Reduced Order Modeling

: Linear VLSI Systems

Vineet Sahula, *sahula@ieee.org*

Short Term course on
Mathematical Methods and its Applications in Engg & Sc. 29^{th} Oct. – 3^{rd} Nov 2012

Malaviya National Institute of Technology, Jaipur

Outline

Physical System Classification

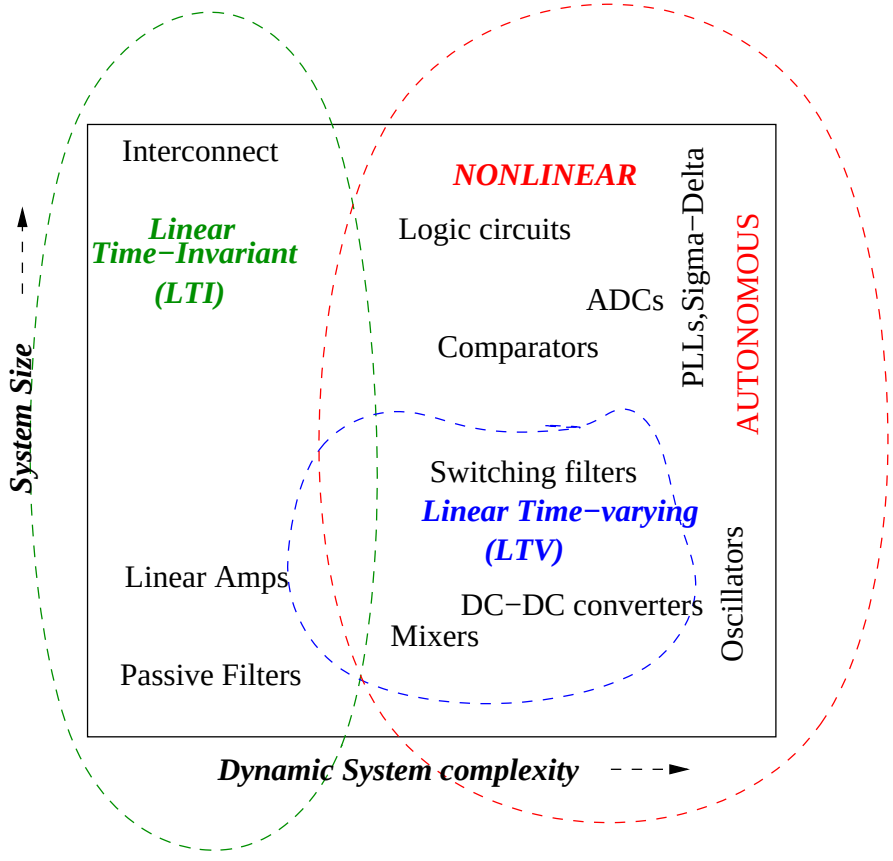


Figure: Complexity vs Size of physical systems

Reduced Order Modeling- Why & How?

- VLSI Systems today are more complex

  Issue Verification of such systems post-fabrication is challenging

  - size & complexity

  Approach Automated model generation for efficient verification

Model Order Reduction An idead in CAD

- for replacing the original large scale systems with much ***smaller*** one
- yet retains behavioral accuracy

Design Space exploration is much easier now as we need to simulate reduced samll system

Model Order reduction

- Indispensable tool for numerous areas- VLSI interconects, MEMS (micro-electro-mechanical systems)
- Nevertheless, the concept is general enough to be applicable to other areas
- Model Order Reduction is very intersting and meaningful mathematical problem in its own right
- Objectives
  - Model-based approach that propagates circuit performance characteristics in a bottom-up manner
  - Accuracy of models must be predictable and controllable
  - Model reduction implies automatic generation of the macro models by operation on the detailed descriptions
  - Reduction exploits information about internal structure of the system
  - More effective control on error is possible
  - Influence of complicated second-order physical effects included at system level

Linear Time Invariant (LTI) Systems

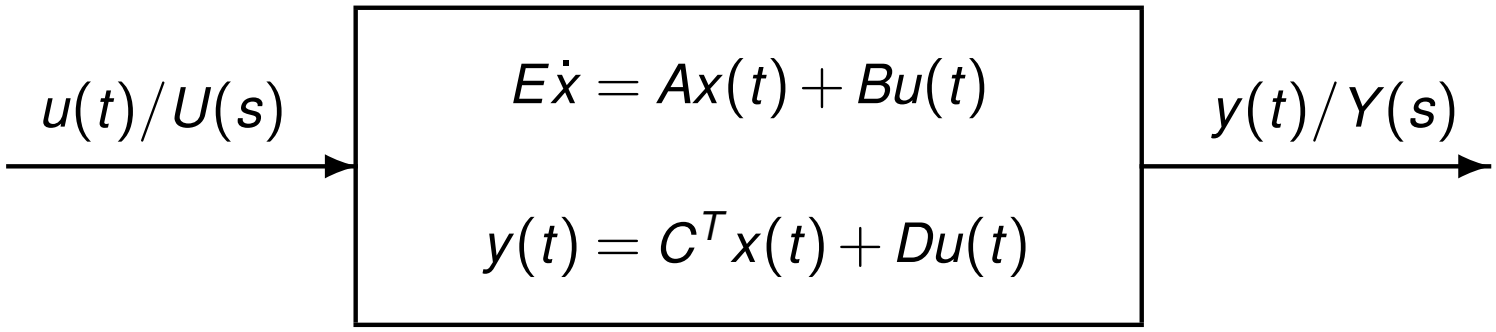


Figure: Linear Time Invariant block

Formulation of an LTI

Applying MNA to the RLC interconnect model

$$\begin{bmatrix} C & 0 \\ 0 & L \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{i} \end{bmatrix} = - \begin{bmatrix} G & B \\ -B^T & 0 \end{bmatrix} \begin{bmatrix} v \\ i \end{bmatrix} + \begin{bmatrix} i_s \\ 0 \end{bmatrix}$$

The equation can be simplified as

$$\begin{aligned} \mathcal{L}\dot{x} &= -\mathcal{R}x + e_j u \\ y &= e_k^T x \end{aligned}$$

$$\begin{aligned} A\dot{x} &= x + bu \\ y &= c^T x \end{aligned}$$

Formulation of LTI ...

The transfer function of the system is

$$Z_{jk}(s) = \frac{y(s)}{u(s)} = -c^T(I - sA)^{-1}b$$

Reduced order model that the algorithm is expected to generate

$$\begin{aligned} A_q\dot{x}_q &= x_q + b_q u \\ \tilde{y} &= c_q^T x_q \end{aligned}$$

Such that the reduced order transfer function

$$\tilde{Z}_{jk}(s) = \frac{\tilde{y}(s)}{u(s)} = -c_q^T(I - sA_q)^{-1}b_q$$

matches Z_{jk} with sufficient accuracy

s Domain Analysis

1. Moments
   - 1.1 In case of input being $\delta(t)$, the response at ouput port is the transfer function itself; since Laplace tramsform of impulse function is UNITY
   - 1.2 Moments of impulse response $h(t)$ are coefficients of powers of s in Maclaurin expansion of transfer function $H(s)$
   - 1.3
$$H(s) = \sum_{k=0}^{\infty} m_k s^k \tag{1}$$
   - 1.4
$$m_k = \frac{1}{k!} \times \left. \frac{\mathrm{d}^k H(s)}{\mathrm{d}s^k} \right|_{s=0} \tag{2}$$

Characteristic of Impulse Response

$$
\begin{aligned}
H(s) &= \int_0^\infty H(t)\, e^{-st} \mathrm{d}t && (3)\\
&= \int_0^\infty H(t) \left(1 - st + s^2 \frac{t^2}{2} + \cdots + s^k \frac{(-1)^k t^k}{k!} + \cdots \right) \mathrm{d}t && (4)\\
&= = \int_0^\infty \sum_{k=0}^{\infty} s^k \frac{(-1)^k t^k}{k!} H(t) \mathrm{d}t && (5)\\
&= \sum_{k=0}^{\infty} s^k \frac{(-1)^k}{k!} \int_0^\infty t^k H(t) \mathrm{d}t && (6)
\end{aligned}
$$

Moments

- Comparing 1 and 6,

$$m_k = \frac{(-1)^k}{k!} \int_0^\infty t^k H(t) \mathrm{d}t \tag{7}$$

$$m_0 = \int_0^\infty H(t) \mathrm{d}t \tag{8}$$

$$m_1 = - \int_0^\infty t H(t) \mathrm{d}t \tag{9}$$

Padé approximation

- Representing a function $f(x)$ as quotient of two polynomials, $R_N(x)$-

$$f(x) = R_N(x) = \frac{a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n}{1 + b_1 x + b_2 x^2 + \ldots + b_m x^m} \quad (10)$$

$$f(x) - R_N(x) = \left(c_0 + c_1 x + c_2 x^2 + \ldots + c_N x^N\right) - \frac{a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n}{1 + b_1 x + b_2 x^2 + \ldots + b_m x^m} \quad (11)$$

- The no. of constants in R_N is $n + m + 1$, and in $f(x)$ is $N + 1$;
- Also, here cooefficients c_i are $f^{(i)}(0)/(i!)$ *of Maclaurin's series*,
- The first N derivatives of $f(x)$ and $R_N(x)$ have to be equal at $x = 0$
- The coefficients of the powers of x up to and including x^N in numerator ust all be zero.

