SAMSON: A Generalized Second-order Arnoldi Method for Reducing Multiple Source Linear Network with Susceptance

Yiyu Shi, Hao Yu and Lei He

EE Department, UCLA

Partially supported by NSF and UC MICRO Analog Devices, Intel and Mindspeed

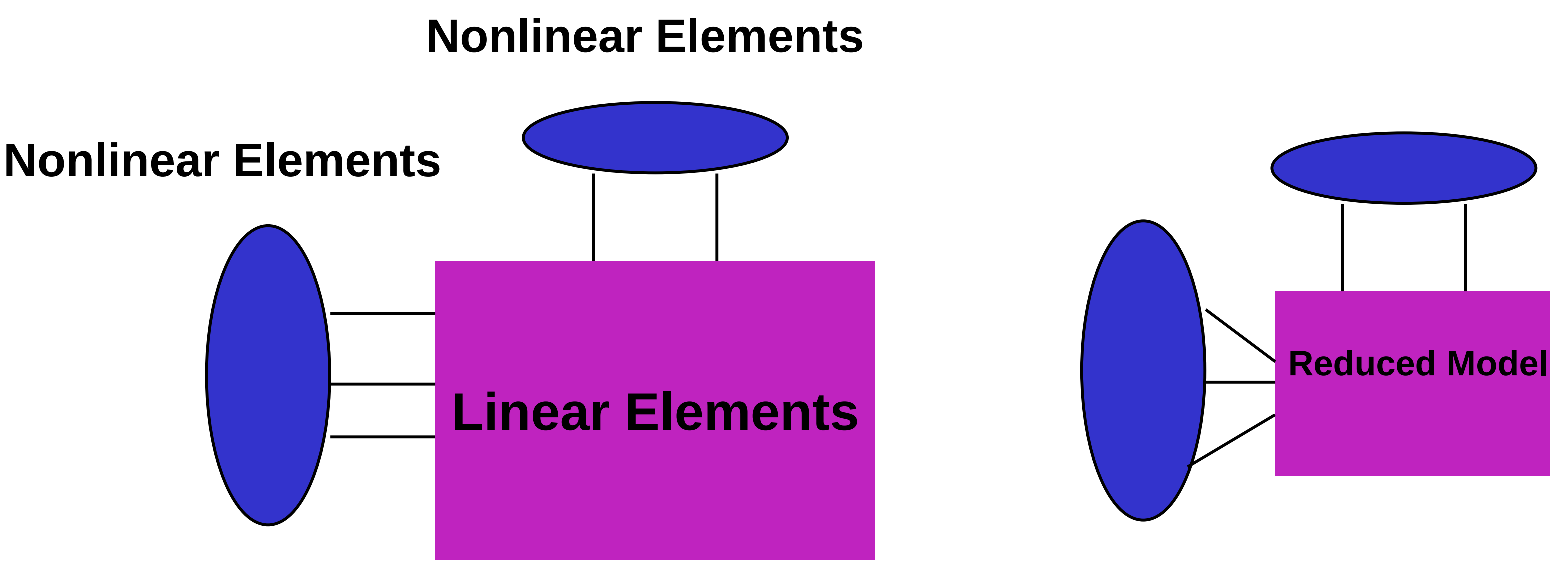
Address comments to lhe@ee.ucla.edu

Outline

- Review and Motivation
- SAMSON Algorithm
- Experimental Results
- Conclusions

Motivation for Model Order Reduction

- Deep submicron design needs to consider a large number of linear elements
  - Interconnect, Substrate, P/G grid, and Package
- Accurate extraction leads to the explosion of data storage and runtime
- Need efficient macro-model



Non-RHS MOR vs. RHS MOR

- Non-RHS MOR reduces transfer function H(s)
  - First Order Method to handle L
    - PRIMA [Pileggi et al, TCAD'98]
  - Second Order Methods to handle S (susceptance)
    - ENOR [Sheehan, DAC'99]
    - SMOR [Pileggi et al, ICCAD'02]
    - SAPOR [Su et al, ICCAD'04] [Liu et al, ASPDAC'05]
- Main Limitation
  - Can only match up to ceil(n/n_p) moments
  - Accuracy is significantly limited when the port number n_p is large

- RHS MOR reduces output vector y(s)
  - [Chiprout, ICCAD'04]
    - explicit moment matching is used (lack in numerical stability)
  - EKS [J. M. Wang et al, DAC'00]
    - Implicit moment matching based on Incremental orthonormalization
    - frequency domain shifting to deal with 1/s and $1/s^2$ terms
  - IEKS [Y. Lee et al, TCAD'05]
    - Based on the observation that there are no 1/s and $1/s^2$ terms for PWL sources in finite time

Problems Still Remain Unsolved

- None of the existing RHS MOR methods can deal with RCS circuits with susceptance elements
  - Both EKS and IEKS are first order methods
  - Directly applying them to RCS circuits cannot guarantee passivity
- There is still much room to improve accuracy
  - Incremental Orthonormalization causes error to accumulate
  - When matching high order moments, it becomes inaccurate
- None of the existing MOR methods can handle arbitrary independent inputs
  - Especially when they contain $1/s^i$ terms (i>0)
    - frequency domain shifting (inaccurate)
  - In the existence of infinite PWL sources
    - EKS and IEKS cannot consider s=0 (cannot perform DC analysis)

