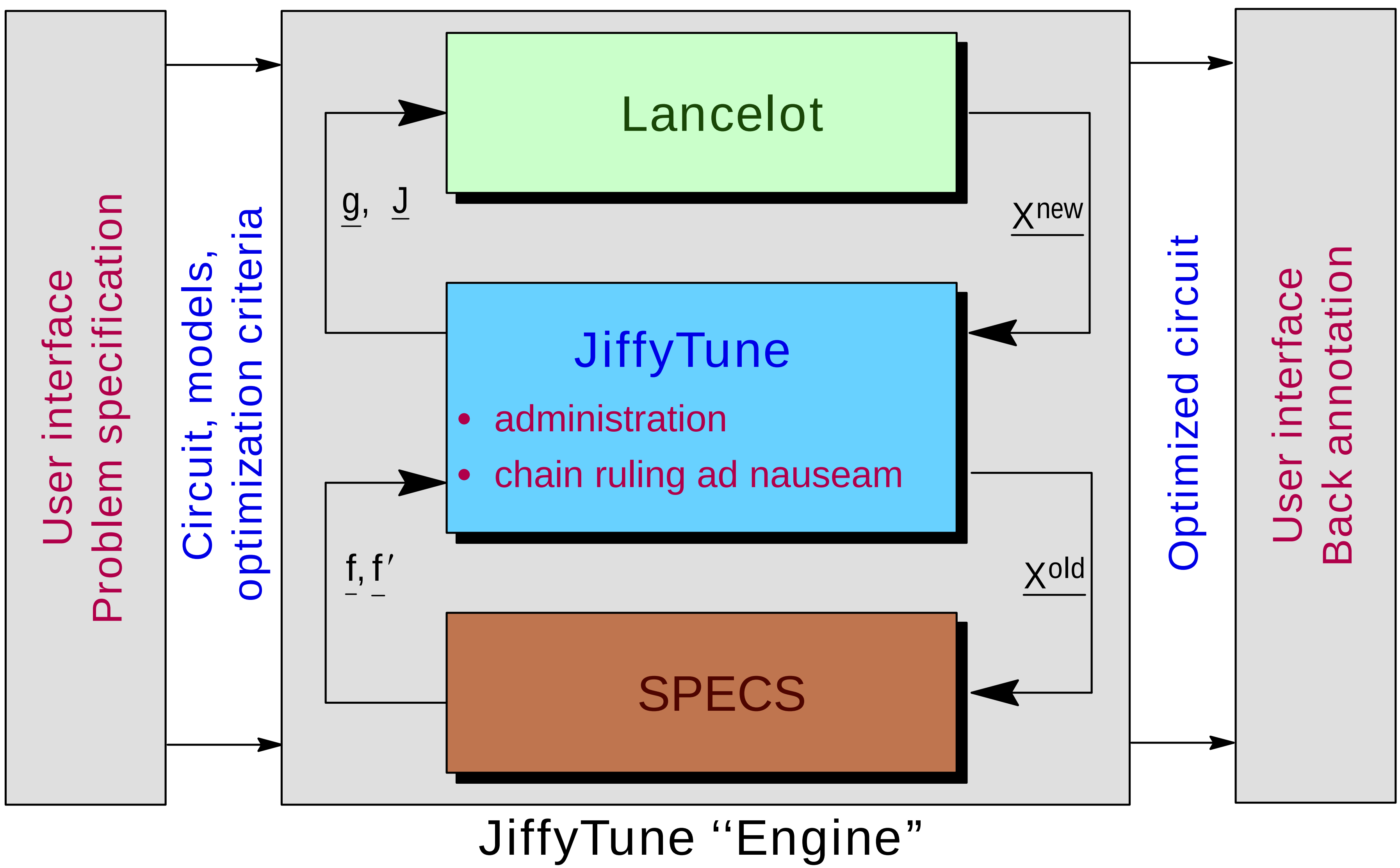
<u>Determine</u>

- optimal transistor size assignments

<u>While supporting...</u>

- flexible objective functions and constraints
- simple bounds
- ratio–ing of transistors and grouping of similar structures
- delay, rise/fall time, area and power measurements
- minimax optimization: $\min_{\underline{x}} \left[\max_{i} \{ f_i(\underline{x}) \} \right]$
- easy–to–use graphical user interface in schematic environment

JiffyTune Overview



What is Lancelot?