Padé approximation ...

- Generate coefficients of rational functions (ratio of polynomials) that are much more efficient;
  - If $f(x) = R_N(x)$at $x = 0$, the numerator must have no constatn term. Hence, $c_0 - a_0 = 0$.
  - For first N derivatives of $f(x)$ and $R_N(x)$to be equal at $x = 0$, the coefficients of powers of x up to and including x^Nin the numerator must all be zero also.

$$
\begin{aligned}
b_1 c_0 + c_1 - a_1 &= 0 \\
b_2 c_0 + b_1 c_1 + c_2 - a_2 &= 0 \\
b_3 c_0 + b_2 c_1 + b_1 c_2 + c_3 - a_3 &= 0 \\
&\vdots \\
b_m c_0 + b_{m-1} c_1 + b_{m-2} c_2 + \ldots c_m - a_m &= 0 \\
&\vdots \\
b_m c_{n-m} + b_{m-1} c_1 + b_{m-2} c_2 + \ldots + c_n - a_n &= 0 \\
b_m c_{n-m+1} + b_{m-1} c_{n-m+2} + b_{m-2} c_{n-m+3} + \ldots + c_{n+1} &= 0 \\
&\vdots \\
b_m c_{N-m} + b_{m-1} c_{N-m+1} + b_{m-2} c_{N-m+2} + \ldots + c_N &= 0
\end{aligned}
$$

Eigen Value computation

- Eigen Values & vector
- $\mathbf{Ax} = \lambda\mathbf{x} \qquad \Rightarrow \qquad (\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0}$
- Two matrices **A** and **B** are said to be similar, while **T** is non-singular
  - $\mathbf{A} = TBT^{-1} \qquad \Rightarrow \qquad \mathbf{B} = \mathbf{T}^{-1}\mathbf{AT}$
- Similarity matrices have identical Eigen values
  - replacing **A**: $\mathbf{TBT}^{-1}\mathbf{x} = \lambda\mathbf{x}$
  - $\mathbf{BT}^{-1}\mathbf{x} = \lambda\mathbf{T}^{-1}\mathbf{x}$
- Hence **A** and **B** have identical Eigen values, i.e. λ

QR Method for Eigen value computation

- Let a tri-diagonal matrix $A = QR$ be broken into two matrics, where Q is orthogonal, and R is upper triangular
- And, further B is formed with $B = RQ$
- Here, $R = Q^T A$, hence $B = Q^T AQ$ and is similar to A

The QR process is iterative, we obtain a sequence of matrices A_i, where as $i \longrightarrow \infty$ A_i tends to be a diagonal (block diagonal with max size 2×2)

Vector Space

- Let $\mathcal{F}$ be a field and V be an abelian addititve group such that there is a scalr multiplication of V by $\mathcal{F}$, which associates with each $s \in \mathcal{F}$ and $\xi \in V$ the element $s\xi \in V$. Then V is called a vector space over $\mathcal{F}$ provided, with u the unity of $\mathcal{F}$, following holds
  - $s(\xi+\eta) = s\xi + s\eta \qquad (s+t)\xi = s\xi + t\xi \qquad s(t\xi) = (st)\xi$
    $u\xi = \xi$
- Sub space-
  - A non empty U of a vector space V over $\mathcal{F}$ is a *subspace* of V provided U is itself a vector space over $\mathcal{F}$.

Vector Sub-space

Theorem A non-empty subset U of a vector space V over $\mathcal{F}$ is a subspace of V iff U is closed wrt scalar multiplication and vector addition as defined on V.

Theorem The set U of all linear combinations of an arbitrary set S of vectors ($2^{|S|}$) of a space V is a sub space of V.

- In turn vectors of S area called *generators* of the space U.
- Let $U = \{k_1\xi_1 + k_2\xi_2 + \cdots k_m\xi_m : k_i \in F\}$ be the space spanned by $S = \{\xi_1, \xi_2, \ldots, \xi_m\}$ a subset of vectors of V over $\mathcal{F}$
- It remains to find minimum set of vectors necessary to span a given space U
  - as any ξ_j if can be written as combination of other vectors of S, then ξ_j may be excluded from S, and remaining vectors will still span U.

Linear Dependence

- $\sum k_i \xi_i = k_1\xi_1 + k_2\xi_2 + \cdots k_m\xi_m = \zeta$
- A non-empty subset S of a vector space V over $\mathcal{F}$ is called *linearly dependent* over $\mathcal{F}$ iff there exists $k_1, k_2, \cdots k_m \in \mathcal{F} : \exists\, k_i \neq z$
- A non-empty subset S of a vector space V over $\mathcal{F}$ is called *linearly independent* over $\mathcal{F}$ iff there exists $k_1, k_2, \cdots k_m \in \mathcal{F} : \textit{every}\ k_i = z$

Theorem If some one of the set $S = \{\xi_1, \xi_2, \ldots, \xi_m\}$ of vectors in V over $\mathcal{F}$ is zero vecor ζ, then necessarily S is a linearly depdent set.

Theorem A set of non-zero vectors S of V over $\mathcal{F}$ is also *linearly dependent* iff some one of $\xi_j$can be expressed as linear combination of the vectors $\xi_1, \xi_2, \ldots, \xi_{j-1}$, which precedes it.

Theorem Any finite set S of vectors, not all the zero vector, contains a linearly indepdendent subset U which spans the same vector space as S.

Bases of a Vector Space

- A set $S = \{\xi_1, \xi_2, \ldots, \xi_m\}$ of vectors of a vector space V over $\mathcal{F}$ is called a basis of V provided
  1. S is linearly independent set,
  2. the vectors of S span V
- Let's define unit vectors of $V_n(\mathcal{F})$

$$\begin{aligned} \varepsilon_1 &= (u, 0, 0, 0, \ldots, 0, 0) \\ \varepsilon_2 &= (0, u, 0, 0, \ldots, 0, 0) \\ \vdots &\quad \vdots \quad \vdots \\ \varepsilon_n &= (0, 0, 0, 0, \ldots, 0, u) \end{aligned}$$

- and consider linear combination,
  $\xi = a_1\varepsilon_1 + a_2\varepsilon_2 + \cdots a_n\varepsilon_n = (a_1, a_2, \ldots, a_n) \quad a_i \in \mathcal{F}$
- If $\xi = \zeta$, then $a_1, = a = \ldots = a_n = z$; and hence
  $E = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)$ is a linearly independent set.

Bases of a Vector Space

Theorem If $S = \{\xi_1, \xi_2, \ldots, \xi_m\}$ is a basis of the vector space V over $\mathcal{F}$ and $T = \{\eta_1, \eta_2, \ldots, \eta_n\}$ is any linearly independent set of vectors of V, then $n \leqslant m$.

Theorem As a consequence, if If $S = \{\xi_1, \xi_2, \ldots, \xi_m\}$ is a basis of the vector space V over $\mathcal{F}$, then any $m+1$ vectors of V necessarily form a linearly dependent set.

Theorem Every basis of a vector space V over $\mathcal{F}$ has the same number of elements. This number is called *dimension* of V.

Sub-spaces of a vector

- Let V, of dimension n, be a vector space over $\mathcal{F}$ and U, of dimension $n < m$ having $B = \{\xi_1, \xi_2, \ldots, \xi_m\}$ as basis, be a sub-space of V. Then, only m of the unit vectors of V can be written as linear combination of elements of B; hence there exist vectors of V which are not in U.
  - $k_1\xi_1 + k_2\xi_2 + \cdots + k_m\xi_m + k\eta_1 = \zeta \qquad \forall k_i, k \in F$
  - now $k = z$ since otherwise $k^{-1} \in F$, and $\eta_1 = k^{-1}(-k_1\xi_1 - k_2\xi_2 - \cdots - k_m\xi_m)$, and $\eta_1 \in U$, which is contrary to definition of η_1 , hence PROVED.

Theorem If $B = \{\xi_1, \xi_2, \ldots, \xi_m\}$ is basis of $U \subset V$, V having dimension n, there exist vectors $\eta_1, \eta_2, \ldots, \eta_{n-m}$ in V such that $B \cup \{\eta_1, \eta_2, \ldots, \eta_{n-m}\}$ is basis of V.

Theorem If, in $V_n(R)$, a vector η is orthogonal to each vector of the set $\{\xi_1, \xi_2, \ldots, \xi_m\}$, then η is orthogonal to every vector of the space spanned by this set.

Vector Spaces over R

- Let's focus on to vector space $V = V_n(R)$ over R.
  - for 2-dimensional vetors, $\xi = (a_1, a_{2)}$ and $\eta = (b_1, b_2) \cos\theta = \frac{a_1 b_1 + a_2 b_2}{|\xi| \cdot |\eta|}$
  - Hence, inner product is defined as, $\xi \cdot \eta = a_1 b_1 + a_2 b_2$
  - For n-dimensional $V_n(R)$, for all $\xi = (a_1, a_2, \ldots, a_n)$ and $\eta = (b_1, b_2, \ldots, b_n)$
    - $\xi \cdot \eta = \sum a_i b_i$
- Suppose in $V_n(R)$, a vector η is orthogonal to each vector of the set $\{\xi_1, \xi_2, \ldots, \xi_m\}$, then η is orthogonal to every vector of the space spanned by this set.