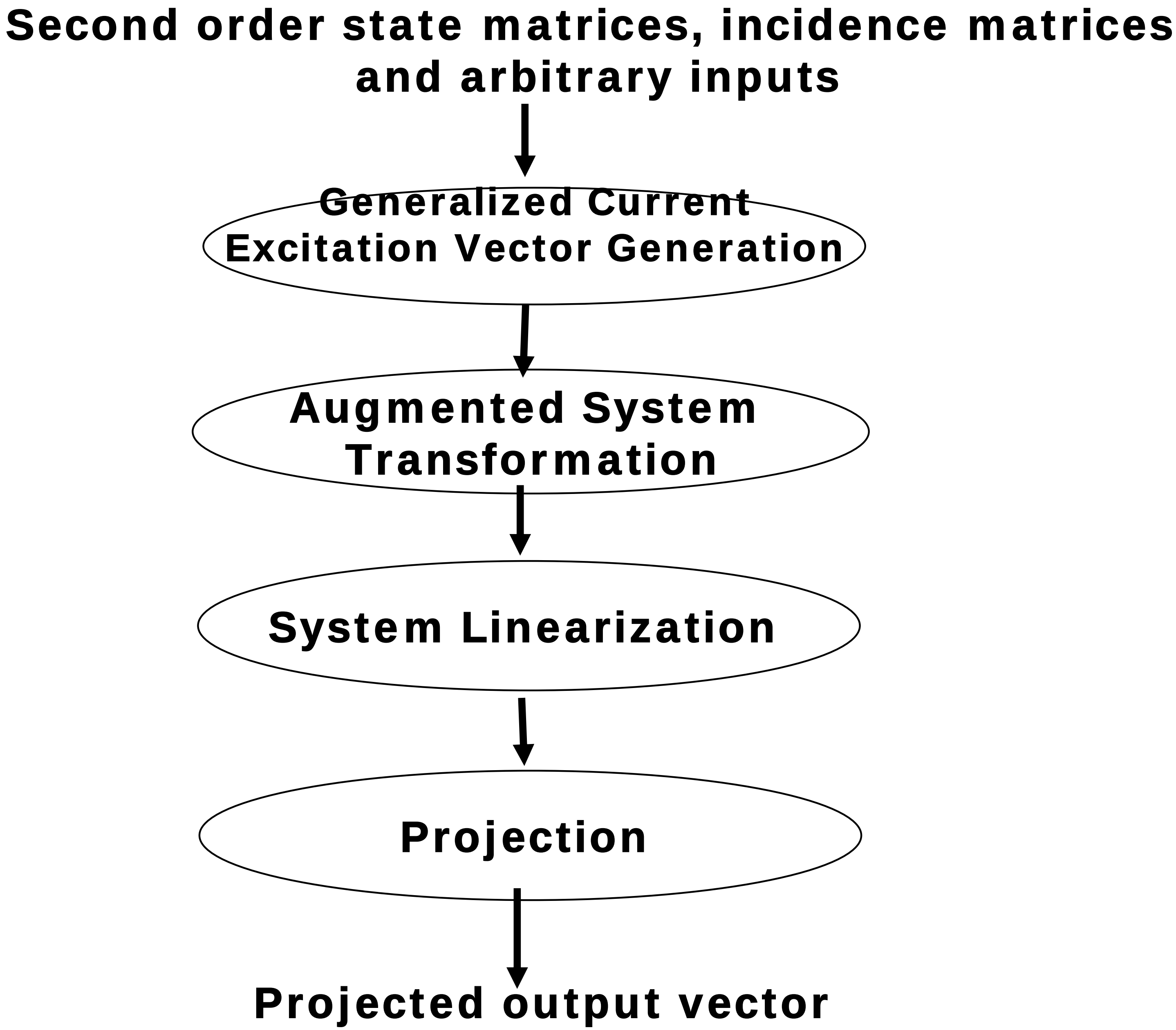
Major Contributions of SAMSON

- It is an RHS MOR method
  - Compare with SAPOR and other non-RHS methods, it is more accurate
  - Can handle a large number of ports
- It can deal with all kinds of input sources accurately without frequency domain shifting or incremental orthonormalization
  - Numerically more stable, more efficient and more accurate in the whole frequency domain, especially at DC (s=0)
- It is based upon generalized second order Arnoldi method
  - Can handle RCS circuits with passivity

Outline

- Review and Motivation
- **SAMSON Algorithm**
- Experimental Results
- Conclusions

SAMSON Algorithm: Overview



Generalized Current Excitation Vector

$$(G + sC + \frac{\Gamma}{s})V(s) = BJ_e$$

Original system equation

Superposed SIMO system equation

$$\sum_{i=0}^{m+1} \gamma_i s^i V(s) = \sum_{i=0}^{n} \Theta_i s^i$$

- Define the generalized current excitation vector J_{ex} as

$$J_{ex} = BJ_e$$

- J_{ex} can be divided into two categories
  - Rational

$$J_{ex}(s) = \frac{a_0 + a_1 s + \ldots + a_n s^n}{b_0 + b_1 s + \ldots + b_m s^m}$$

