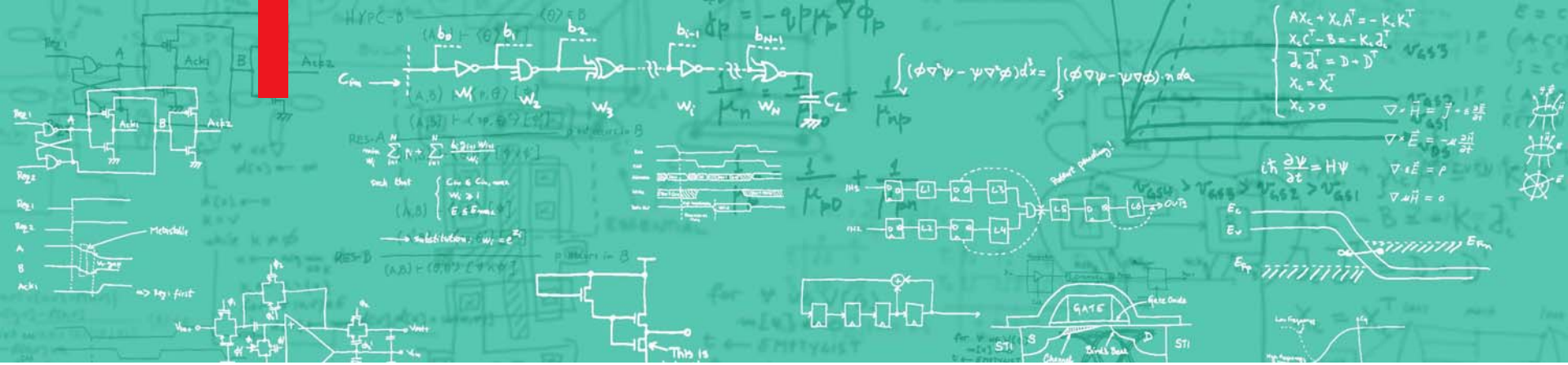


Model Order Reduction for Circuit Applications

Joel Phillips
Cadence Design Systems
3 December 2010
Collaborators: L. Miguel Silveira, Zuochang Ye, Zhenhai Zu, Saurabh Tiwary

cadence™
RESEARCH LABORATORIES

cadence™

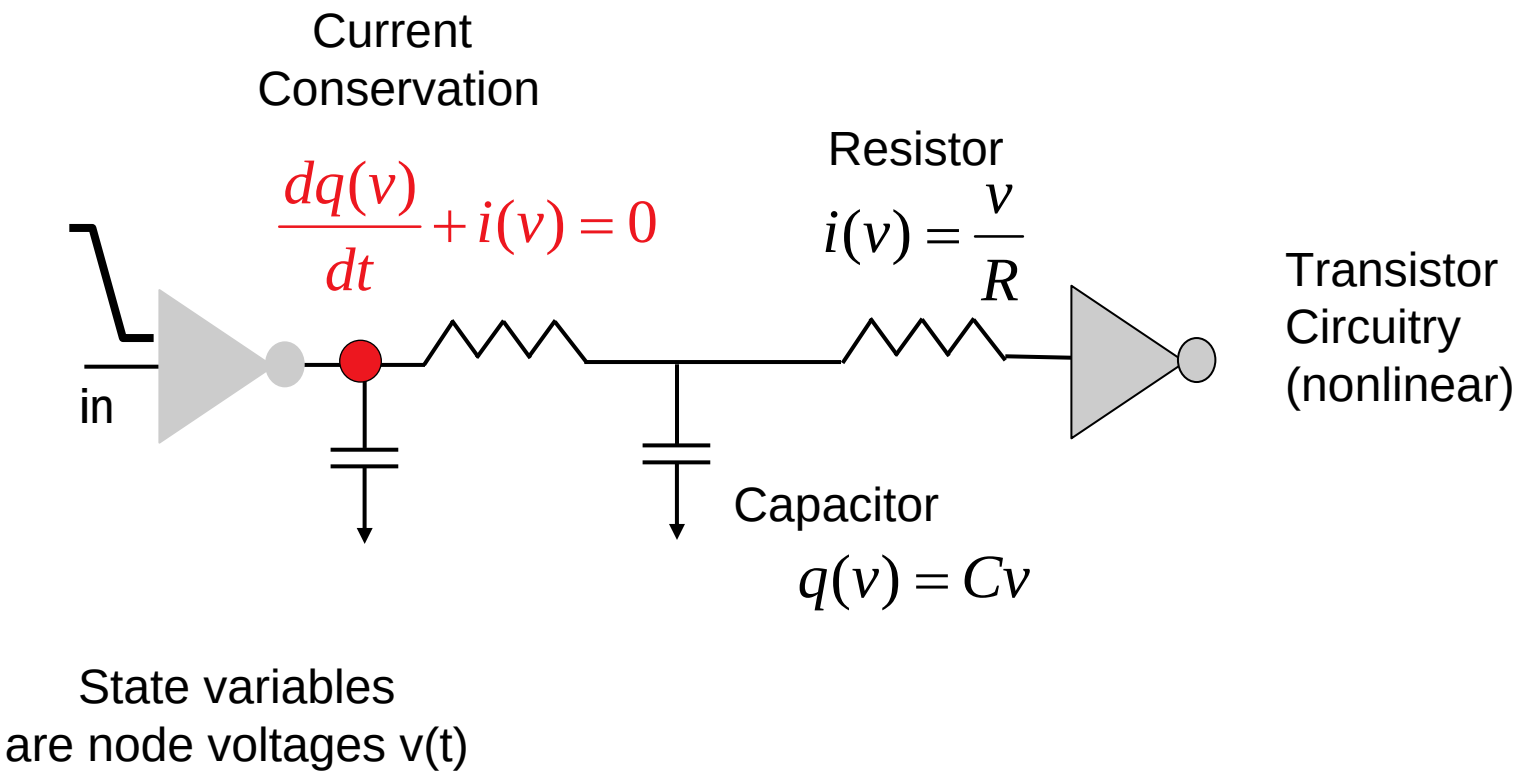
Overview

- Who am I?
  - Application practitioner:
    - Boundary of electrical engineering and applied mathematics
  - Domain: computer-aided design of silicon integrated circuits
  - Company (Cadence): mid-sized publicly-traded software company, “old” by Silicon Valley standards, “large” within EDA industry
- This talk:
  - Heavy on applications and open problems
  - Not too much in mathematics & algorithms – most has been published
- Themes for MOR
  - Constraints of problem scale
  - Context-aware error analysis
  - Sparse, structured networks
  - Incrementality