Orthogonal & Orthonormal Bases

- Two vectors u and v are orthogonal, if their inner product $\langle u, v \rangle = 0$
- The set of vectors $\{u_i\}$in V are orthonormal, if they are orthogonal among themselves $\langle u_i, u_j \rangle = 0$ for any $i \neq j$, and each of u_i has length 1.
- When $\{u_i\}$ is a basis of V, then it is orthonormal basis of V

Gram-Schmidt Process (Orthogonalization)

- In an N-dimensional space, there can be no more than N vectors that are independent.
- Alternatively, it is always possible to find a set of N vectors that are independent.
- Once this set is chosen, any vector in this space can be represented as the linear combination of the vectors in this set [**?**].
- This set of N independent vectors is by no means unique.
- An orthogonal set of vectors however becomes of great interest when we are representing any system because it is easier to deal with as compared to a non-orthogonal set.
- Finding an orthogonal set of vectors from any given set of vectors is what is done by the Gram-Schmidt process.

Gram-Schmidt Process ...

The projection of a vector x_2 upon another vector x_1 is $c_{12}x_1$, where

$$c_{12} = \frac{x_1 . x_2}{|x_1|^2} x_1 \tag{12}$$

The error in this approximation is the vector $x_2 - c_{12}x_1$, that is

$$Error \;\; vector = x_2 - \frac{x_1 . x_2}{|x_1|^2} x_1 \tag{13}$$

The error vector is orthogonal to vector x_1.

Gram-Schmidt Process (2D)

In order to get insight in the process we consider a simple case of 2-dimnesional space. Let x_1 and x_2 be two independent vectors 2-dimensional space. If we want to generate a new set of two orthogonal vectors y_1 and y_2 from x_1 and x_2. For convenience, we choose,

$$y_1 = x_1 \tag{14}$$

For finding another vector y_2 that is orthogonal to y_1 (and x_1). Since the error vector is orthogonal to y_1 and can be taken as an approximation for y_2 i.e.

$$\begin{aligned} y_2 &= x_2 - \frac{x_1 . x_2}{|x_1|^2} x_1 \\ &= x_2 - \frac{y_1 . x_2}{|y_1|^2} y_1 \end{aligned} \tag{15}$$

Equations (14) and (15) yield the desired orthogonal set. This set is not unique and an infinite number of orthogonal vector sets (y_1, y_2) can be generated from (x_1, x_2).

Gram-Schmidt Process (N-Dimensional)

This result can be extended to an N-dimensional space. In general, if we are given N independent vectors $x_1, x_2, \ldots, x_N$, the by proceeding along similar lines, we can obtain an orthogonal set $y_1, y_2, \ldots, y_N$, where

$$y_1 = x_1$$

and

$$y_j = x_j - \sum_{k=1}^{j-1} \frac{y_k . x_j}{|y_k|^2} y_k \qquad j = 2, 3, \ldots, N \tag{16}$$

This set generated is not orthonormal. To generate an orthonormal set $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N$ can be obtained by normalizing the lengths of the respective vectors,

$$\hat{y}_k = \frac{y_k}{|y_k|}$$

Pade Approximation, for system $H(s)$

$$H_{p,q} = \frac{P(s)}{Q(s)} = \frac{a_0 + a_1 + \cdots + a_p s^p}{1 + b_1 + \cdots + b_q s^q} \tag{17}$$

$$H(s) = \sum_{k=0}^{\infty} m_k s^k \text{ limited to first (p+q+1) terms} \tag{18}$$

$$= m_0 + m_1 s + m_2 s^2 + \cdots + m_{p+q} s^{p+q} + r(s) s^{p+q+1} \tag{19}$$

$$= H_{p,q}(s) + O(s^{p+q+1}) \tag{20}$$

$$\begin{aligned} a_0 + a_1 + \cdots + a_p s^p &= (1 + b_1 + \cdots + b_q s^q) \\ &\times \left(m_0 + m_1 s + m_2 s^2 + \cdots + m_{p+q} s^{p+q} + r(s) s^{p+q+1}\right) \end{aligned}$$

Deriving Moments from MNA Formulation

$$M\dot{\mathbf{X}}(t) = -G\mathbf{X}(t) + P\mathbf{U}(t) \quad (21)$$

$$\mathbf{Y}(t) = Q\mathbf{X}(t) \quad (22)$$

Assuming that $\mathbf{X}(0) = 0$, taking Laplace transform of above,

$$\begin{aligned} sM\mathbf{X}(s) &= -G\mathbf{X}(s) + P\mathbf{U}(s) \\ \mathbf{Y}(s) &= Q\mathbf{X}(s) \end{aligned}$$

$$\begin{aligned} \mathbf{X}(s) &= (G + sM)^{-1} P\mathbf{U}(s) \\ \mathbf{Y}(s) &= Q\mathbf{X}(s) \\ &= Q(G + sM)^{-1} P\mathbf{U}(s) \\ \Rightarrow H(s) &= Q(G + sM)^{-1} P \end{aligned}$$

Deriving Moments from MNA Formulation ...

$$\Rightarrow H(s) \quad = \quad Q(G+sM)^{-1}P$$

- Here, coefficients of Maclaurin expansion of $H(s)$ are given by,

$$M_j \quad = \quad (-1)^j Q\left(G^{-1}M\right)^j G^{-1}P$$

- Computation of moments requires G to be invertible. This requirement is easily satisfied by most interconnect circuits in which each node has a DC path to the ground.

Deriving Moments from for RLC circuits

- Lets consider all entries in unknown vector $\mathbf{X}(s)$ as outputs, i.e. Q is an identity matrix,

$$
\begin{aligned}
\mathbf{M}_0 &= G^{-1}PU \\
\mathbf{M}_1 &= G^{-1}MG^{-1}P = G^{-1}M\mathbf{M}_0 \\
\mathbf{M}_2 &= \left(G^{-1}M\right)^2 G^{-1}M = G^{-1}M\mathbf{M}_1 \\
&\dots
\end{aligned}
$$

$$
\begin{aligned}
G\mathbf{M}_0 &= PU \\
G\mathbf{M}_1 &= M\mathbf{M}_0 \\
G\mathbf{M}_2 &= M\mathbf{M}_1 \\
&\dots \\
G\mathbf{M}_{i+1} &= M\mathbf{M}_i
\end{aligned}
$$

Deriving Moments from for RLC circuits ...

- Moments are to be evaluated iteratively,
- Matrix G, is admittance matrix of resistive tree derived from original RLC tree, by removing all C and L. Solving $G\mathbf{M}_0 = PU$ implies obtaining DC solution, which is very straight-forward.
- Next, supposing $\mathbf{M}_i$ is given, lets compute $\mathbf{M}_{i+1}$
  - $G\mathbf{M}_{i+1} = M\mathbf{M}_i$
  - $\Rightarrow$ tree remains same (as G if on LHS)
  - Inputs are changed to $M\mathbf{M}_i$, M is interpretable BUT $\mathbf{M}_i$ needs further interpretation

Deriving Moments from for RLC circuits ...

- $M \equiv \begin{bmatrix} C & 0 \\ 0 & L \end{bmatrix}$, as C imples KCL (currents through capacitors) & L KVL (voltages across inductors)
- We partition $M\mathbf{M}_i$ according to the composition of M
  - $M\mathbf{M}_i = \begin{bmatrix} C & 0 \\ 0 & L \end{bmatrix} \begin{bmatrix} M_{iV} \\ M_{il} \end{bmatrix} \equiv \begin{bmatrix} I_C \\ V_L \end{bmatrix}$
- Entries in I_C are related to product of capacitance and the i^{th} moment of voltage (M_{iV})
- Entries in V_L are realted to product of inductance and the i^{th} moment of voltage (M_{il})
- We generate NEW tree from OLD tree
  - zeroing out oldl sources
  - adding CURRENT sources and VOLTAGE sources at location of capacitors and inductors of original tree
- The solution is now trivial

Asymptotic Waveform Evaluation

In their seminal 1990 paper [**?**], Pileggi and Rohrer used moments of the transfer function as fidelity metrics, to be preserved by the model reduction process. The moments m_i of an LTI transfer function $H(s)$ are related to its derivatives, i.e.,

$$m_1 = \left.\frac{dH(s)}{ds}\right|_{s=s_0} \qquad m_2 = \left.\frac{d^2H(s)}{ds^2}\right|_{s=s_0} \qquad ,\ldots, \tag{23}$$

AWE first computes a number of moments of the full system, and then uses these in another set of linear equations, the solution of which results in the reduced model. Such a procedure is termed explicit moment matching.