  - Irrational
    - Can be expanded into Taylor series and take dominant terms. Then it becomes a special case of the Rational category

$$J_{ex}(s) = \sum_{i=0}^{n} J_i s^{i-m}$$

- Multiply both sides by

$$b_0 + b_1 s + \ldots + b_n s^m$$

SAMSON Algorithm: Overview

Second order state matrices, incidence matrices and arbitrary inputs

↓

Generalized Current Excitation Vector Generation

↓

Augmented System Transformation

↓

System Linearization

↓

Projection

↓

Projected output vector

Augmented System Transformation

$$\sum_{i=0}^{m+1} \gamma_i s^i V(s) = \sum_{i=0}^{n} \Theta_i s^i$$

Superposed SIMO system equation

Augmented system equation

$$\sum_{i=0}^{n+1} \Psi_i s^i U = \begin{bmatrix} \sum_{i=0}^{n} \Theta_i s^i \\ 0 \end{bmatrix}$$

$$U = [V, V_1, V_2, \ldots, V_{n-m}]^T$$

- To linearize, we have to make LHS exactly one order higher than RHS by raising the order of LHS by n-m .
  - Introduce auxiliary variables V_1, V_2, …, V_n satisfying

$$V = sV_1, \quad V_1 = sV_2,$$

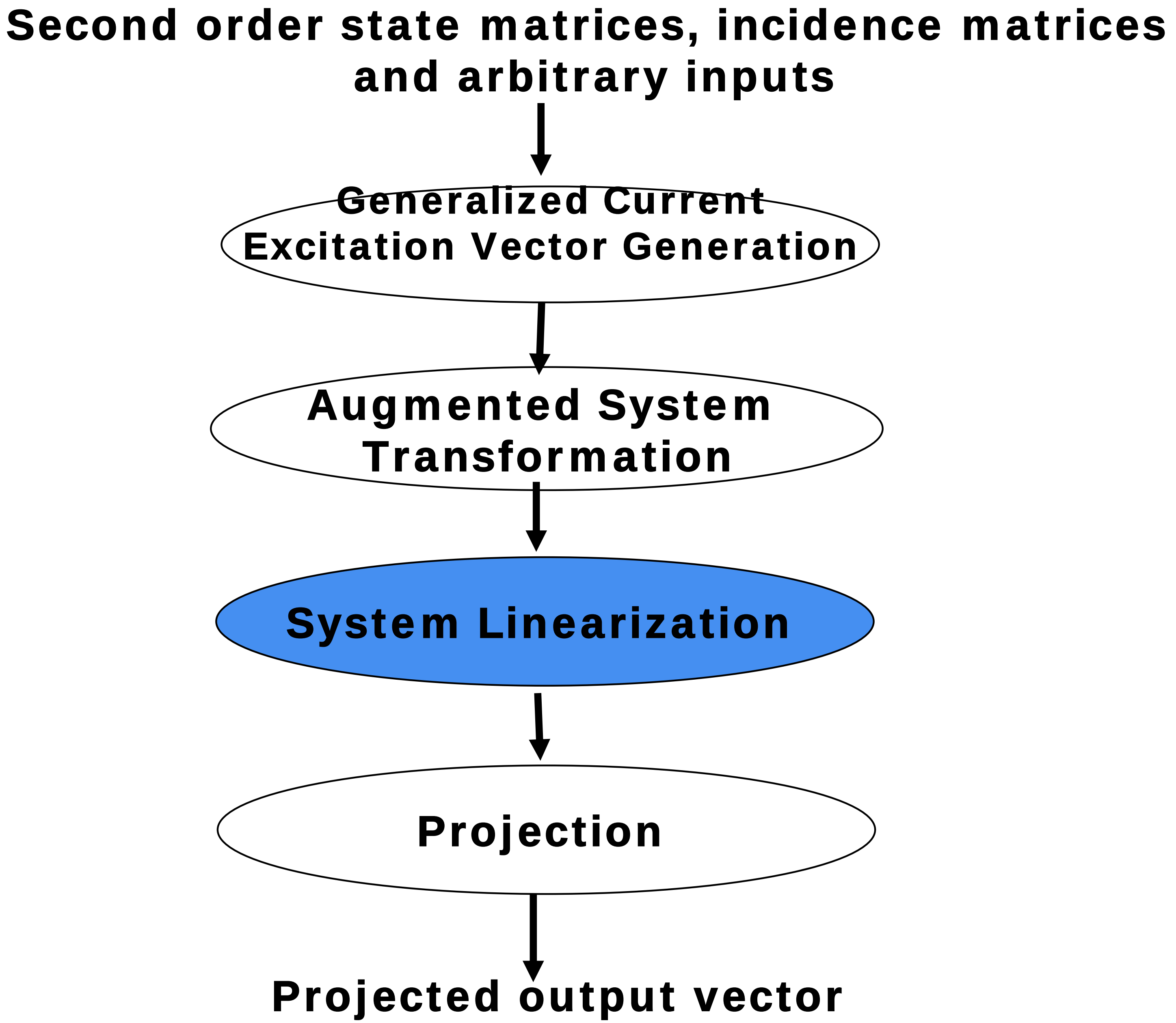
$$\ldots, \quad V_{n-m-1} = sV_{n-m}$$

Here we assume m<n. The transformation for the n<m case can be derived similarly

  - Insert into superposed SIMO system equation =>

$$\sum_{i=0}^{m+1} \gamma_i s^{i+n-m} V_{n-m}(s) = \sum_{i=0}^{n} \Theta_i s^i$$

SAMSON Algorithm: Overview