Application Background

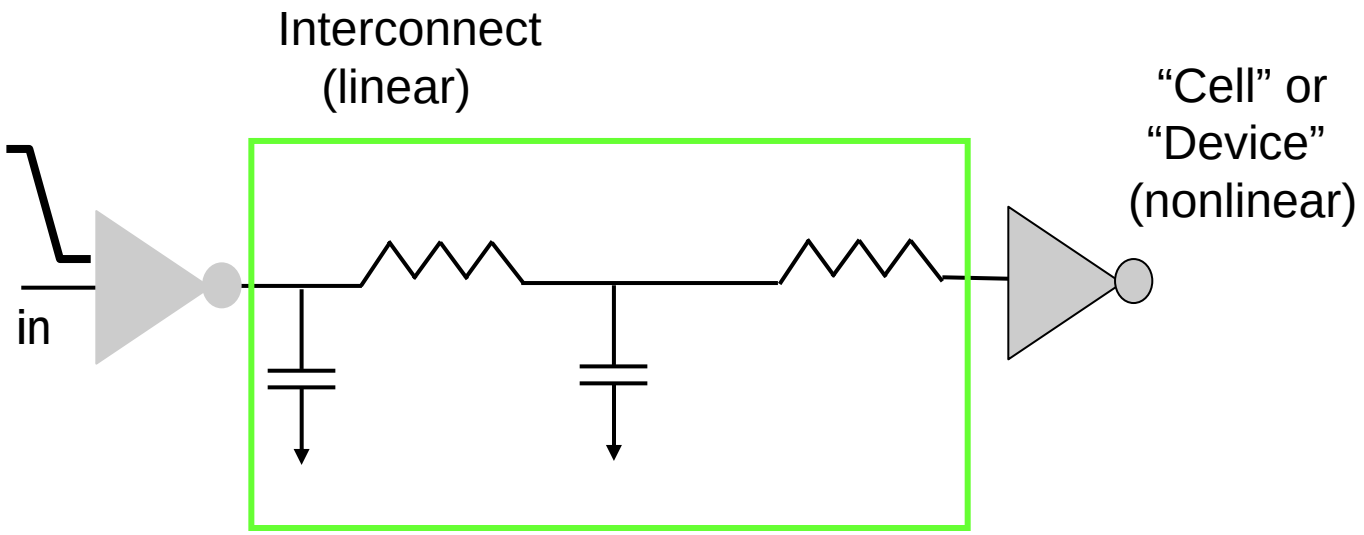
cadence™

Circuit Analysis



- Circuit analysis → nonlinear system of DAEs

Circuit Analysis



- The linear interconnect is usually large in size
  - Wiring to connect designed circuit blocks
  - Parasitic resistance and capacitance that degrades performance
- Reduction is used to compress interconnect which is then simulated together with nonlinear elements to assess circuit performance

Design & Analysis Methodologies

- Analog/Mixed-Signal/RF
  - Part of a cellular phone that communicates with the tower
- Digital ASIC (application specific integrated circuit)
  - The “thinking” (data processing) part of your cell phone
- Custom Digital
  - Intel processor in a desktop computer

- Circuit Simulation. Large systems of (nonlinear) DAEs, one per block (“SPICE”)
- Static Timing. Large numbers of relatively small (nonlinear) systems (wire-by-wire analysis)
- Somewhere in between

Algorithms

- Virtually all successful reduction algorithms in EDA are moment-matching variants or close cousins thereof
  - PVL, PRIMA, …
  - Graph-based techniques (“TICER”)

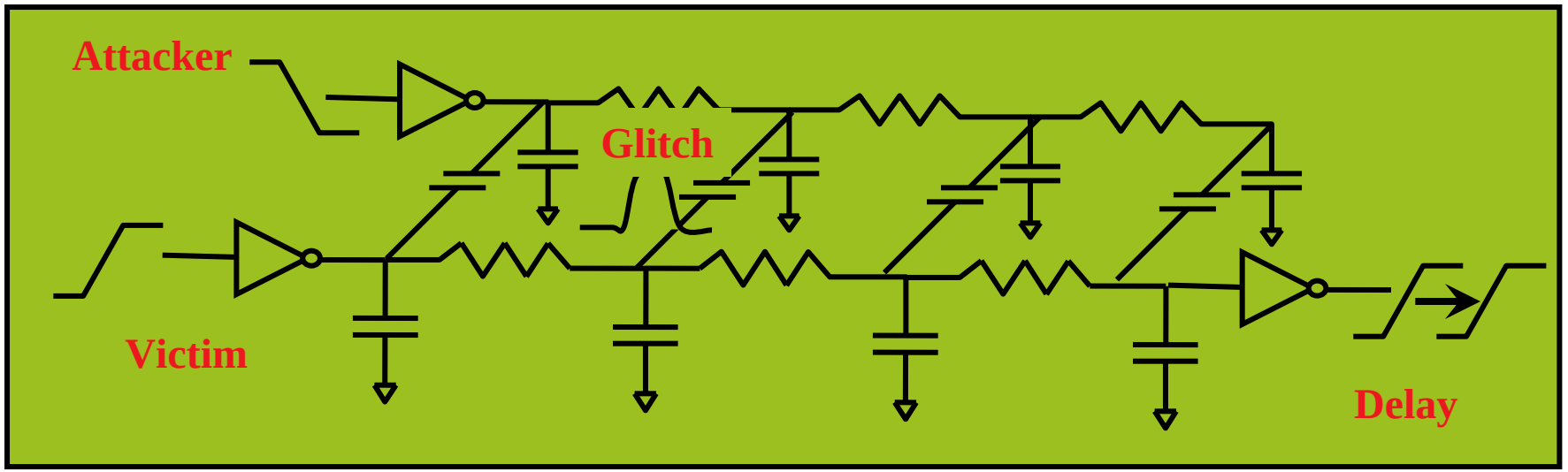
- Question for me: Why this choice?

- Question for you: Technology has not changed dramatically in some time, despite known shortcomings. Why?

ASIC Interconnect

cadence™

Success Story: ASIC Signal-Path Interconnect



- Scale
  - Millions of coupled nets to analyze, each net 1k-100k parasitic elements → Cannot analyze without reduction
  - Electrical verification of every chip made past 10+ years depended critically on MOR.
- Accuracy/Robustness
  - Must provide < 1% error
  - Must provide physical models → passivity concerns

Typical Profile of Interconnect Reduction Problems (Moderate size block, one corner, ~ hour for analysis)

Incidence	Size	Desciption
$O(10^7)$	$O(10^2)$	Local signal nets
$O(10^5)$	$O(10^4)$	Top-level block routing
$O(10^3)$	$O(10^6)$	Global signals, clock
$O(10^0)$	$O(10^9)$	Power/ground networks

Too many occurrences to be able to do anything complicated

Too big to do anything complicated

Scale of *everything* is roughly $O(10^9)$

- Implications
  - Speed: 10K-100K models *per second*.
    Reduction must be fast, final models must be small.
  - Robustness: 0.01% failure rate is not useful

Myths

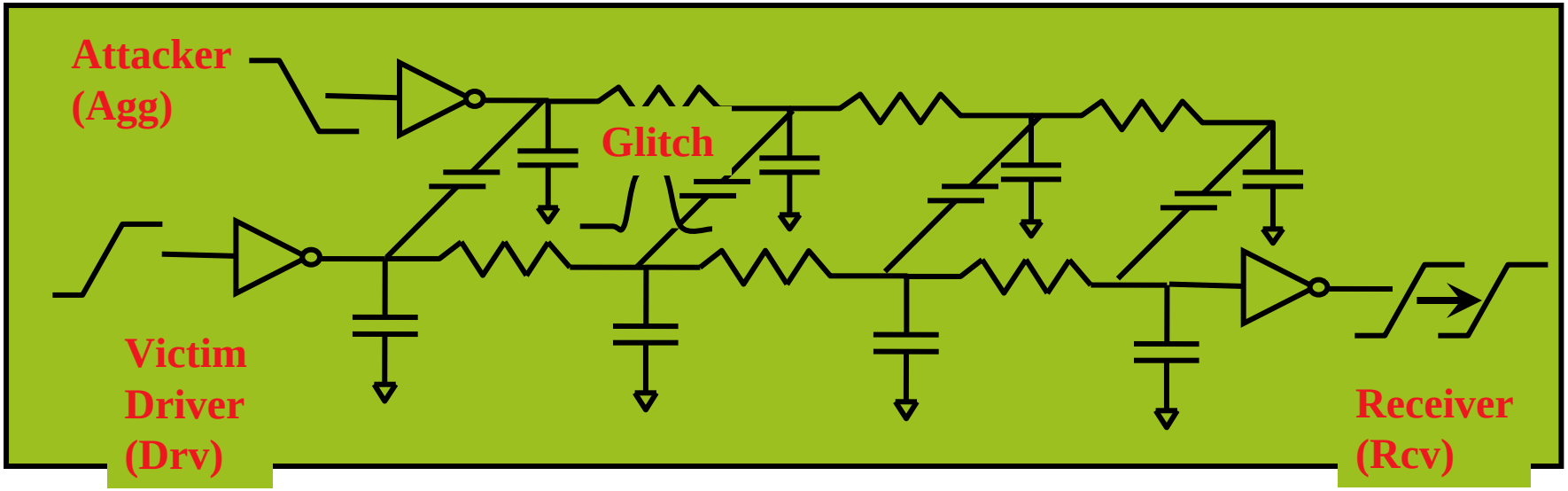
- "Our approach is very fast…<X> executed in only a few seconds"
  - About 1000X too slow
- "The smaller models justify extra effort spent in MOR"
  - Reality: reduction costs matters.
  - Sort of. But not a 1-1 trade. 2X increase in MOR time requires much more than 2X improvement in model size. 5X increase is impossible to justify.
  - Problem is not data rich. Every matrix operation is very expensive. May have ten or fewer XF samples to work with.
- Big problems. "On large problems the sample cost dwarfs the reduction cost, and then our approach is competitive"
  - Most *truly* big problems are not solveable "out-of-the-box" with existing methods.
    - Too many ports, too many intertwined nonlinearities, lack of clearly mappable error metrics, etc. etc.
  - Some of them are so big they will not fit in main memory! This is why they need to be reduced in the first place!
  - *Main issue in these cases is figuring out how to break up the problem into manageable pieces.*