AWE Approximation

AWE is most conveniently explained in general in terms of the differential state equations for a lumped, linear, time-invariant circuit:

$$\dot{x} = Ax + Bu \tag{24}$$

where x is the n-dimensional state vector and u is the m-dimensional excitation vector. Suppose that the particular excitation is of the form

$$u_p(t) = u_0 + u_1 t \tag{25}$$

where u_0 and u_1 are constant m-dimensional vectors. In general the form of $u_p(t)$ need not be confined to such simple signals, but rather could assume any form of input excitation for which a particular solution can easily be obtained. Inputs that are polynomials in time or sums of complex-valued exponentials can in theory be as easily accommodated as the step/ramp combination in expression.
For the excitation u_p,(25), the differential-state equations(24) has the particular solution

$$x_p(t) = -A^{-1}Bu_0 - A^{-2}Bu_1 - A^{-1}Bu_1 t \tag{26}$$

AWE ...

$$\dot{x}_h = Ax_h \tag{27}$$

Now with the initial condition

$$x_h(0) = x_0 + A^{-1}Bu_0 + A^{-2}Bu_1 \tag{28}$$

where x_0 is the initial state at time zero. The Laplace transform solution of the homogeneous equation is

$$X_h(s) = (sI - A)^{-1}x_h(0) \tag{29}$$

To approximate this solution, $X_h(s)$ is first expanded in a Maclaurin series

$$X_h(s) = -A^{-1}(I + A^{-1}s + A^{-2}s^2 + \cdots)x_h(0) \tag{30}$$

and as many moments as necessary or desirable are matched in terms of lower order approximating functions. The justification for such a moment matching approach follows from the Laplace transforms definition [**?**]. The time moments are given by:

$$X(s) = \int_0^\infty e^{-st}x(t)\ dt = \sum_{k=0}^{\infty} \frac{1}{k!}(-s)^k \int_0^\infty t^k x(t)\ dt \tag{31}$$

AWE ...

The time moments are given by the power of t:

$$m_k = \frac{-1^k}{k!} \int_0^\infty t^k x(t) \; dt \tag{32}$$

depending upon the output needed we consider a specific component of $X_h(s)$,say the ith, its initial conditions and first $2q-1$ moments that are characterized as:

$$\begin{aligned}
[m_{-1}]_i &= [x_h(0)]_i \\
[m_0]_i &= \left[-A^{-1}x_h(0)\right]_i \\
[m_1]_i &= \left[-A^{-2}x_h(0)\right]_i \\
&\vdots \quad\quad \vdots \\
[m_{2q-2}]_i &= \left[-A^{-2q+1}x_h(0)\right]_i
\end{aligned} \tag{33}$$

The rest of the process follows similar method as for MNA formulation of RLC circuits.

AWE ...

Now these moments are matched to a lower order frequency-domain function of the form:

$$\begin{aligned}\hat{X}_i(s) &= \frac{k_1}{s-p_1} + \frac{k_2}{s-p_2} + \cdots + \frac{k_q}{s-p_q} \\ &= \sum_{l=1}^{q} \frac{k_l}{s-p_l} = -\sum_{l=1}^{q} \frac{k_l/p_l}{1-s/p_l}\end{aligned} \tag{34}$$

where p_1 through p_q are the complex approximating poles and k_l through k_q their appropriate residues. In other words, the time-domain moments are to be matched to those of an approximating function of the form

$$\hat{x}_i(t) = \sum_{l=1}^{q} k_l e^{p_l t} \tag{35}$$

AWE ...

Expanding each of the terms in (34) into a series about the origin, and upon inclusion of the initial conditions, the following set of nonlinear simultaneous equations for the ith state variable is obtained:

$$\begin{aligned}
-(k_1 + k_2 + \cdots + k_q) &= [m_{-1}]_i \\
-\left(\frac{k_1}{p_1} + \frac{k_2}{p_2} + \cdots + \frac{k_q}{p_q}\right) &= [m_0]_i \\
-\left(\frac{k_1}{p_1^2} + \frac{k_2}{p_2^2} + \cdots + \frac{k_q}{p_q^2}\right) &= [m_1]_i \\
\vdots &\quad \vdots \\
-\left(\frac{k_1}{p_1^{2q-1}} + \frac{k_2}{p_2^{2q-1}} + \cdots + \frac{k_q}{p_q^{2q-1}}\right) &= [m_{2q-2}]_i
\end{aligned} \tag{36}$$

A solution for the approximating poles and residues from this set of nonlinear equations could proceed in terms of Newton-Raphson or a similar iteration method.

AWE ...

we reformulate the problem to allow for direct solution of the approximating poles and residues. The set of equations in (36) can be summarized in matrix form as

$$-\mathcal{V}k = [m_l]_i \tag{37}$$

and

$$\mathcal{V}\Lambda^{-q}k = [m_h]_i \tag{38}$$

where m_l, represents the low-order moments $(-1, 0, \cdots, q-2)$. m_h represents the high-order moments $(q-1, q, \cdots, 2q-2)$. Λ^{-1} is a diagonal matrix of the reciprocal complex poles, and $\mathcal{V}$ is the well-known Vandermonde matrix:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ p_1^{-1} & p_2^{-1} & \cdots & p_q^{-1} \\ p_1^{-2} & p_2^{-2} & \cdots & p_q^{-2} \\ \vdots & \vdots & & \vdots \\ p_1^{-q+1} & p_2^{-q+1} & \cdots & p_q^{-q+1} \end{bmatrix} \tag{39}$$

AWE . . .

It follows then from (37) that

$$k = -\mathcal{V}^{-1} m_l \tag{40}$$

$$\mathcal{V}\Lambda^{-q}\mathcal{V}^{-1} m_l = m_h \tag{41}$$

Since the Vandermonde matrix is the modal matrix for a system matrix in companion form, (41) is equivalent to:

$$A_c^{-q} m_l = m_h \tag{42}$$

where

$$A_c^{-1} = \begin{bmatrix} 0 & 1 & 0 & . & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{q-1} \end{bmatrix} \tag{43}$$

with the coefficients normalized so that $a_q = 1$. This matrix is characterized as A_c^{-1} rather than A_c, because its eigenvalues are the reciprocals of the approximating poles for the original system (24).

AWE ...

The set of simultaneous nonlinear equations (42) for the coefficients a_0, through a_{q-1}, a_c can be written recursively to yield the following set of linear equations:

$$\begin{bmatrix} m_{-1} & m_0 & \cdots & m_{q-2} \\ m_0 & m_1 & \cdots & m_{q-1} \\ \vdots & \vdots & & \vdots \\ m_{q-2} & m_{q-1} & \cdots & m_{2q-3} \end{bmatrix} \begin{bmatrix} -a_0 \\ -a_1 \\ \vdots \\ -a_{q-1} \end{bmatrix} = \begin{bmatrix} m_{q-1} \\ m_q \\ \vdots \\ m_{2q-2} \end{bmatrix} \tag{44}$$

It is in terms of a_c, we can form a characteristic polynomial

$$a_0 + a_1 p^{-1} + a_2 p^{-2} + \cdots + a_{q-1} p^{-q+1} + p^{-q} = 0 \tag{45}$$

Krylov Subspace Approach

Computing the kth moment explicitly involves evaluating terms of the form $A^{-k}r$,i.e., the kth member of the Krylov subspace of A and r. If A has well separated eigenvalues (as it typically does for circuit matrices), then for $k \simeq 10$ and above, only the dominant eigenvalue contributes to these terms, with non-dominant ones receding into numerical insignificance.
Krylov-subspace methods are best viewed as reducing the system via projection. They produce two projection matrices, $V \in R^{n\times q}$ and $W^T \in R^{q\times n}$, such that the reduced system is obtained as

$$\begin{aligned} \underbrace{W^T E}_{\hat{E}}\dot{x} &= \underbrace{W^T AV}_{\hat{A}}x(t) + \underbrace{W^T B}_{\hat{B}}u(t) \\ y(t) &= \underbrace{C^T V}_{\hat{C}^T}x(t) + Du(t) \end{aligned} \tag{46}$$

For the reduction to be practically meaningful, q, the size of the reduced system, must be much smaller than n, the size of the original. If the Lanczos process is used, then $W^T V \approx I$(i.e., the two projection bases are bi-orthogonal). If the Arnoldi process is applied, then $W = V$ and $W^T V = I$.

Padé via the Lanczos (PVL) Algorithm

- Limitations of AWE
  - AWE computes with relatively a small no. of poles
  - Less accurate reduced order model H(s) improves only up to a small value of q
- Improvements in PVL
  - Arbitrary number of poles and zeros are used
  - Computational cost similar to AWE

PVL . . .

Definition 1**-Krylov Subspace**: The Krylov subspace $\mathcal{K}_m(A,p)$ generated by a matrix A and vector p, of order m, is the space spanned by the set of vectors $\left\{p, Ap, A^2p, \ldots, A^{m-1}p\right\}$.