System Linearization

$$\sum_{i=0}^{n+1} \Psi_i s^i U = \begin{bmatrix} \sum_{i=0}^{n} \Theta_i s^i \\ 0 \end{bmatrix}$$

$$U = [V, V_1, V_2, \ldots, V_{n-m}]^T$$

Augmented system equation

e.g., $A_{n+1}\sigma U + Z_n = R_n$

$$\Rightarrow A_{n+1}\sigma^{n+1} U + Z_n \sigma^n = R_n \sigma^n$$

Insert it into

$$A_{n+1}\sigma^{n+1} U + \ldots = R_n \sigma^n + \ldots$$

and we get

$$(A_n U + Z_n)\sigma^n + \ldots = R_{n-1}\sigma^{n-1} + \ldots$$

- Expand the augmented system at a given frequency $s = s_0 + \sigma$

$$\Rightarrow \sum_{i=0}^{n+1} A_i \sigma^i U(\sigma) = \sum_{i=0}^{n} R_i \sigma^i$$

- Introduce auxiliary variables Z_1, Z_2, …Z_n, satisfying

$$A_{n+1} U \sigma + Z_n = R_n$$

$$(A_n U - Z_n)\sigma + Z_{n-1} = R_{n-1}$$

$$\ldots$$

$$(A_0 U - Z_1)\sigma + Z_1 = R_0$$

and we can obtain

$$(A_0 + A_1 \sigma) U - \sigma Z_1 = R_0$$

System Linearization

$$\sum_{i=0}^{n+1}\Psi_i s^i U = \begin{bmatrix}\sum_{i=0}^{n}\Theta_i s^i \\ 0\end{bmatrix}$$

$$U = [V, V_1, V_2, \ldots, V_{n-m}]^T$$

Augmented system equation

Moment space equation

$$(I-\sigma A)\begin{bmatrix}V\\ D\end{bmatrix} = \begin{bmatrix}q_0\\ p_0\end{bmatrix}$$

- Expand the augmented system at a given frequency $s = s_0 + \sigma$

$$\Rightarrow \sum_{i=0}^{n+1} A_i \sigma^i U(\sigma) = \sum_{i=0}^{n} R_i \sigma^i$$