Circuit problems have particular internal structure

- Exploiting network structure is critical – both for top-level algorithms and for innermost kernels.
  - Topologies are sparse and restricted in nature.
  - Black-box system view is too slow.

- Error requirements are inhomogenous and difficult to capture with a single “number”
  - E.g. <X> norm
  - Need to think of more as an optimization problem (minimize runtime) with a set of constraints coming from different error measures

Model Requirements

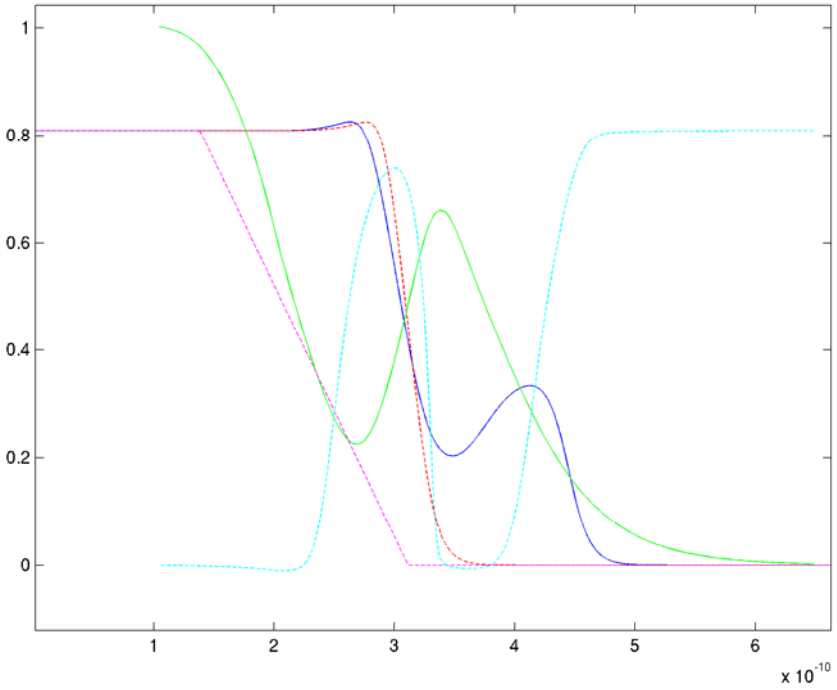


- Different inputs, outputs have different error constraints
- Accuracy requirements are analysis-specific and may vary with time

Transfer Function	Accuracy Requirement
I(Drv)→V(Drv)	High
I(Drv)→V(Rcv)	High
I(Agg)→V(Rcv)	Moderate
I(Agg)→V(Agg)	Low
I(Drv)→V(Agg)	Low
I(Rcv)→V(Rcv)	Low

Circuit problems have restricted excitations & response

- System dynamics are restricted
  - Frequency ranges are fairly well known a-priori
  - Excitations typically defined in time domain. Time-domain waveforms shapes are highly restricted for voltage responses (less so for currents)
- Error estimation needs to be highly tuned
  - But no real good theory → approaches are somewhat ad-hoc.
  - Best we can do is moment counting or (weighted) norms for error control?



cadence™

Implications of Analysis Context

- Runtime is king. Need an algorithm that applies minimal amount of effort to get a minimally sized model that meets error specifications
  - Can't "back off" from a larger model → a-posteriori error analysis is not competitive

- Sometimes the same piece of interconnect will be analyzed at different time in different ways
  - This can be exploited.

Incrementality

cadence™

Incrementality

- What is it?
  - Assume for a given system a model of acceptable (and known) accuracy has been built
  - Some change is made to the analysis context or starting system model
    - It may be known in advance what types of changes are allowed, but the details of the change are not known in advance
      - Distinguished in this way from parametric models, though parametric model technologies can in some cases be adapted to an incremental context
  - What is desired is to “quickly” build a similar model for the “new” context/system
    - Without repeating the work used to form the first model

- Where it occurs
  - Error estimation
  - Derivative models
  - Design variants
    - Higher-level nonlinear analysis
    - MMMC
    - Updates to physical design

Incrementality: Error Estimation

- Ideally, every quantity in an analysis algorithm should be obtained with minimal incremental work

- For example, if we are working upward in model order, work for order Q-1 should be subset of work for order Q

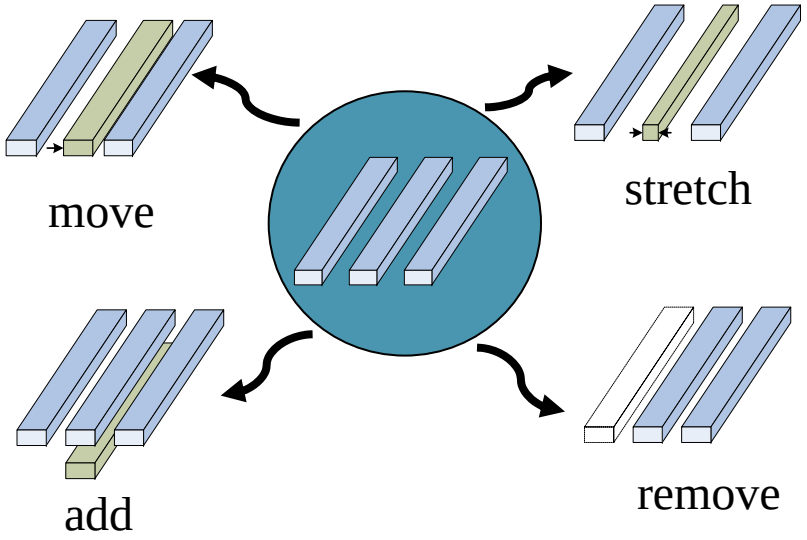
- Yet, most work on error estimation contains expensive-to-evaluate terms

Incrementality: Design Variants

- MMMC
  - “multi-mode multi-corner”
  - Modes == different operating conditions of timing analysis. Doesn’t really affect interconnect physics – does affect excitations.
  - Corners == different process conditions
    - Interconnect corners may affect some networks or some parts of some networks.
    - Process corners may not affect interconnect – or may affect it only on the boundaries.
- Physical Design Changes
  - Move a wire, insert a buffer, resize a driver.
  - Some, possibly small, piece of starting network is changed, the rest remains the same.
  - Change is partially topological, not parametric.
  - Parametric changes are large and not known in advance.
- What I would like:
  - A “model updating” approach whose cost is proportional to degree of change.
  - If a small portion of the network changed, cost to produce the new model should be significantly less than starting from scratch.