- general–purpose nonlinear optimization package
- designed for large–scale problems for which it is remarkably robust
- efficient, especially for nonlinear constraints
- competitive for small and medium sized problems

Problem size and history

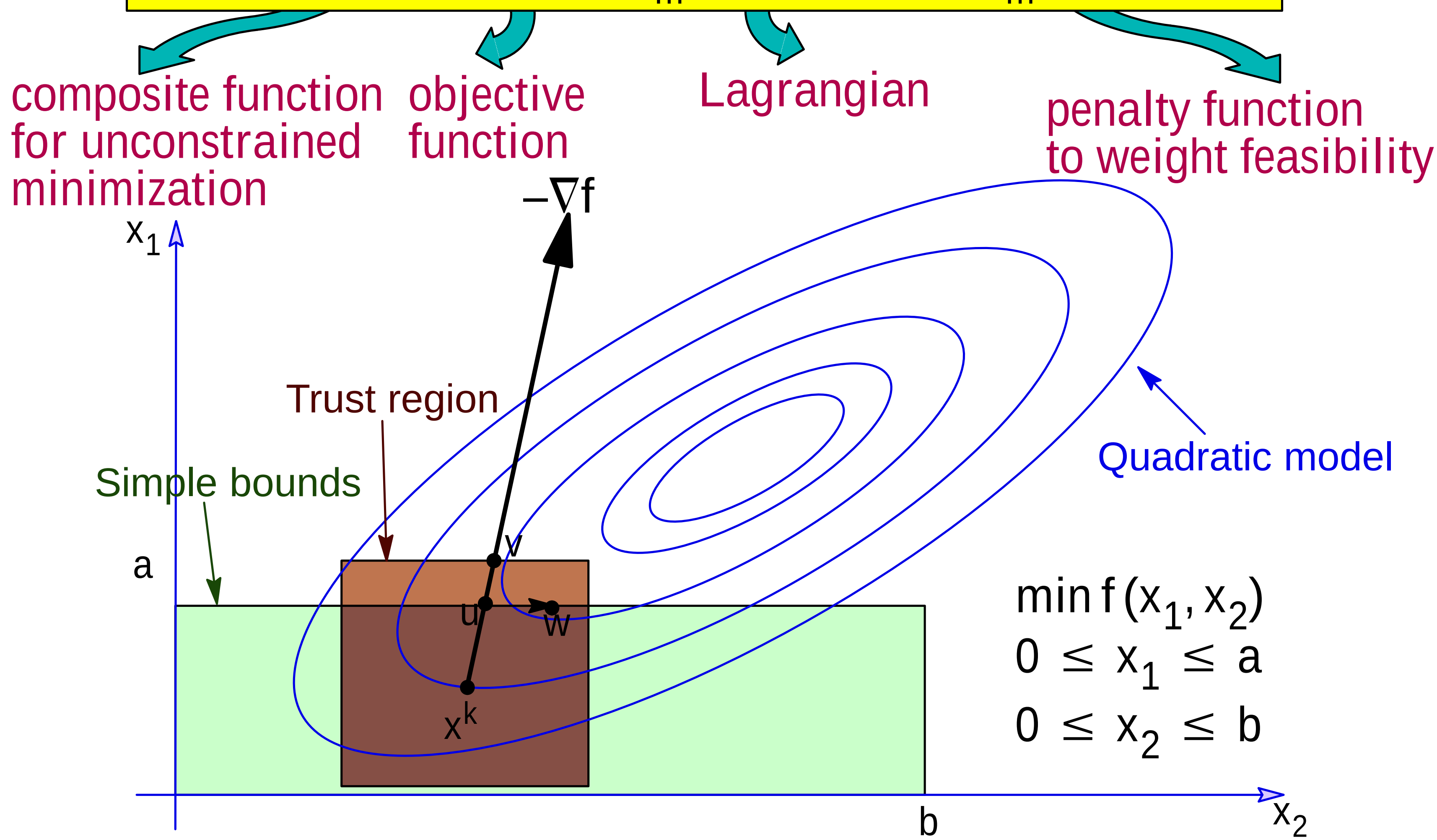
- has successfully handled 20,000 variable problems with 20,000 nonlinear constraints
- has been tested on 15,000 test cases representing 1,000 unique problems

Technical features

- augmented Lagrangian to handle general constraints
- simple bounds handled explicitly and well
- trust region approach
- many options available
  - l_∞ / l_2 trust–region
  - approximate / accurate bounded quadratic problem (BQP) solver
  - various preconditioners, different initializations