- Introduce auxiliary variables Z_1, Z_2, …Z_n, satisfying

$$A_{n+1} U\sigma + Z_n = R_n$$

$$(A_n U - Z_n)\sigma + Z_{n-1} = R_{n-1}$$

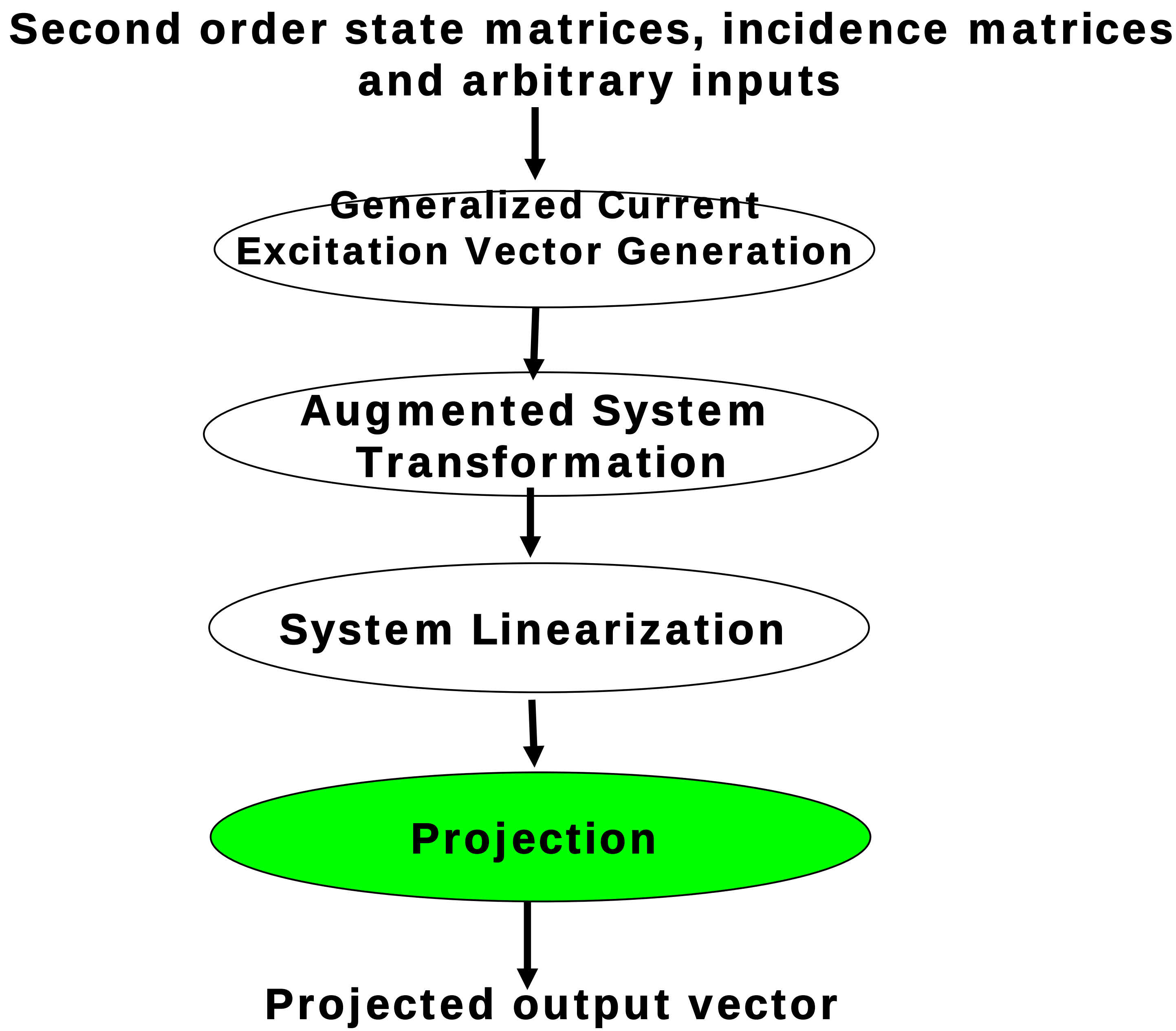
$$\ldots$$

$$(A_0 U - Z_1)\sigma + Z_1 = R_0$$

and we can obtain

$$(A_0 + A_1\sigma)U - \sigma Z_1 = R_0$$

SAMSON Algorithm: Overview



Projection

$$(I-\sigma A)\begin{bmatrix} V \\ D \end{bmatrix} = \begin{bmatrix} q_0 \\ p_0 \end{bmatrix}$$

Moment space equation

- Find projection matrix using method similar to PRIMA
- But only take the first N rows (N is the number of the original variables)
- The moment calculation is efficient because the linearized system is sparse

Projection matrix Q

$$\hat{C} = Q^T C Q,\ \hat{G} = Q^T G Q,\ \hat{\Gamma} = Q^T \Gamma Q,\ \hat{V} = Q^T V$$