A Related Problem: Incremental Electromagnetics

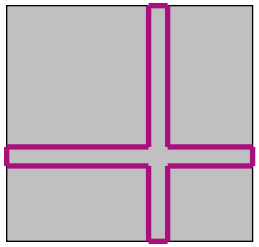
– Large set of system changes: move conductors, resize conductors, add/delete conductors. Changes are large, not just small perturbations.



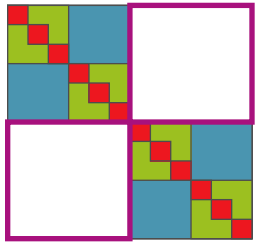
- Matrix updates are often sparse and structured

$$A = A_0 + \Delta$$

$\Delta =$



or

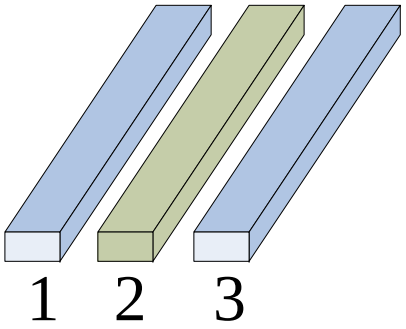


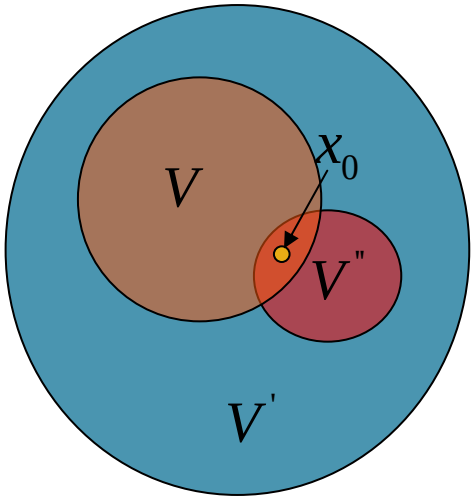
Application-Specific Subspace Selection

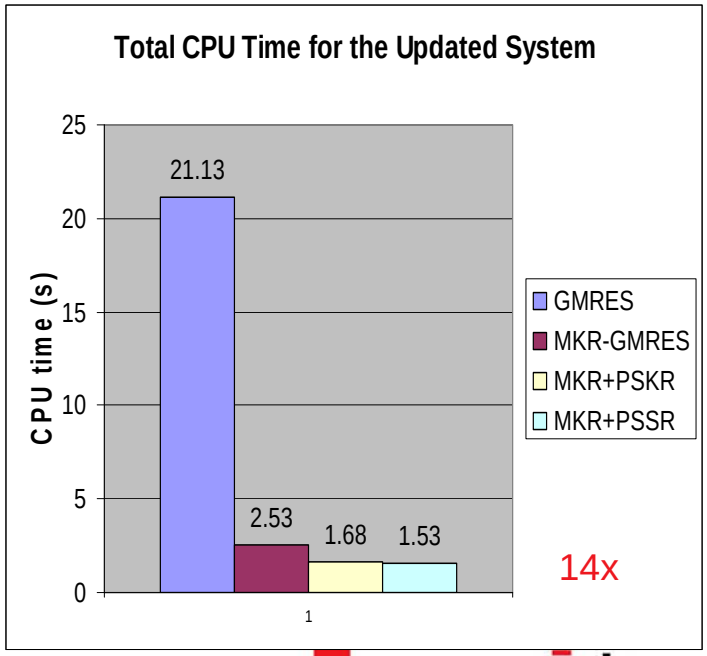
GMRES ⇒ V, x_0

$$V = \overbrace{\begin{bmatrix} V^{(1)} \\ V^{(2)} \\ V^{(3)} \end{bmatrix}}^{q} \qquad V' = \overbrace{\begin{bmatrix} V^{(1)} & & \\ & V^{(2)} & \\ & & V^{(3)} \end{bmatrix}}^{3q} \qquad x_0 = \begin{bmatrix} x_0^{(1)} \\ x_0^{(2)} \\ x_0^{(3)} \end{bmatrix} \qquad V'' = \begin{bmatrix} x_0^{(1)} & & \\ & x_0^{(2)} & \\ & & x_0^{(3)} \end{bmatrix}$$