How Lancelot Works

$$\Phi(x,\lambda,s,\mu) = f(x) + \sum_{m} \lambda_i \, c_i(x) + \frac{1}{2\mu} \sum_{m} s_{ii} \, c_i(x)^2$$



Minimax Optimization

Original problem

$$\min_{\underline{x}} \left[\max_{i} \{ f_i(\underline{x}) \} \right]$$

Remapped problem

$$\min \ (kz)$$

$$\text{Subject to } kz > f_i(\underline{x}) \quad \text{for all } i$$

Special considerations

- the scaling constant k helps create a well–scaled problem and helps with stopping criteria
- kz is initialized to the largest of the $f_i(\underline{x})$'s after the first function evaluation

Lancelot: Special Issues

Problems due to noise

- unable to recover from poor initializations
- optimizer wastes time going nowhere
- does not stop gracefully or predictably

Special considerations for noisy optimization

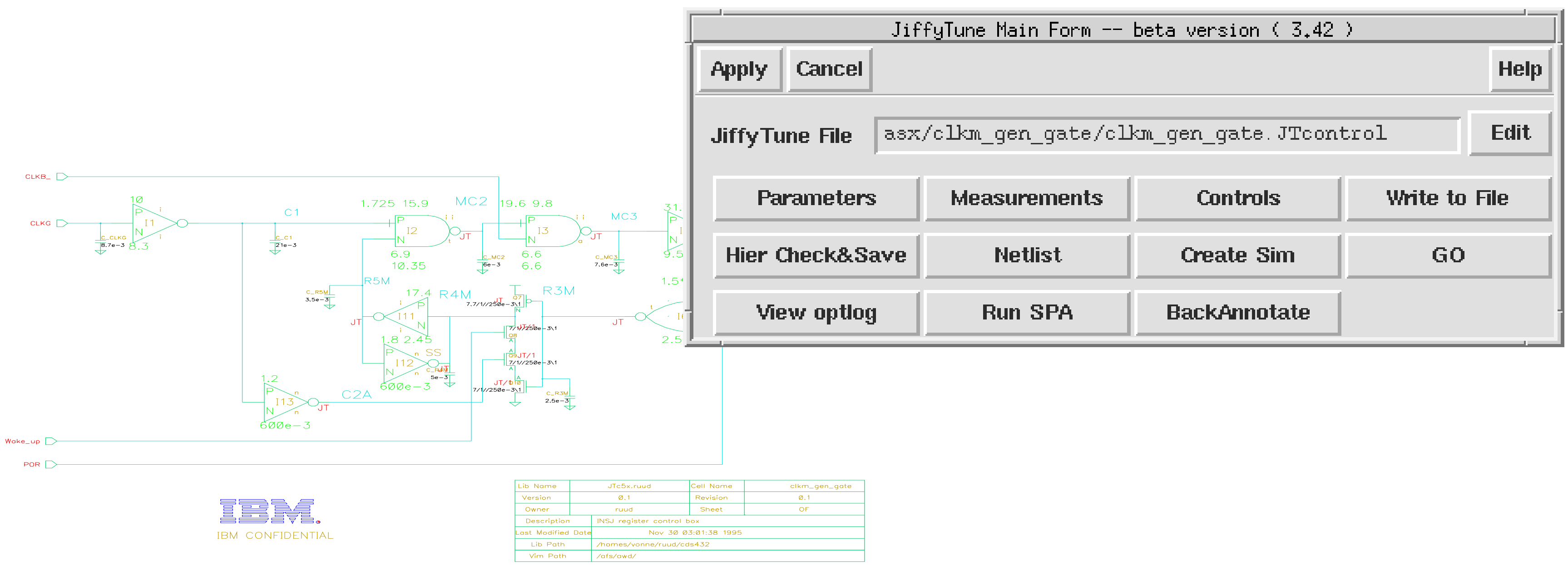
- tolerance for feasibility
- tolerance for line search break points
- consideration of small step beneath which further progress unlikely
- initial trust region radius
- initial Hessian approximation for secant methods
- stopping criteria

Miscellaneous

- slack variables updated at each major iteration
- testing environment has allowed us to integrate MINOS from Stanford as an alternative optimizer

Results

- JiffyTune has been used to tune about 120 circuits of a custom dynamic–logic microprocessor; it is a key productivity tool
- forces careful and explicit specification of circuit requirements: the issue becomes how to correctly specify the optimization problem rather than solving the optimization problem itself!
- produces “sensible” results, like progressively sizing up buffers, and tapering stacks of transistors
- design re–use a bonanza