- A basis for a Krylov subspace can be quickly computed if A can be rapidly applied to p, e.g. due to sparsity. This fact gives Krylov-based model reduction the potential for cost and savings. Extremely simple [**?**] but not particularly effective choices for V and W are

$$\begin{aligned} colsp\, V(s) &= \mathcal{K}_M((A-sE),B) \\ &= span\left\{B,(A-sE)B,\ldots,(A-sE)^{M-1}B\right\} \qquad (47) \\ colsp\, W(s) &= \mathcal{K}_M((A-sE)^T,C) \\ &= span\left\{C,(A-sE)^TC,\ldots,((A-sE)^T)^{M-1}C\right\} \qquad (48) \end{aligned}$$

- The essential elements of Krylov-subspace-based reduction are given by Suppose $K_m(A^{-1},p) \subset \text{colspan}(V)$, then $V(V^TAV)^{-k}V^TB = V\hat{A}^{-k}b = A^{-k}b$, for $k < m$.

PVL . . .

The algorithm of Lanczos computes rectangular matrices V and $W \in R^{N\times M}$ that restrict a specified matrix G to a tridiagonal form,

$$S = W^T G V = \begin{bmatrix} \alpha_1 & \beta_2 & 0 & \cdots \\ \gamma_2 & \alpha_2 & \ddots & \ddots \\ 0 & \ddots & \ddots & \beta_M \\ 0 & \cdots & \gamma_M & \alpha_M \end{bmatrix}$$

and that satisfy

$$colsp\{V\} \in \mathcal{K}_M(G, \hat{v}_1) \quad and \quad colsp\{W\} \in \mathcal{K}_M(G^T, \hat{w}_1) \qquad (49)$$

The vectors $\hat{v}_1$ and $\hat{w}_1$ are user-specified starting vectors which lie in the direction of the first column of V and W. Alternatively and equivalently, the Lanczos method can be viewed as an approach for constructing biorthogonal V and W, i.e., $W^T V = I$, that satisfy the same Krylov subspace conditions (49).

PVL: Basic algorithm layout

- System of equations & transfer function, defined by

$$C\dot{x} = -Gx + bu \qquad y = l^T x$$

$$H(s) = l^T (G + sC)^{-1} b$$

- Using s_0 as an arbitrary expansion point the modified transfer function is

$$H(s_0 + \sigma) = l^T (I - sA)^{-1} r$$

PVL . . .

- The reduced order transfer generated by PVL:
  - The Taylor coefficients necessary for the Padé approximant H_q result from the following expansion of $H(s)$ about s_0:

$$H(s_0+\sigma) = l^T(I+\sigma A+\sigma^2 A^2+\cdots)r = \sum_{k=0}^{\infty} m_k \sigma^k,$$

where

$$m_k = l^T A^k r, \qquad k = 0,1,\ldots \tag{50}$$

$$H_q(s_0+\sigma) = \sum_{j=1}^{q} \frac{l^T r.\mu_j \nu_j}{1-\sigma\lambda_j}$$

PVL: Basic algorithm layout . . .

Run q steps of the Lanczos Algorithm to obtain the tridiagonal matrix T_q

- Compute $\rho_n = ||v||_2$ and $\eta_n = ||w||_2$.
  If $\rho_n = 0$ or $\eta_n = 0$, then stop.
- Set

$$v_n = \frac{v}{\rho_n}, \quad w_n = \frac{w}{\eta_n},$$

$$\delta_n = w_n^T v_n, \quad \alpha_n = \frac{w_n^T A v_n}{\delta_n},$$

$$\beta_n = \eta_n \frac{\delta_n}{\delta_{n-1}}, \quad \gamma_n = \rho_n \frac{\delta_n}{\delta_{n-1}},$$

$$\begin{aligned} v &= A v_n - v_n \alpha_n - v_{n-1} \beta_n \\ w &= A^T w_n - w_n \alpha_n - w_{n-1} \gamma_n \end{aligned}$$

PVL: Basic algorithm layout . . .

Matrix T_q is given by

$$T_q = \begin{bmatrix} \alpha_1 & \beta_2 & \cdots & 0 \\ \rho_2 & \alpha_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \beta_q \\ 0 & \cdots & \rho_q & \alpha_q \end{bmatrix}$$

Compute the eigen decomposition of the matrix T_q and set:

$$T_q = S_q \, diag(\lambda_1, \lambda_2, \ldots, \lambda_q) S_q^{-1}$$

$\mu = S_q^T e_1$ and $\nu = S_q^{-1} e_1$.

Compute poles and residues by setting:

$$p_j = 1/\lambda_j \text{ and } k_j = \frac{I^T r . \mu_j \nu_j}{\lambda_j} \text{ for all } j = 1, 2, \ldots, q$$

Arnoldi Approach

Arnoldi algorithm is a better conditioned process that direct evaluation of the moments because it generates an orthogonal set of vectors which span $A^k b,\ for\ k = 0, 1, \ldots, 2q-1$.
Restating the system of equations for the linear system used in the Arnoldi method as input can be written as:

$$\dot{x} = Ax + bu(t) \tag{51}$$

$$y = c^T x \tag{52}$$

The Arnoldi method is based on the following “Arnoldi algorithm" which has 3 inputs:
A, an N by N matrix; $b \in R^N$; and an integer q, which will be the size chosen for the reduced system and usually $q \ll N$
and will generate 3 outputs:
V, an N by q matrix; H, an q by q matrix; and $v_{q+1} \in R^q$ which will satisfy the following 3 properties:

Arnoldi . . .

Property

V is an column-orthonormal matrix; its q columns form a set of orthonormal vectors in R^N;

$$AV = VH + hv_{q+1}e_q^T \tag{53}$$

where h is a scalar and the e_q is the qth standard unit vector in R^q

$$A^k b = ||b|| VH^k e_1 \tag{54}$$

where $k = 0, \cdots, q-1$ and $||.||$ is the 2-norm.
The Arnoldi algorithm [**?**] is actually a *modified* Gram-Schmidt process for the *Krylov subspace* $\mathcal{K}(A,b) = span\{b, Ab, A^2b, \cdots, A^{q-1}b\}$ with the orthogonalized q vectors $v_1, \cdots, v_q$ constitute of the q columns of the matrix V as described below: First form unit vector $v_1 = b/||b||$

Arnoldi . . .

- Trade off between optimality to gain guaranteed stability
- It is a modified Gram-Schmidt process
- Arnoldi process creates two matrices V_q and H_q satisfying

$$AV = VH + h_{q+1,q} v_{q+1} e_q^T$$

- For the moment $A^k b$ of the system to match

$$\begin{aligned} A^k b &= ||b|| A^k V_q e_1 = ||b|| V_q H_q^k e_1 \\ m_k &= c^T A^k b = \underbrace{||b|| c^T V_q}_{c_q^T} \underbrace{H_q^k}_{A_q^k} \underbrace{e_1}_{b_q} \end{aligned}$$

- Requiring the state space realization of system to be A

$$A_q = H_q \;\; b_q = e_1 \;\; c_q = \|b\| \, V_q^T c$$

- we get

$$H\dot{z} = z + V^T b u(t) \tag{55}$$

$$y = c^T V z \tag{56}$$

Arnoldi Approach: Basic algorithm layout ...

$$
\begin{aligned}
&v_1 = b/||b|| \\
&for \;\; j = 1 : q \\
&\quad \{w = Av_j \\
&\quad for \;\; i = 1 : j \\
&\qquad \{h_{i,j} = w^T v_i \\
&\qquad w = w - h_{i,j} v_i\} \\
&\quad h_{j+1,j} = ||w|| \\
&\quad if(h_{j+1,j} \neq 0) \\
&\qquad v_{j+1} = w/h_{j+1,j}\}
\end{aligned}
$$

Jacobi-Davidson Method

The time domain formulation of any system is given by as follows:

$$\begin{cases} C\frac{dx(t)}{dt} + Gx(t) = u(t), \\ x(0) = 0 \end{cases} \tag{57}$$

where $x \in R^n$ contains the circuit state, $C \in R^{n\times n}$ is the capacitance matrix, $G \in R^{n\times n}$ is the conductance matrix and $u(t)$ models the excitation.

Jacobi-Davidson Method ...

Because all the properties cannot be computed in time-domain, the problem is formulated in frequency-domain by applying the Laplace transforms:

$$(sC+G)\mathcal{X}(s) = \mathcal{U}(s), \tag{58}$$

where $\mathcal{X}, \mathcal{U}$ are the Laplace-transform of the variable x, u and s is the variable in the frequency domain. The transfer function of the circuit is given by:

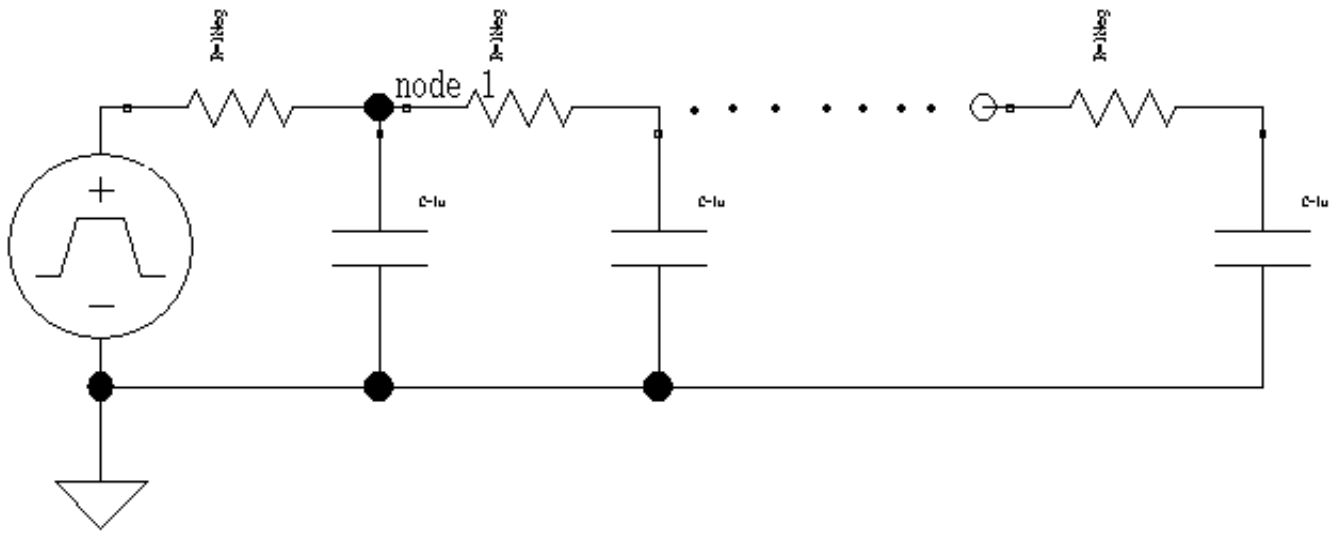
$$\mathcal{H}(s) = (sC+G)^{-1} \tag{59}$$

The poles are the values $p_k \in C$ that satisfy $\det(p_k C+G) = 0$, hence $(G+p_k C)x = 0$, for some $x \neq 0$, which leads to the eigenvalue problem ($\lambda = -p_k$):

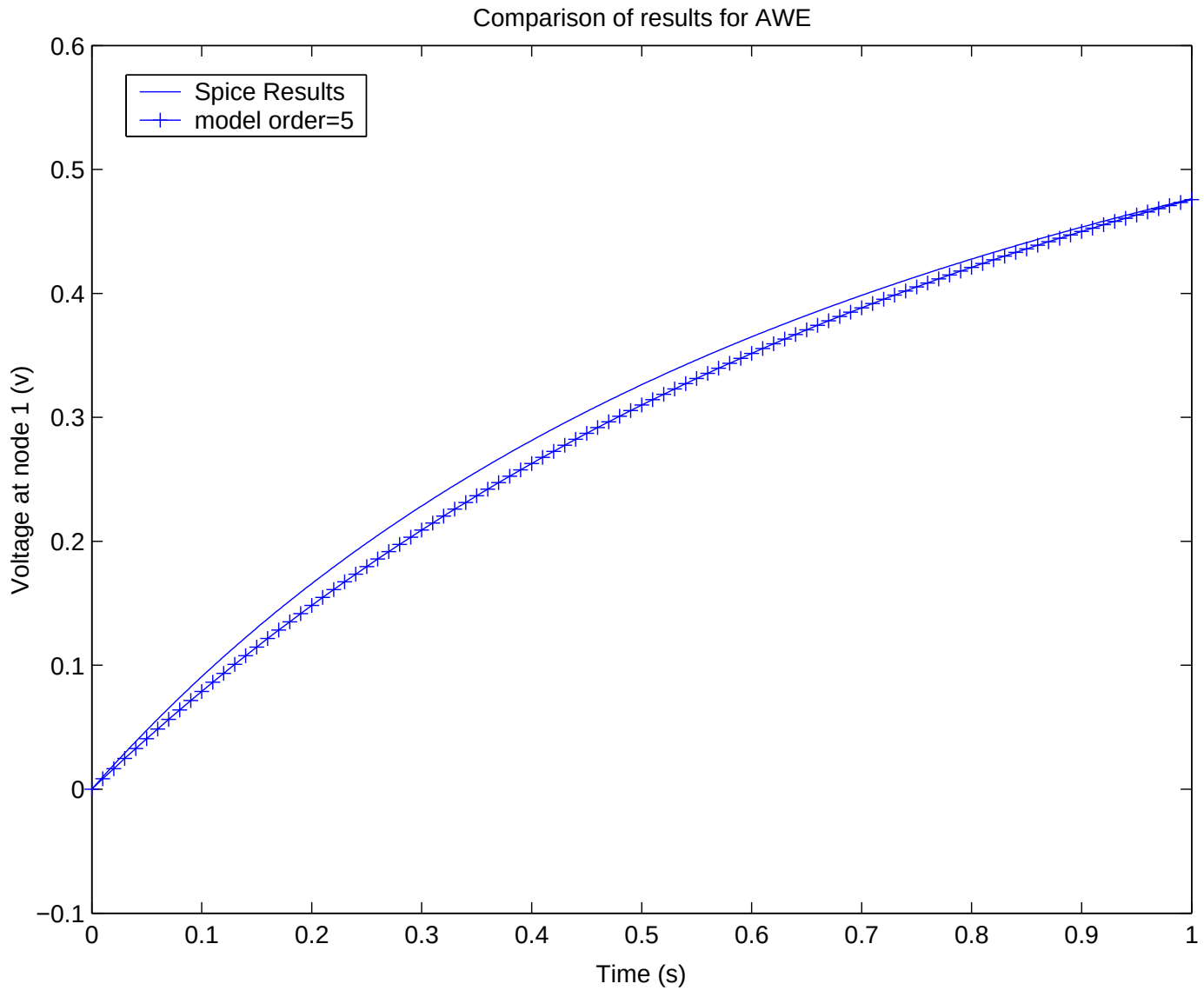
$$Gx = \lambda Cx, \qquad x \neq 0 \tag{60}$$

The problem of computing zeros is similar to that of computing poles. Especially for large circuits ($n > 10^4$), robust, iterative methods for the generalized eigenvalue problem with sufficient accuracy and acceptable computational cost are needed. The Jacobi-Davidson method is one of the approaches which like Arnoldi method does separate pole-zero computation while the approaches like AWE and PVL do combined pole-zero computation.

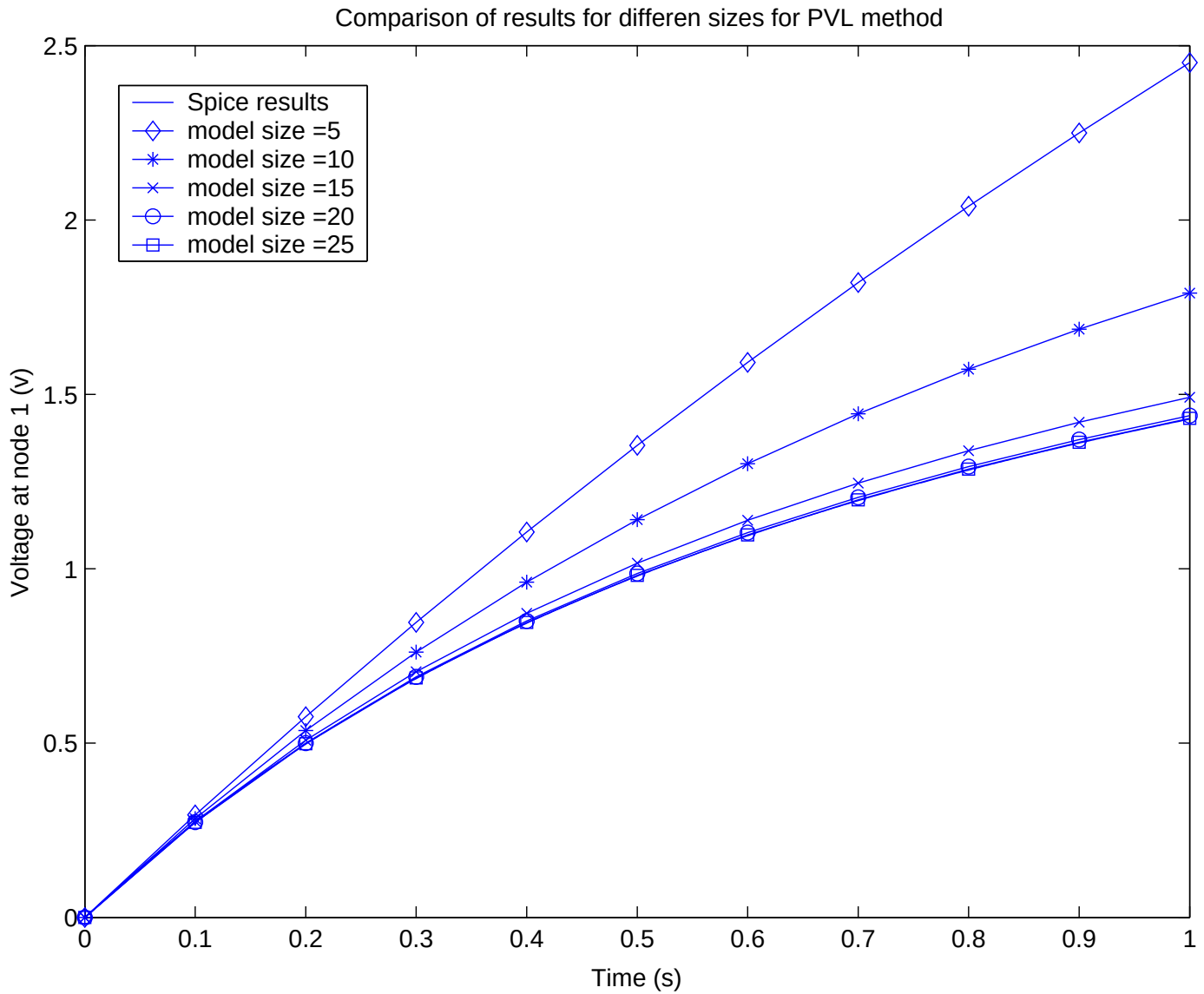
Application of algorithms for linear circuits