$$\min_{x \in S_q} \|b - Ax\| = \min \|b - AV_q y\|$$

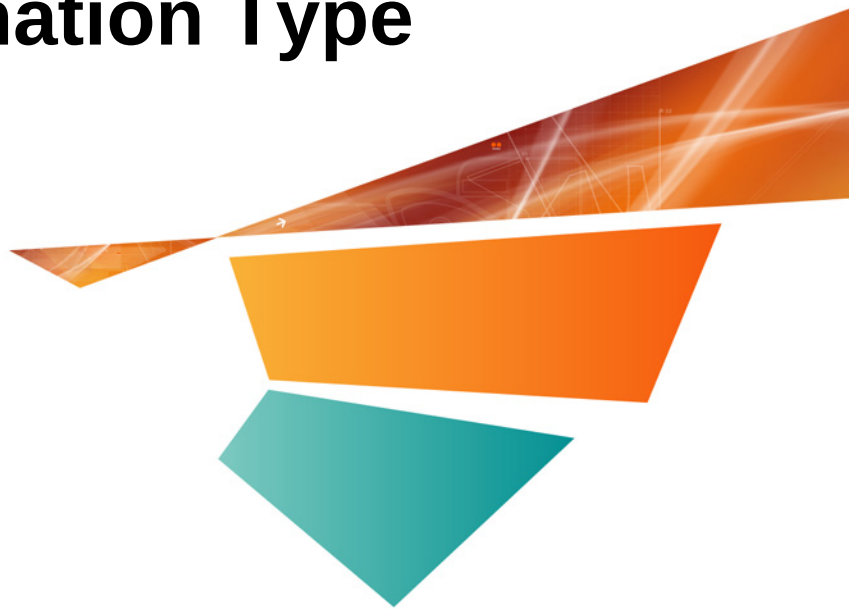






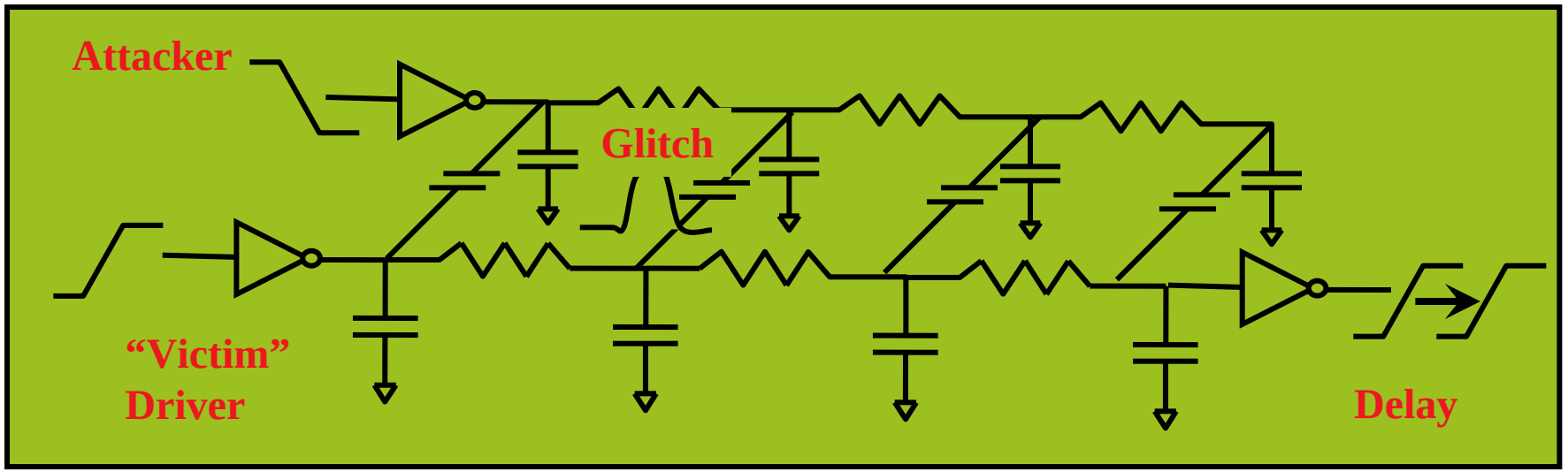
cadence™

Exploiting Internal Structure with Symbolic Elimination Type Approaches



cadence™

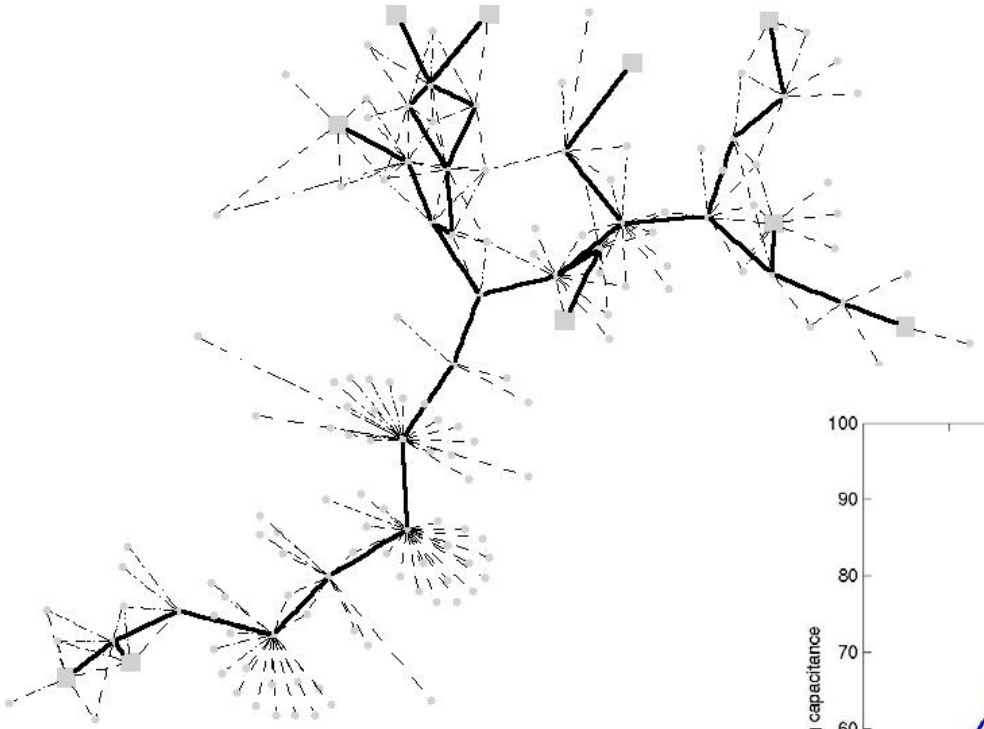
Analysis, One Level Down



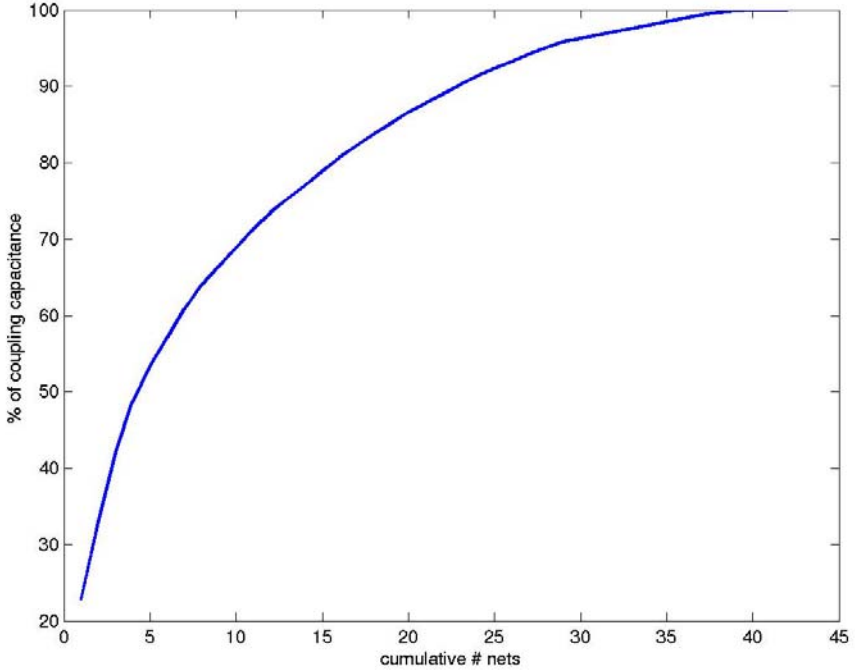
- Driver (Drv)
- Aggressor (Agg)
- Receiver (Rcv)

- Typical questions :
  - How much is the delay from driver input event to switching on receiver input?
    - Simple simulation problem. High accuracy requirement.
  - At what time does the "aggressor" have to switch to cause the *worst-case* change in delay measured at the receiver?
    - Non-convex optimization problem. Moderate accuracy requirement.

Real-World Interconnects Are Complex!

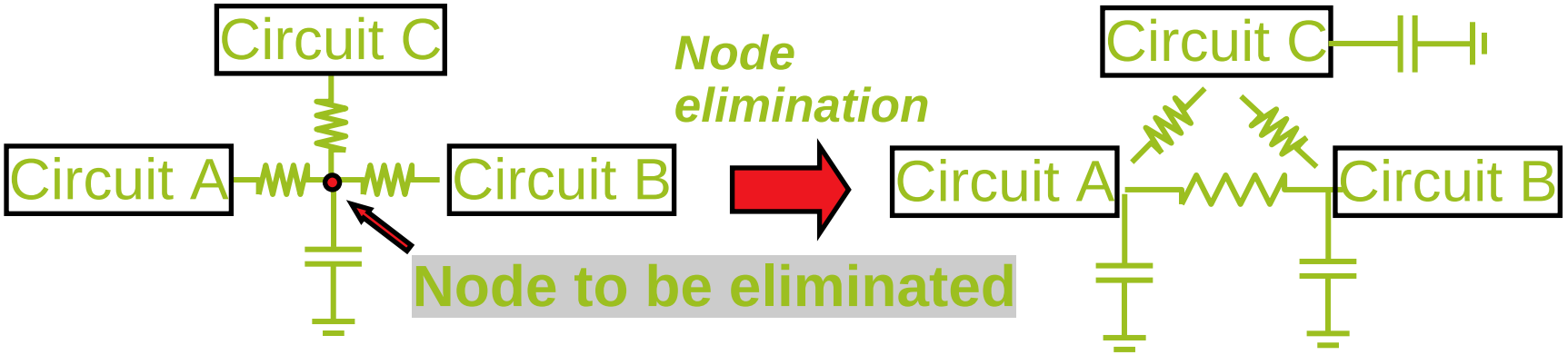


- ~20 internal nodes
- ~10 transistor connections
- > 100 coupling cap connections