Conclusions

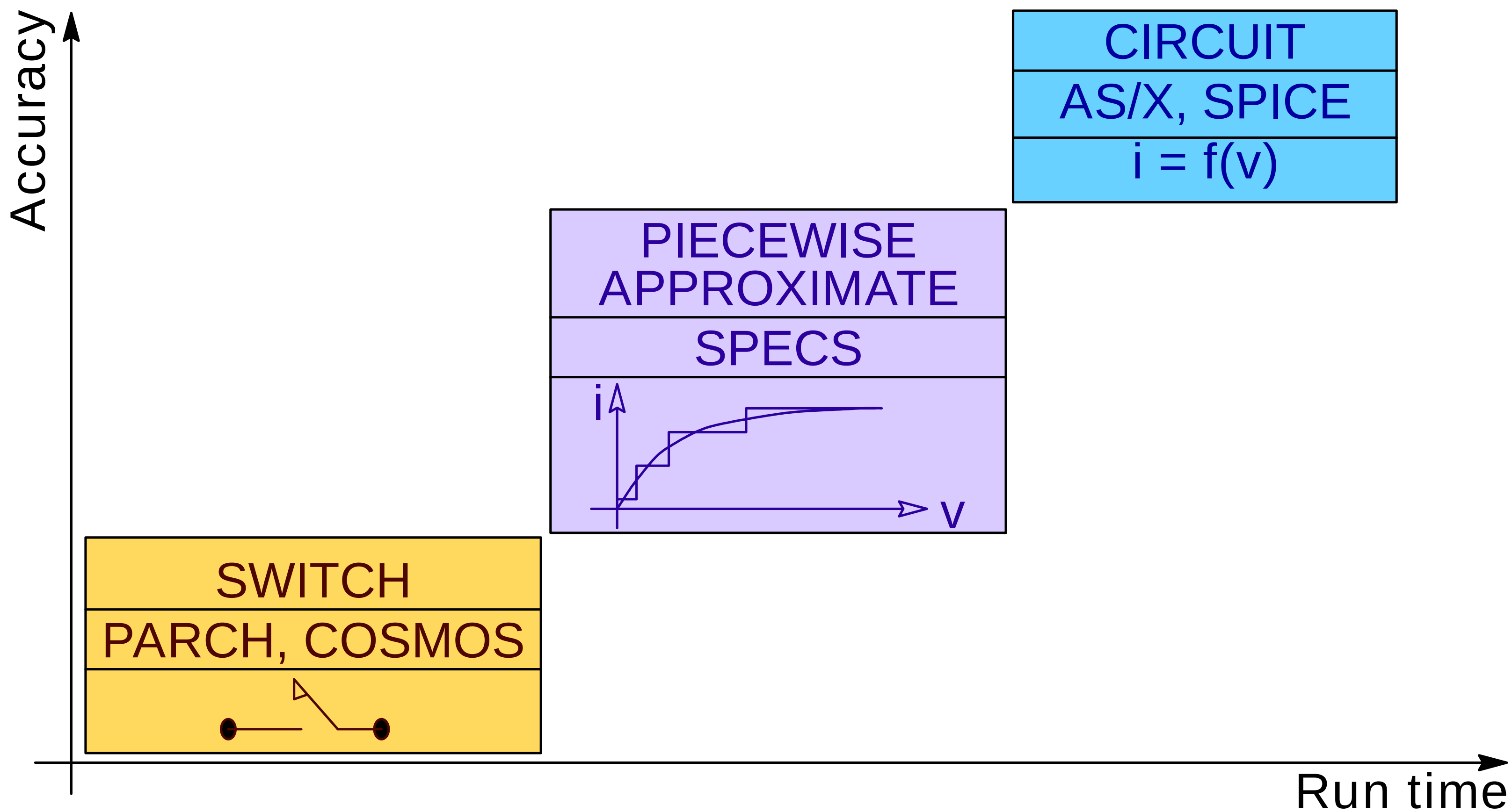
Conclusions

1. By simplifying device models, we are able to perform fast, event–driven circuit analysis.
2. The simplified analysis lends itself to efficient time–domain gradient computation by both the direct and adjoint methods.
3. The simulation and gradient computation has been placed in the inner loop of a nonlinear optimization package to tune custom circuits.