$$\hat{J}_{ex} = Q^T J_{ex}$$

Projected system equation

$$(\hat{G} + s\hat{C} + \frac{\hat{\Gamma}}{s})\hat{V}(s) = \hat{J}_{ex}$$

- Directly project on J_{ex} instead of B

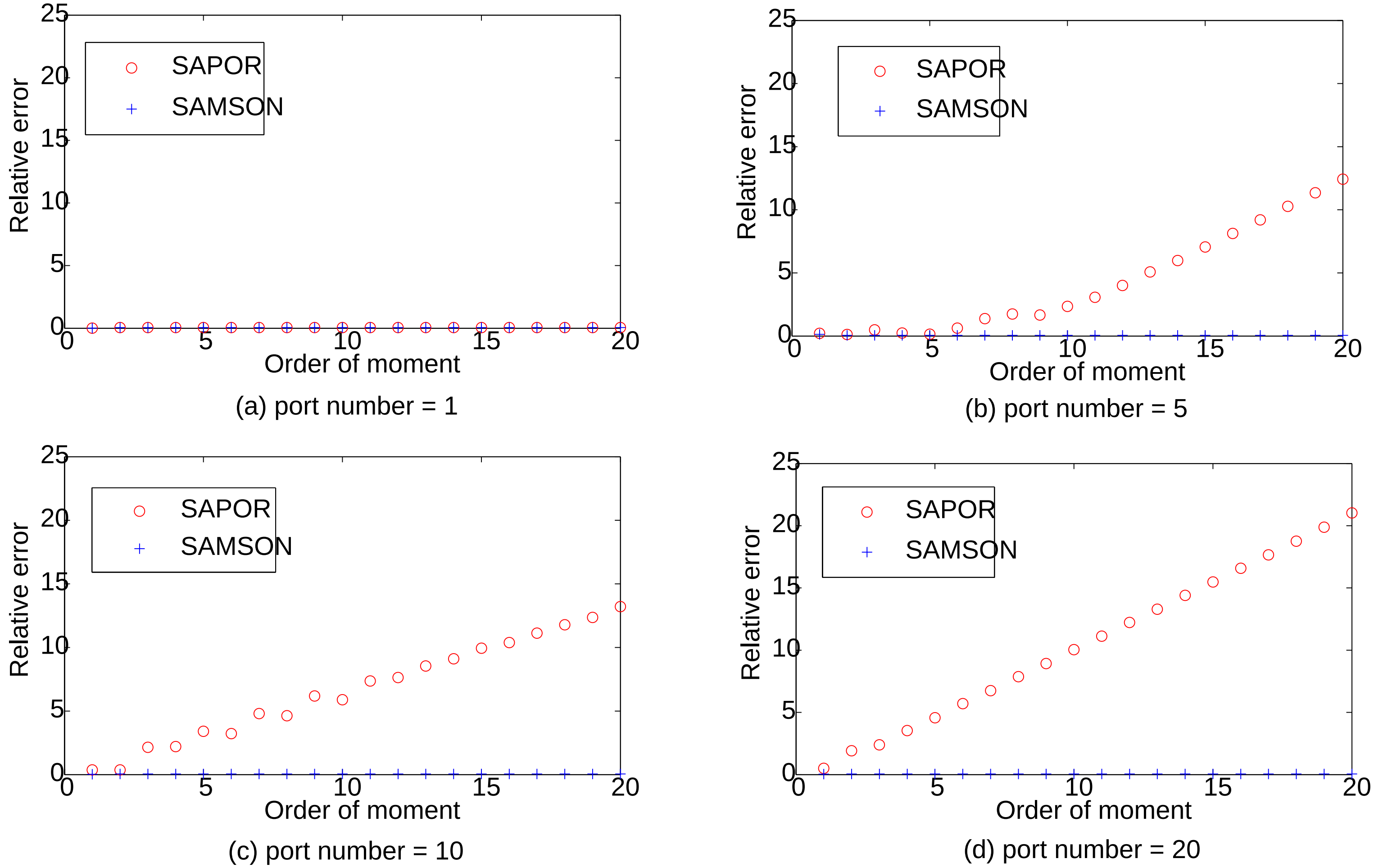
Outline

- Review and Motivation
- SAMSON Algorithm
- **Experimental Results**
- Conclusions

Experimental Setting

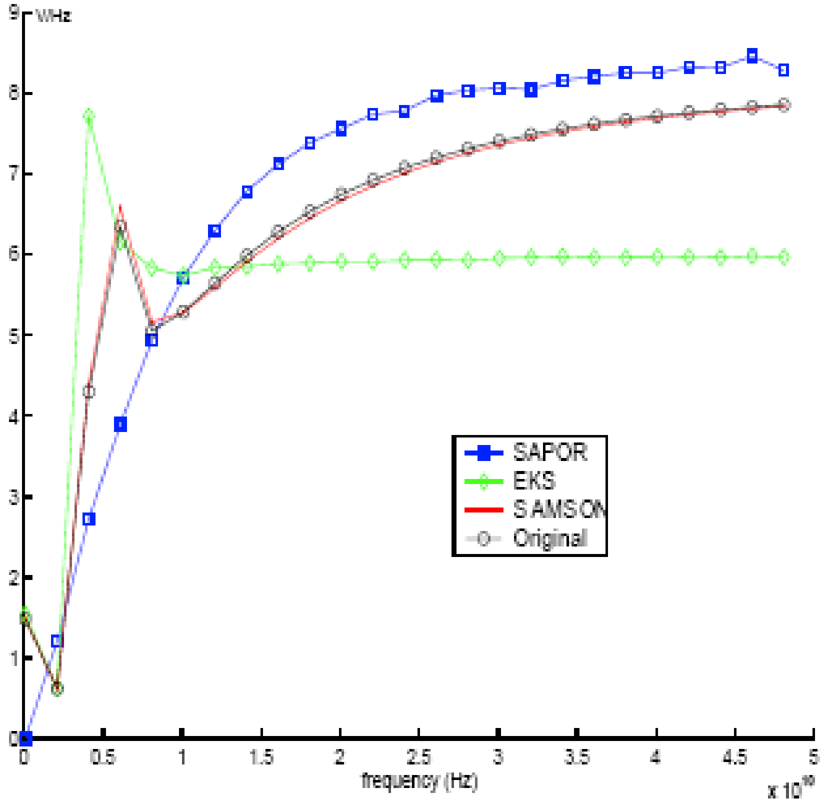
- Experiments are run on a PC with Intel Pentium IV 2.66G CPU and 1G RAM.
- All methods are implemented in MATLAB
- Time domain responses are calculated by IFFT (Inverse Fast Fourier Transformation with 1024 sampling points)
- The examples to be presented are from real industry applications (courtesy of Rio Design Automation)
  - Power planes and packages are modeled by RCS meshes
  - On chip power/ground grids are modeled by RC meshes
  - Vias and bumps are modeled by RC elements
  - PWL sources are generated from SPICE characterization of FPGA circuits