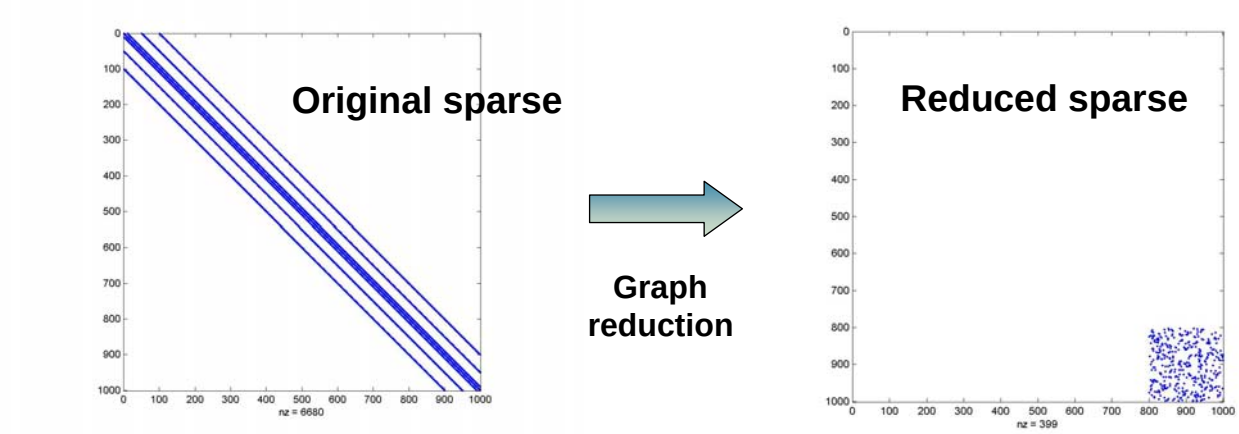
Lots of coupling capacitors!
Lots of nonlinear connections!

“Graph-Based” Reduction for RC Circuits



- 25+ year history in electrical engineering
  - Harbor&Drake, van Genderen & van der Meijs, McCormick, Sheehan
  - Most recent/familiar variant: “TICER”
    - Some heuristics for error control and removing “negative capacitors”
  - Essentially (sparse) symbolic Gaussian elimination [SGE] w/ 1st order Taylor expansion of rational terms (at each step)
- Very popular in industry
  - Why?
  - Is it more than an “industrial hack”?

Why SGE?



- Performance advantages due to specific structure of IC interconnects. Can be ***very fast*** at low accuracy.
- Often produces sparse model output
- Prevalence of “many-port” problems in circuit design
  - Performance doesn’t degrade as fast as other approaches
- Issues
  - Robustness
    - Particularly for TICER
  - Accuracy
    - Some circuits cannot be reduced (at all) with these techniques without severe accuracy loss

More Sparse, Structured Networks:
Large-Scale Interconnect
Analog / MS / RF Circuits

Projection-Based MOR (For Really Huge RC Networks)

$$C\dot{x}(t) + Gx(t) = Eu(t)$$
$$y(t) = E^T x(t)$$

$$x = Mz$$
$$\hat{G} = M^T G M$$
$$\hat{C} = M^T C M$$
$$\hat{E} = M^T E$$

$$\hat{C}\dot{z}(t) + \hat{G}z(t) = \hat{E}u(t)$$
$$y(t) = \hat{E}^T z(t)$$

- Global on-chip interconnect
  - Power distribution networks
  - Clock distribution networks
- Problem sizes can be huge
  - rank E = m = $O(10^3)$ to $O(10^4)$
  - rank G = n = $O(10^8)$ to $O(10^9)$
- Cannot do much if performing these computations in the usual way
  - PVL, PRIMA, etc.
  - $O(10^{12})$ bytes to store M – *for a first-order approximation*
  - $O(10^{15})$ operations to compute an inner product $X^T Y$ where X,Y are n X m

Sparse Implicit Projection – Observation #1

- Consider single-point (first-order) moment matching
- One column in projection matrix obtained from following linear system:

$$Gv = E$$

- Write in partitioned form
  - often the case that “ports” are a sub-set of the circuit nodes (not a necessary assumption but instructive to work this case)

$$\begin{bmatrix} A & B \\ B^T & D \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

- Equivalent to solving the Schur-complement system

$$Sv_2 = I \qquad S = (D - B^T A^{-1} B)$$

Sparse Implicit Projection – Observation #2

- Schur complement can be obtained by congruence (orthogonal projection)

$$M = \begin{bmatrix} -A^{-1}B \\ I \end{bmatrix} \qquad S = M^T G M$$

- In fact, M produces a useful reduced-order model
  - Matches two moments in this case
  - Passivity preserving for RC circuits & similar
  - Same column span as "usual" projector V. Specifically,

$$V = MS^{-1}$$

  - Nothing novel or useful so far

Sparse Implicit Projection – Observation #3

- Consider Cholesky factorization of G-matrix

$$G = \begin{bmatrix} L_1 & \\ L_3 & L_2 \end{bmatrix} \begin{bmatrix} L_1^T & L_3^T \\ & L_2^T \end{bmatrix}$$

- The Schur complement (or, equivalent, reduced-G) can be obtained directly from the Cholesky factor
  - Actually it is better to note complete the factorization (obtain S from intermediate step of right-looking factorization)

$$S = L_2 L_2^T$$

  - Reduced C can be obtained from applying the same operations (e.g., set of Gaussian elimination steps) to C-matrix
  - M is never constructed explicitly (will usually be dense)

Multi-point projection

$$G_i = G + s_i C$$

$$M = \begin{bmatrix} M_1 & M_2 & \cdots & M_q \end{bmatrix}$$

$$\begin{aligned} (sC + G_i)x &= Eu \\ y &= E^T x \end{aligned}$$

Shifted system

$$M^T G M = \begin{bmatrix} \hat{G}_{11} & \cdots & \hat{G}_{1q} \\ \vdots & \ddots & \vdots \\ \hat{G}_{q1} & \cdots & \hat{G}_{qq} \end{bmatrix} \qquad \hat{G}_{ij} = M_i^T G M_j$$

$$M^T C M = \begin{bmatrix} \hat{C}_{11} & \cdots & \hat{C}_{1q} \\ \vdots & \ddots & \vdots \\ \hat{C}_{q1} & \cdots & \hat{C}_{qq} \end{bmatrix} \qquad \hat{C}_{ij} = M_i^T C M_j$$

$$G_i = \begin{bmatrix} A_i & B_i \\ B_i^T & D_i \end{bmatrix} \Rightarrow M_i = \begin{bmatrix} -A_i^T B_i \\ I \end{bmatrix}$$

- Again…M not constructed, work with 1-step sparse transforms
- For q sampling point, cost is
  - CPU time: at worst q^2 * single point SIP.
  - Clever trick for symmetric, build from diagonal terms in O(q) time
  - No additional memory requirement compared to single point SIP.