Future work

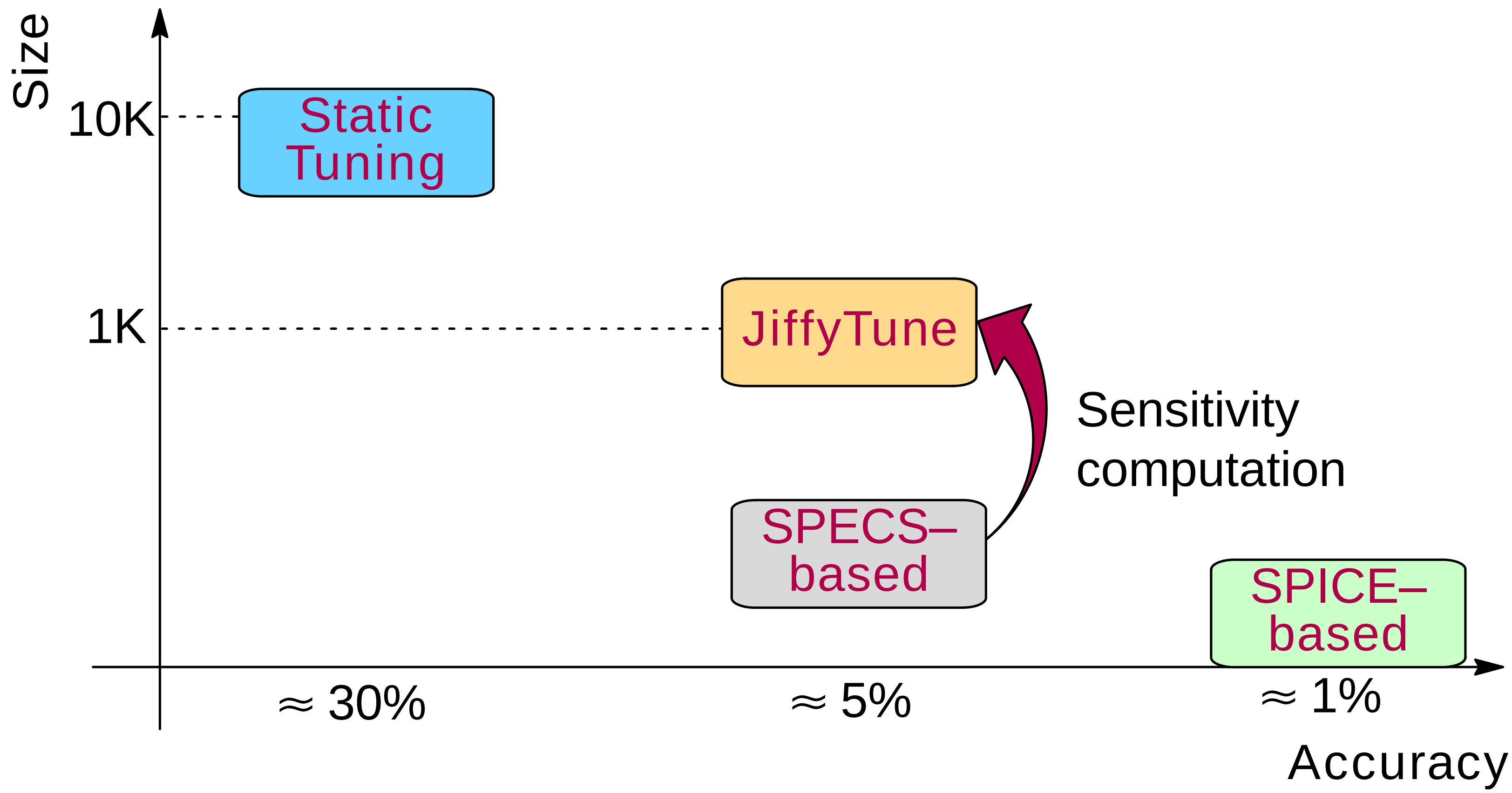
1. Improve adjoint sensitivity performance by event–driven convolution; compute gradients in parallel.
2. Accommodate semi–infinite constraints; perhaps extend JiffyTune to solve manufacturability problems.
3. Automatic recovery from non–working circuits.
4. Investigate efficient computation of the Hessian.

SPECS



Accuracy
Run time
SWITCH
PARCH, COSMOS
PIECEWISE APPROXIMATE
SPECS
i
v
CIRCUIT
AS/X, SPICE
i = f(v)

Static vs. Dynamic Tuning



- need for pre–characterization
- applicability to custom circuits
- false–path problem vs. need for input vectors

Static and dynamic tuning complement each other!