Moment Matching

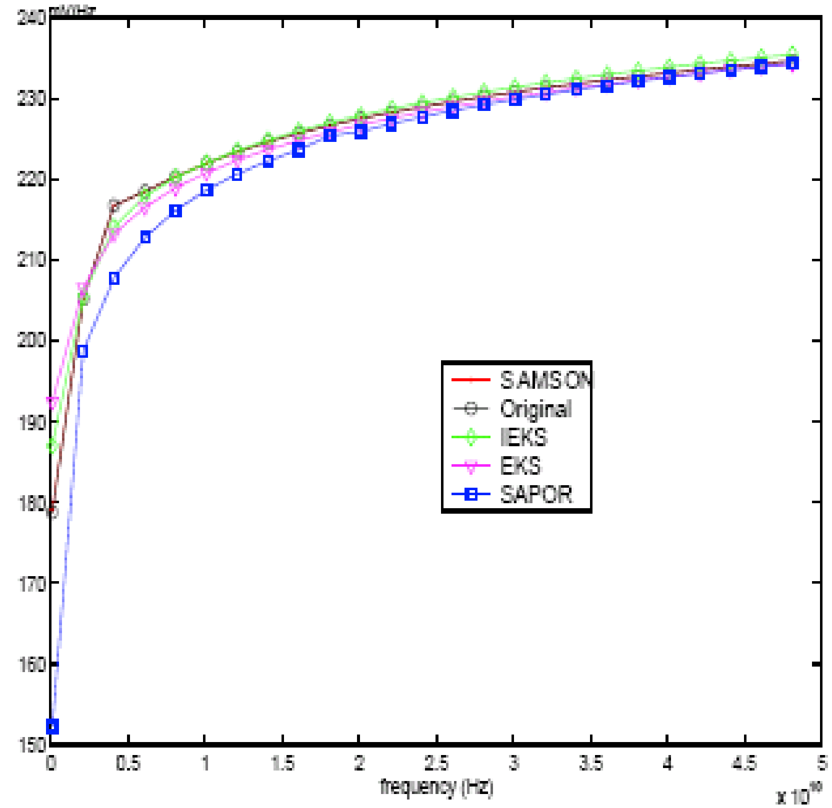


- With the increase of port number (from 1 to 20)
  - SAPOR (second order non-RHS) matches a decreasing number of moments
  - SAMSON always matches the same number of moments.

Frequency Domain Accuracy Comparison



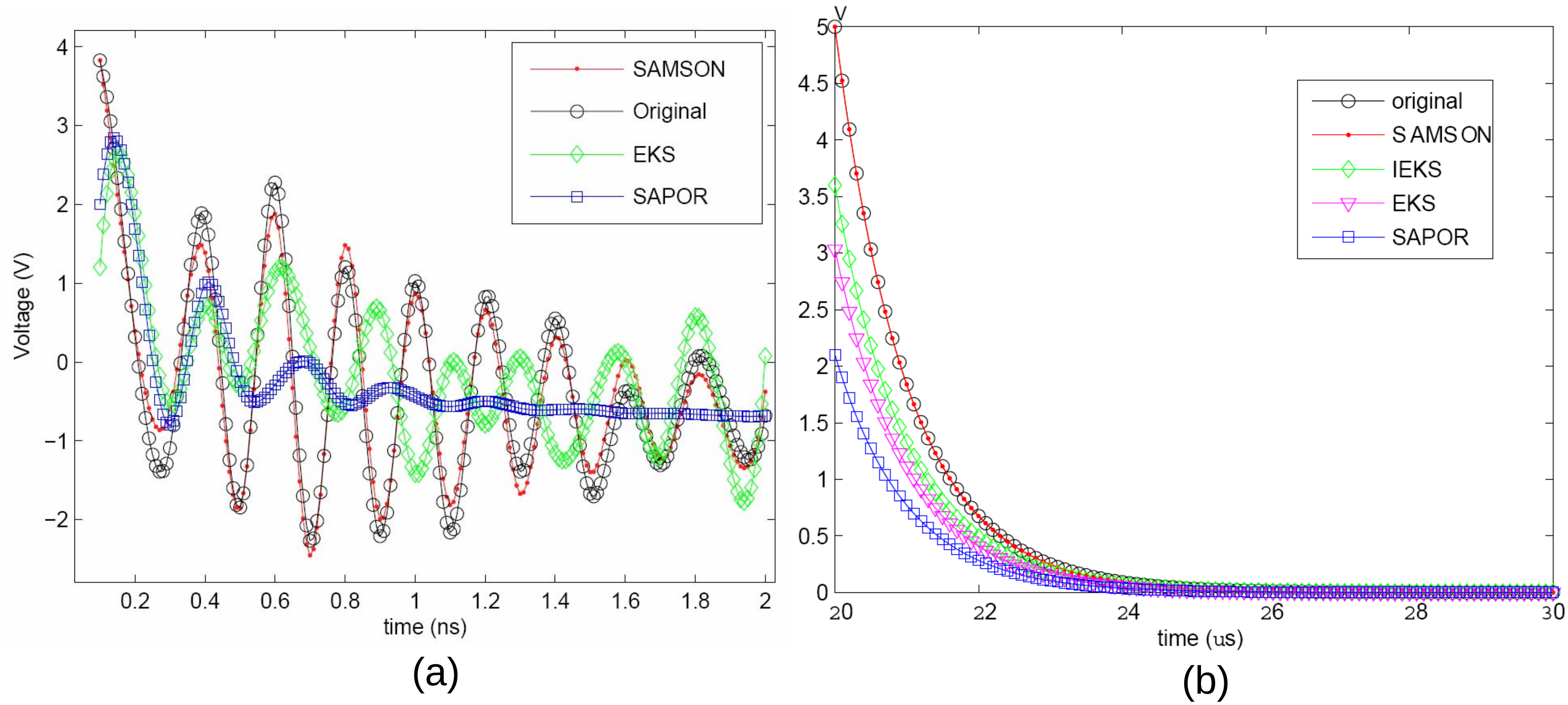
(a)