SIP Summary

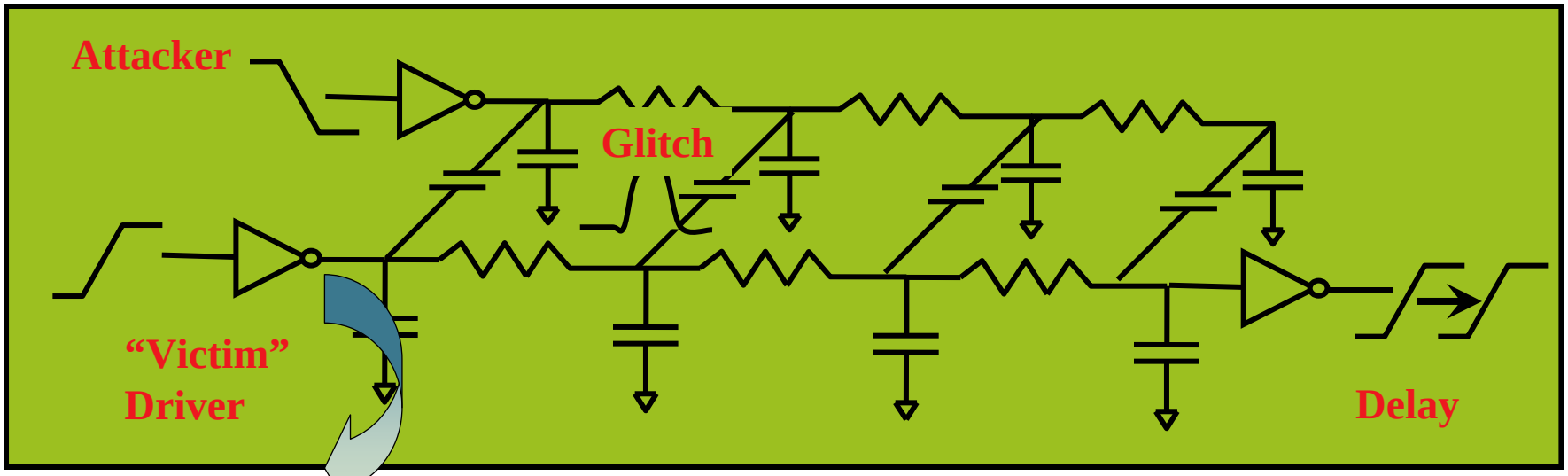
- Main insight: reduced model can be obtained using more or less off-the-shelf sparse-matrix operations
  - Projection computed using Gaussian elimination.
  - No top-level Krylov iteration. Stop the factorization in the middle, paste together models from multiple points, done.
- Mathematically, same as the standard projection formalism.
  - Single-point model is the exact same model as 1st-order multivariate Pade (syPVL, PRIMA), in different coordinates
  - Multi-point is rational Krylov
- Computationally, can be vastly different.
  - Many circuit matrices can be factored with very little fill. Final models are often sparse. Huge advantage for downstream analysis.
  - No inner products
  - Difference of $O(10^{15})$ vs. $O(10^{9})$ operations
    - 1000X difference in speed, 100X in memory on *moderate* sized examples
- Similar analysis holds in non-symmetric case & LU factorization

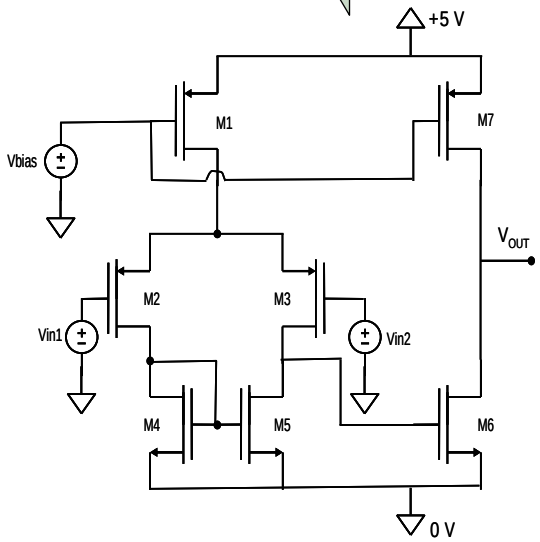
Problem Areas

- For some problems, one really does need the inner product operation
  - Projector can lose rank (in exact arithmetic)
  - Can be fixed locally, and cheaply, for many problems of interest

- Need a fix that is:
  - Robust for circuit networks
  - General
  - Cheap (i.e., linear in port count)

Model Order Reduction for Nonlinear Circuits



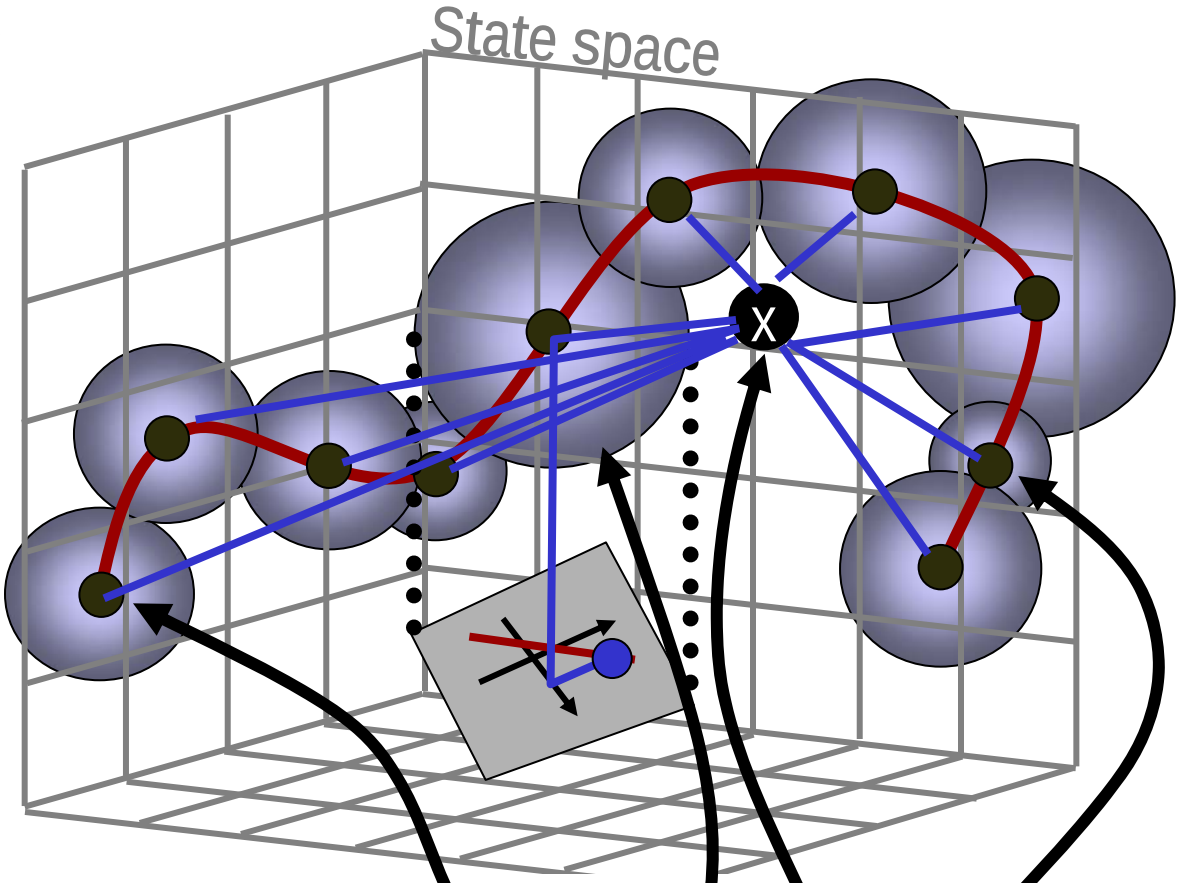


- Now we will consider the cell models (transistors)
- Problems
  - Traditional compact models are breaking down
  - Detailed transistor effects are important
  - Waveform details are important
  - Too many cases and effects to model "by-hand"

cadence™

Trajectory-Based Nonlinear Macromodeling

- Simulate training inputs in state space
- Choose center pts x_i on this trajectory
- Linearize near each pt, projecting down to a smaller state vec
- Reduce linearized models via projection
- Combine linearized models