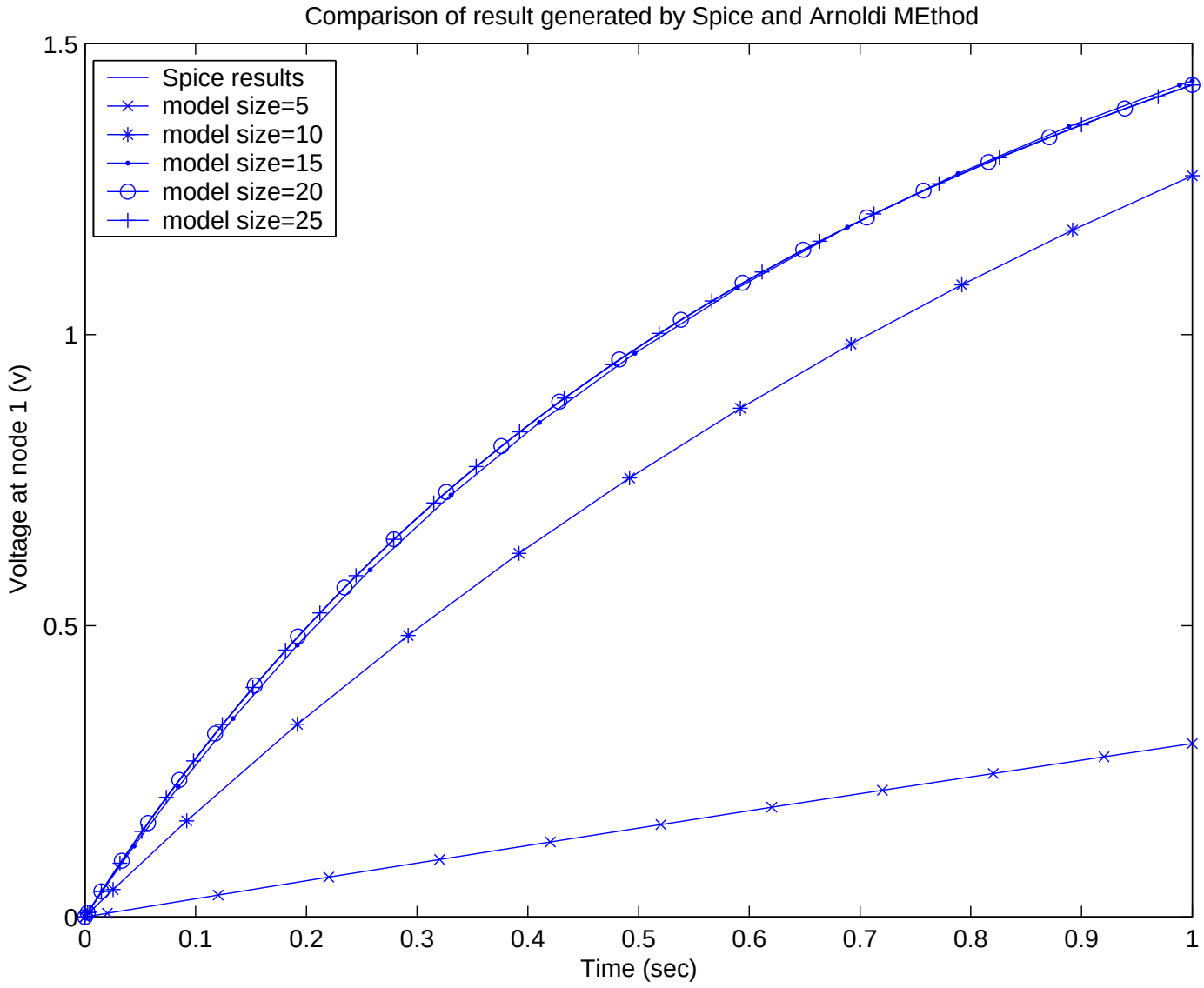
Results: AWE with reduced order =5 for a line with 25 elements



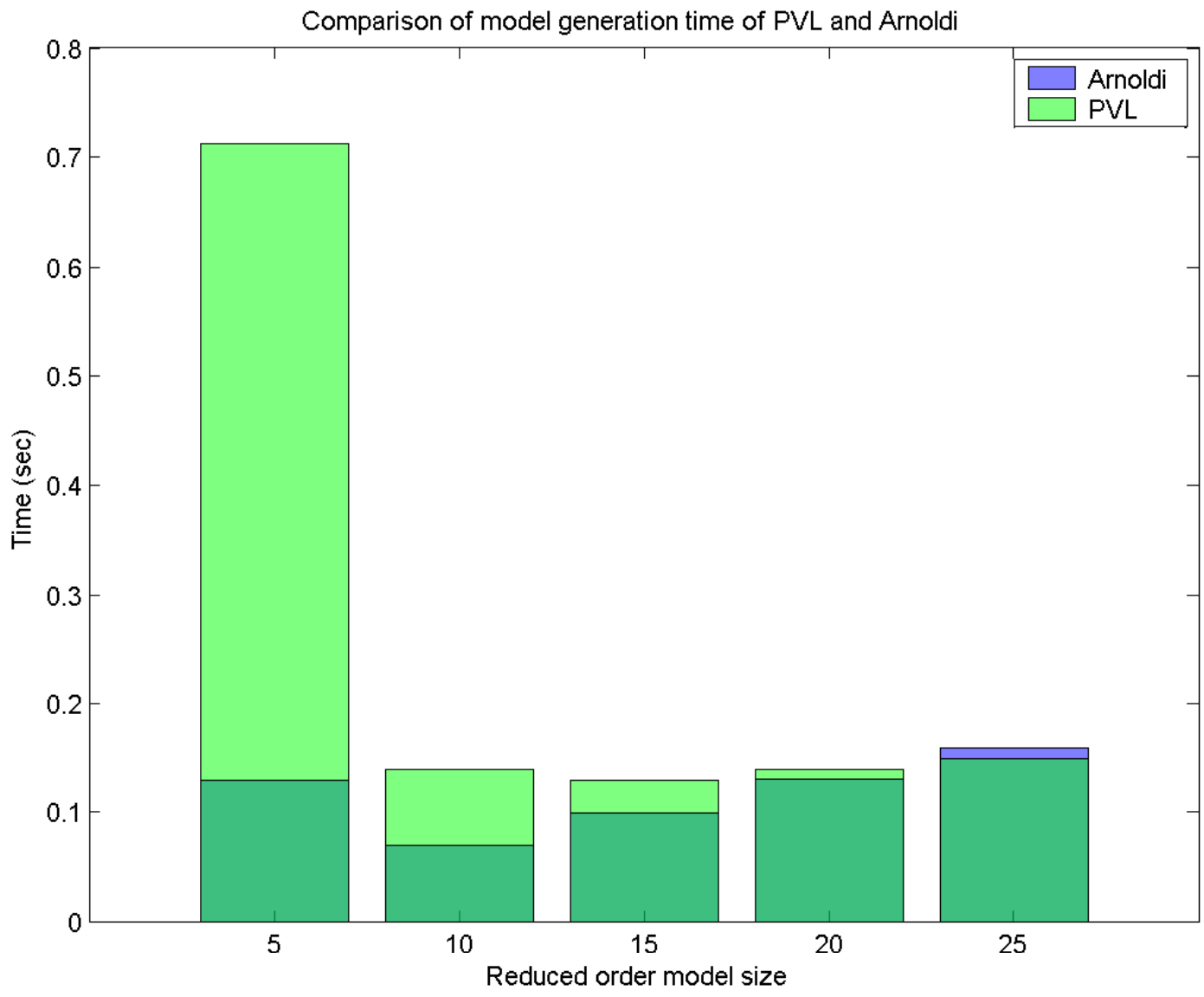
Results: PVL with reduced order =5 – 25 for a line with 100 elements