(b)

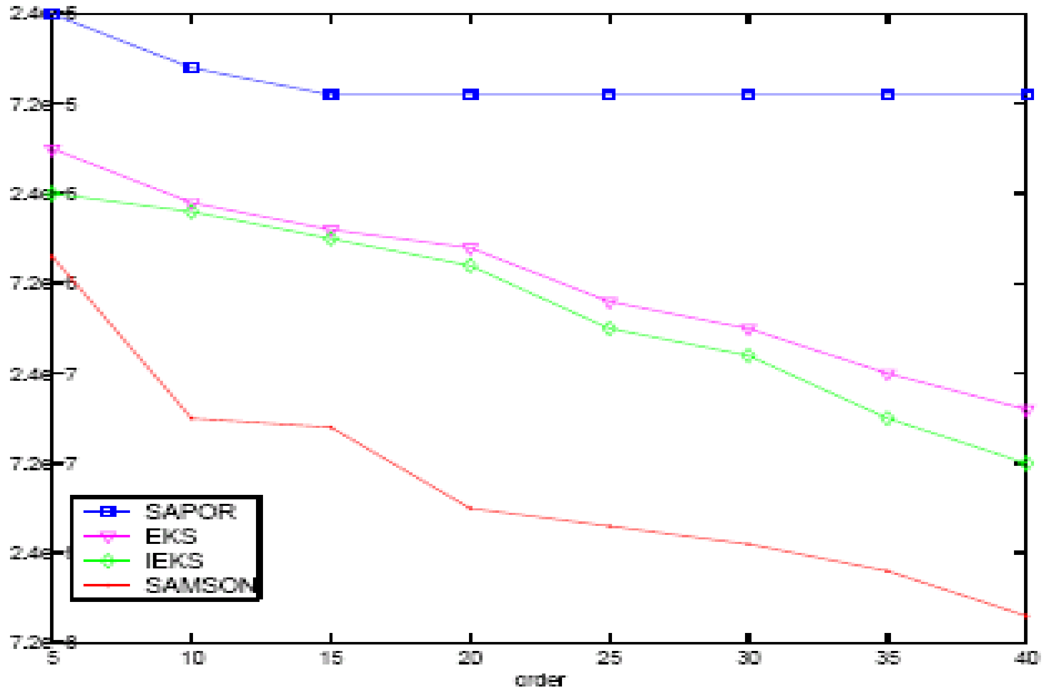
- ***(a) Frequency domain comparison between SAPOR, EKS, SAMSON and original with attenuated sine waveforms . (b) Frequency domain comparison between SAMSON, Original, IEKS, EKS and SAPOR with PWL sources; All circuits are reduced to the same order.***
  - Only SAMSON is identical to the original
  - EKS outperforms SAPOR due to the RHS MOR nature

Time Domain Accuracy Comparison



- ***(a) Time domain comparison between SAPOR, EKS, SAMSON and original with attenuated sine waveforms . (b) Time domain comparison between SAMSON, Original, IEKS, EKS and SAPOR with PWL sources; All circuits are reduced to the same order.***
  - Again only SAMSON is identical to the original
  - EKS outperforms SAPOR due to the RHS MOR nature

Scalability and Accuracy



- *Average time domain waveform error of SAMSON, EKS, IEKS and SAPOR with respect to the reduced order*
- *Different sizes of circuits from 200-70000 nodes are used. For each circuit 30% ports have independent PWL sources.*
- ***SAMSON*** has the fastest waveform convergence
  - 33X more accurate than EKS and IEKS at order 40
  - 48 X more accurate than SAPOR
  - The non-RHS method, SAPOR, does not converge

Runtime Comparison

# of nodes	# of sources	Cir Sim Time (s)	Reduction + Simulation Time (s)		
			EKS	IEKS	SAMSON
192	50	0.18	0.11+0.00	0.09+0.00	0.08+0.00
768	100	106	10.4+0.4	10.2+0.4	7.6+0.4
2048	200	362	20.6+0.8	20.4+0.8	15.8+0.8
11520	800	1164	66.1+3.2	65.2+3.2	47.3+3.2
69380	4000	N/A	384+92	381+92	295+92

- *Comparison of the reduction and simulation time under the same accuracy of up to 50 GHz on an RC mesh with 11,520 nodes and 800 ports*
- ***SAMSON*** runs the fastest
  - 25X faster than direct simulation
  - faster than EKS and IEKS
  - but have a similar trend

Conclusions and Future work

- SAMSON is an RHS MOR method
  - Compare with SAPOR and other non-RHS methods, it is more accurate
  - Can handle a large number of ports
- SAMSON can deal with all kinds of input sources accurately without frequency domain shifting or incremental orthonormalization
  - Numerically more stable, more efficient and more accurate in the whole frequency range, particularly at DC (s=0)
- SAMSON is based upon generalized second order Arnoldi method
  - Can handle RCS circuits with passivity

Thank you!