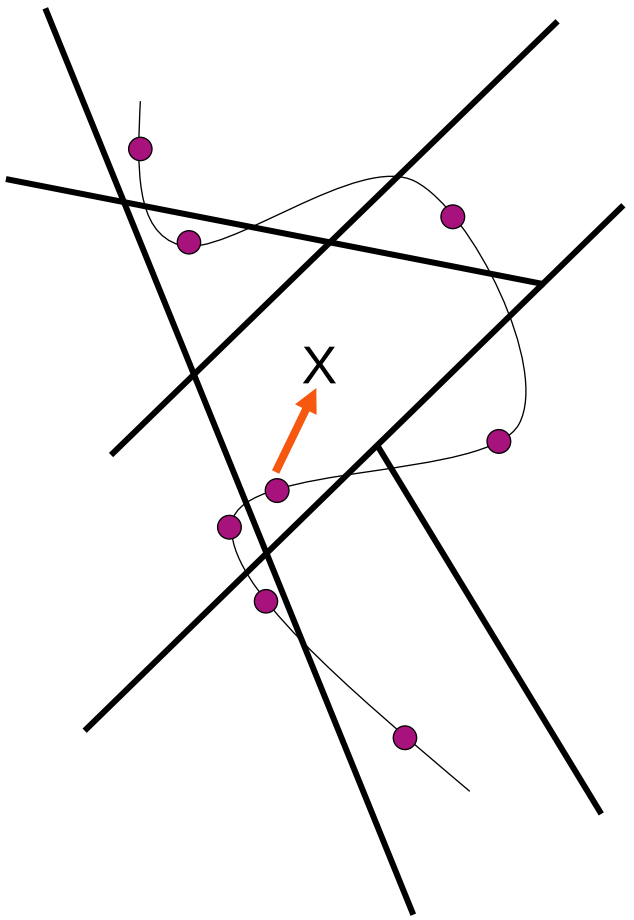
$$f_{Traj}(x) = \sum_{i=1}^{s} \overbrace{w_i(x)}^{\sum w_i(x)=1} \left[f(x_i) + \overbrace{A_i}^{\text{matrix}} (x - x_i) \right]$$

Practical Issues

- Coverage
  - Do the points (waveforms) obtained in characterization represent future inputs adequately?
- Interpolation
  - What is the best way to combine the models? How can continuity in the models be preserved?
- Stability
  - How can we ensure the reduced order model will be well-behaved?

TBM Error Analysis

- Error sources
  - reduction error
  - linearization error
  - sampling error – do we have enough samples given unknown inputs?
- Observation: TBMs de-facto tesselate the reduced state space
  - We can obtain error estimates from the next higher term in the Taylor series (quadratic terms)
  - Can obtain a-posteriori error analysis



$$f_1(x) = f_0 + J_1^T x + error$$

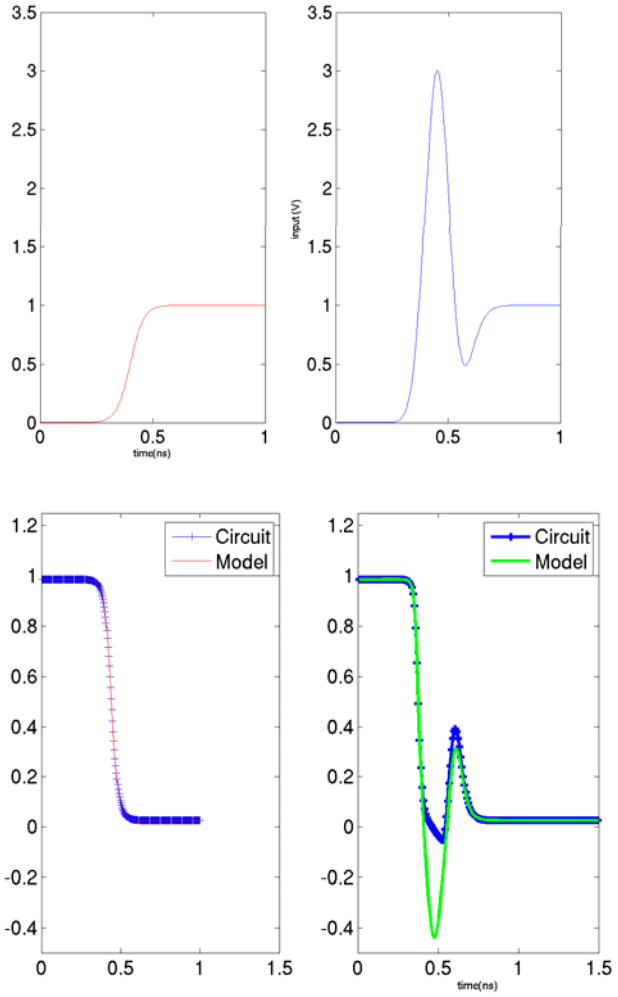
$$error \approx x^T J_2 x$$

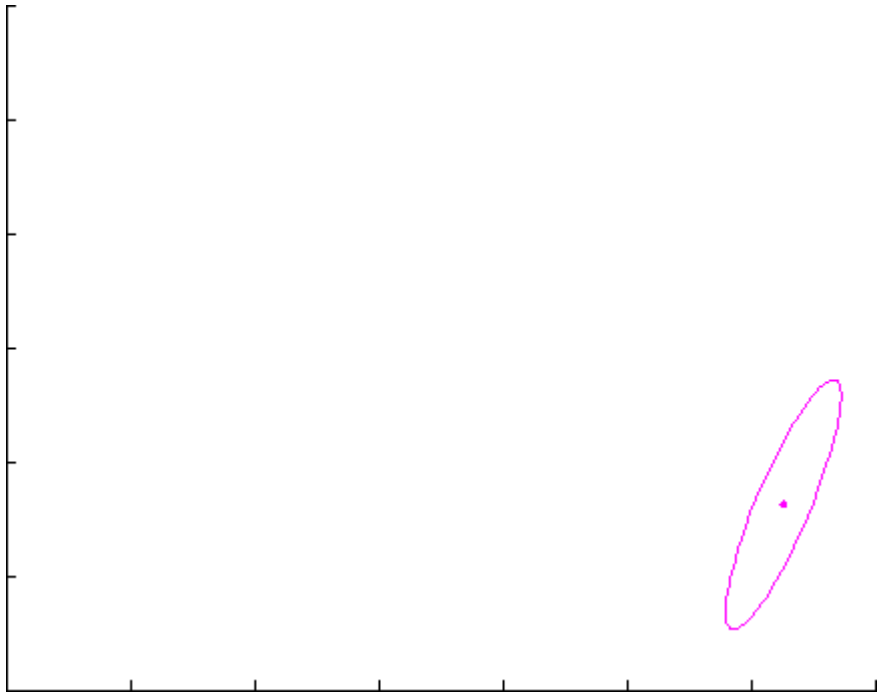
Error Analysis Observations

- Inputs have specific structure
- Don’t need analysis for arbitrary inputs, or in fact knowledge of error
- Need to know if *specific* model meets spec for *specific* inputs expected to be encountered

Example: Model Fidelity Assessment

- Consider “extreme” cell model input





- Valid/invalid models correctly identified

cadence™

TBM Error Analysis

- In principle, given
  - A specific model
  - A specific input set
  - An error constraint
- It is possible to (automatically) determine if the model meets the required error constraints
- With some uncertainty / degree of conservatism
  - Similar issues to interval arithmetic

Summary of Themes for MOR

- Scale
  - Execution time for MOR is a first-order design constraint.
  - May be millions of systems to reduce
- Incrementality
  - May be dozens of variants of each system
  - More work on incremental approaches would be very useful
- Sparse, structured systems
  - Most circuit topologies have sparse structure that is only partially exploited by most MOR algorithms
  - System updates are often themselves structured in ways that can be exploited
- Context-aware [error] analysis
  - Inputs, outputs, internal system dynamics: none are arbitrary.
  - Error analysis can / should exploit this