Results: Arnoldi with reduced order =5 – 25 for a line with 100 elements

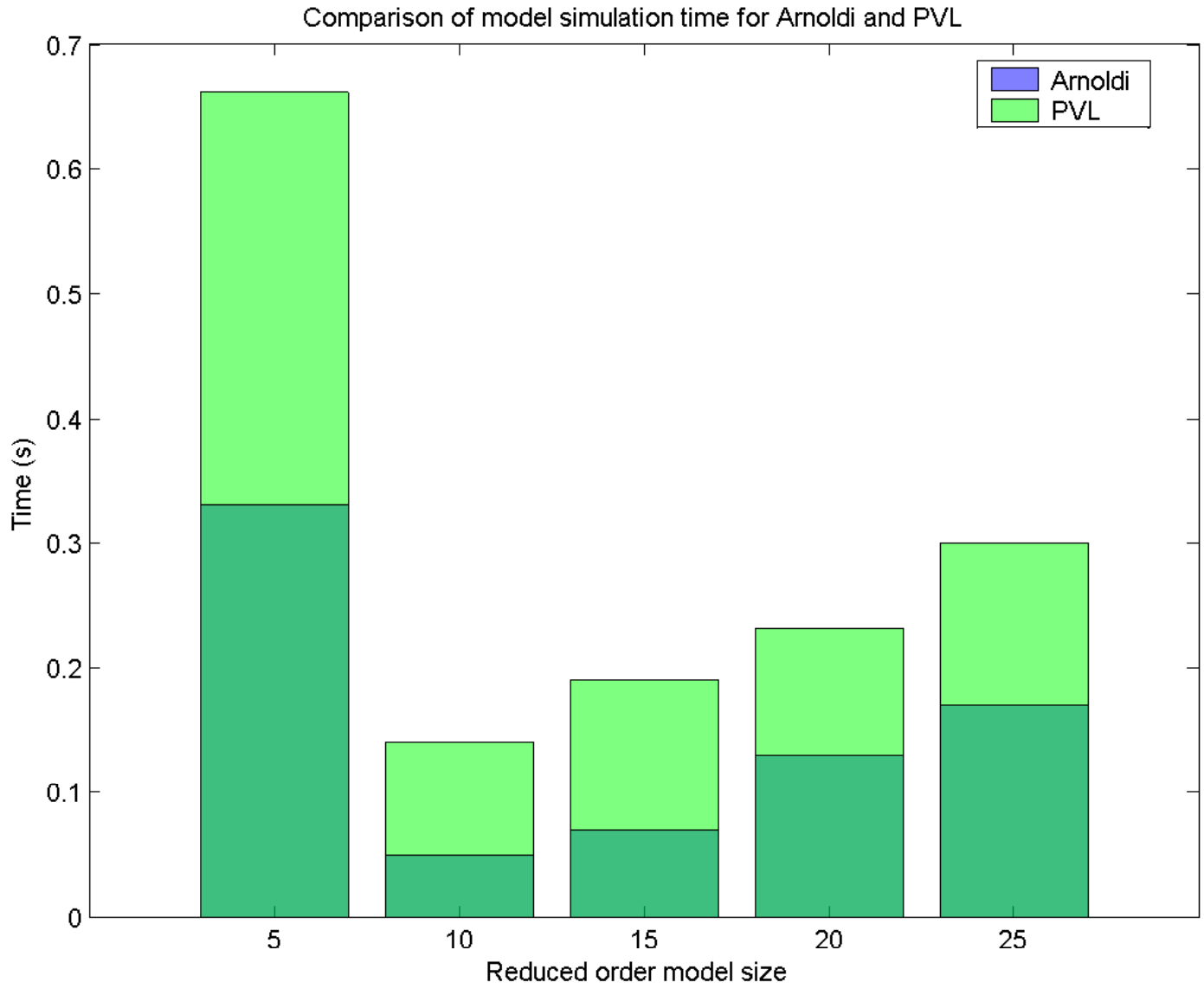


Comparison of model generation time

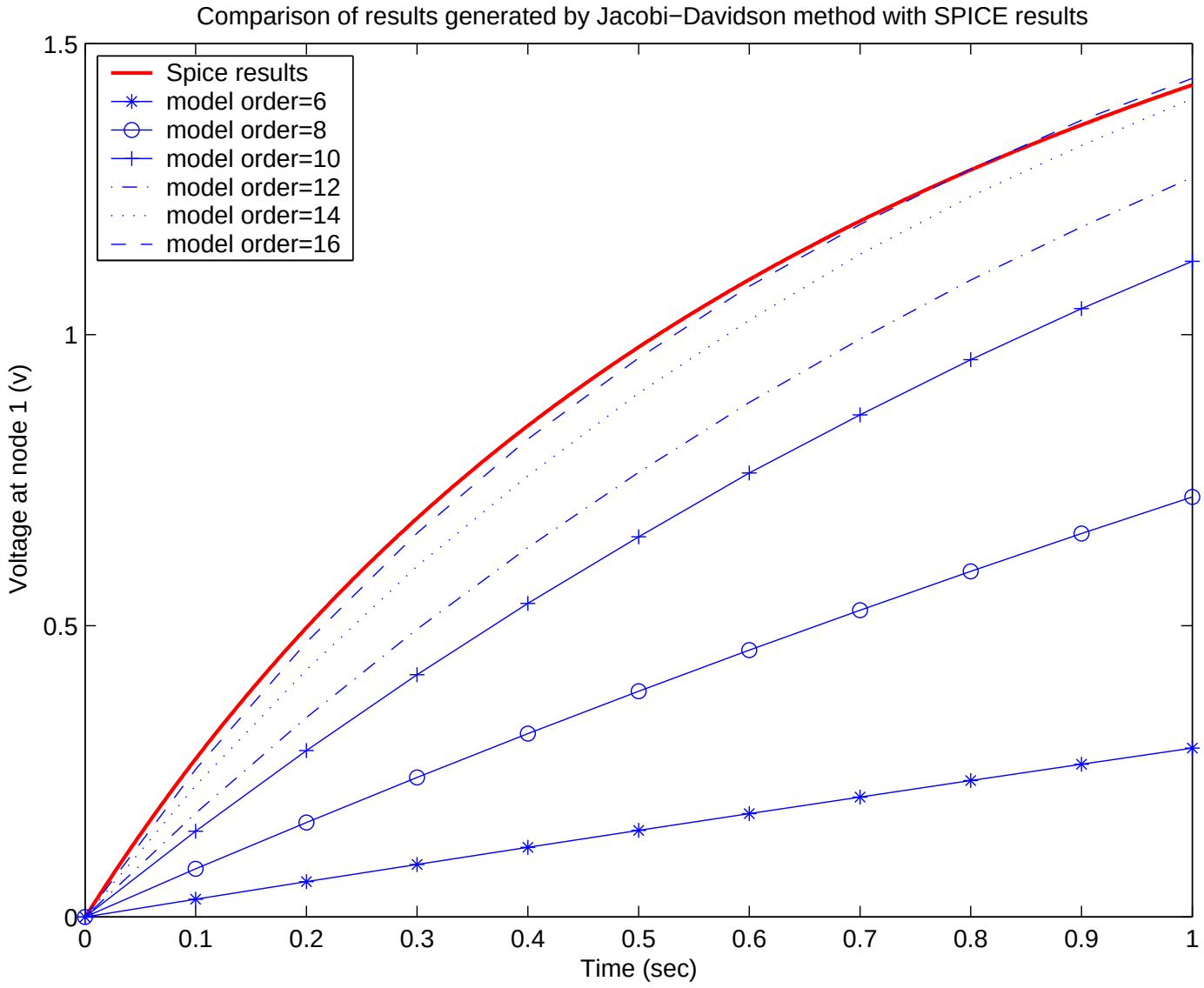


Comparing model simulation time

- Circuit simulation (SPICE) time = 3.36 sec



Results: Jacobi-Davidson with reduced order =6 – 16 for a line with 100 elements



Conclusions

- Model Order Reduction is very attarctive for CAD-synthesis & automated verification
- The linear model order reduction have been reported in literature from around a decade and the techniques AWE, PVL and Arnoldi are already too matured to be improved in any direction
- but still the main drawback that leaves the applicability of these algorithm a bit difficult as faced by us during this work was the generation of the state-space matrix in general for very large asymmetric circuits.
- The improvement in the linear algorithms would come only when we try and suggest any of the newer methods like we did an attempt to bring Jacobi-Davidson method; the simulation is expected to be improved with its two variations JDQR and JDQZ.
- The methods for weakly nonlinear systems have also been investigated by us as reported in lieterature in recent past

References

1. Nidhi Agarwal, Approaches for Reduced Order Modeling of Non-Linear VLSI Circuits: A Comaprison, M. Tech. dissertation, Dept. of ECE, MNIT Jaipur, 2007.
2. Y. Saad, Numerical methods for large Eigenvalue problems, www.umn.edu (e-book)
3. H. A. van der Vorst, Iterative methods for large linear systems, citeseer.ist.psu.edu
4. Cheng et al, Symbolic analysis and reductions of VLSI circuits, Springer, 2005
5. J.Roychowdhury “Algorithmic Macromodelling Methods for Mixed-Signal Systems” In Proc. Conf. on VLSI Design, 2004
6. P.Feldmann and R.W. Freund “Efficient linear circuit analysis by Padé approximation via the Lanczos process” IEEE Trans. CAD, 14(5):639– 649, May 1995.
7. L.M. Silveira, M. Kamon, I. Elfadel, and J. White “A coordinate transformed Arnoldi algorithm for generating guaranteed stable reduced order models of RLC circuits” In Proc. ICCAD, pages 288–294, November 1996.
8. L.T. Pillage and R.A. Rohrer “Asymptotic waveform evaluation for timing analysis” IEEE Trans. CAD, 9:352–366, April 1990.