A Multi-Core High-Order A-stable and L-Stable Integration Methods For Fast Transient Simulation of High-Speed Interconnects and Transmission Line Circuits

Mina Farhan¹, Emad Gad², Michel Nakhla¹, and Ram Achar¹

¹. Dept. of Electronics, Carleton University
². School of Electrical Engineering and Computer Science, University of Ottawa
Background

Nonlinear Devices

Measured Data

Packaging Modules

Full-Wave Modeling

SPICE

Modules

Nonlinear Devices

Full-Wave Modeling

SPICE

Packaging Modules

Measured Data

Nonlinear Devices

Full-Wave Modeling

SPICE

Packaging Modules

Measured Data

Nonlinear Devices

Full-Wave Modeling

SPICE

Packaging Modules

Measured Data
Background

Nonlinear Devices

Full-Wave Modeling

SPICE

Packaging Modules

Measured Data

Measurable Devices
Agenda

- The troubles with classical SPICE
  - The dilemma of Order and Stability
- Recent Progress in a Stable Hi-order SPICE (HiSPICE)
  - Background, Concept, Advantages
- The Multi-Core Hi-SPICE
- Simulation Results
- Conclusion
The troubles with classical SPICE

The dilemma of Order and Stability in traditional SPICE engines.
The troubles with classical SPICE

**SPICE**: A Differential Equations (DE) Solver

A DE solver discretizes the time, using small step size, solving a system of nonlinear algebraic equations at each time point.
The troubles with classical SPICE

- SPICE: A Differential Equations (DE) Solver

But Why Small step sizes? Why not Large Step Sizes?
The troubles with classical SPICE

- **Classical SPICE engine:**
  - based on the Linear Multi-Step (LMS) methods
  - Backward Euler, Trapezoidal Rule, Gear’s method

- **LMS Methods:**
  - Uses the past time steps to construct a low-order polynomials, $\sum_{i=0}^{p} a_i t^i$, to approximate the waveforms

- **Low-Order Polynomials**
  - Force small step sizes and larger than necessary time points
The troubles with classical SPICE

- LMS using High-order polynomial?

  ➢ Theoretically Unstable!

Classical SPICE (LMS) + High-Order

Larger Step

Instability
Stability in DE solvers

Laplace s-domain

A-Stable

Unstable

Stable
Stability in DE solvers

Laplace s-domain

L-Stable

Stable

Unstable
What kind of Stability in SPICE LMS

Laplace s-domain

TR

order 2

A-Stable
What kind of Stability in SPICE LMS

Gear order 2
L-Stable
What kind of Stability in SPICE LMS

Gear order 3

Laplace s-domain

Stable

Unstable
What kind of Stability in SPICE LMS

Laplace s-domain

Gear order 4
The troubles with classical SPICE

- **The Dahlquist Barriers**: it was proved theoretically that LMS are incapable of high-order without losing stability.

The Highest order of absolutely stable LMS method is 2!!
Agenda

- The troubles with classical SPICE
  - The dilemma of Order and Stability
- A-stable and L-stable High-order SPICE (HiSPICE), based on Obreshkov methods
  - Background, Concept, Advantages
- The Multi-Core Hi-SPICE
- Simulation Results
- Conclusion
The Proposed HiSPICE Algorithm

A background on previous approaches to HiSPICE

A linear circuit with $N$ unknown waveforms is represented by

$$C \frac{dx(t)}{dt} + Gx(t) = u(t)$$

- $C$ and $G$ are $N \times N$ matrices
- $u(t)$ are the independent sources

**Time Stepping**

- Assumed Known
  $$x(t_{n-1}) \equiv x_{n-1}$$
  $$h \frac{dx(t)}{dt} \bigg|_{t_{n-1}} \equiv x_{n-1}^{(1)}$$
  $$h^i \frac{d^i x(t)}{dt^i} \bigg|_{t_{n-1}} \equiv x_{n-1}^{(i)}$$

- Need to be computed
  $$x(t_n) \equiv x_n$$

- Added Unknowns
  $$h \frac{dx(t)}{dt} \bigg|_{t_n} \equiv x_n^{(1)}$$
  $$h^j \frac{d^j x(t)}{dt^j} \bigg|_{t_n} \equiv x_n^{(j)}$$
At each time step, we will need to solve an enlarged system of algebraic equations: \( j \) times the size of the original system \( (Nj) \). Obtained from \( u(t) \), its derivatives, and the past time point.
The Proposed HiSPICE Algorithm

A background on previous approaches to HiSPICE

The block structure of the augmented matrices $\tilde{C}, \tilde{G} \in \mathbb{R}^{N_j \times N_j}$

$$C \frac{dx(t)}{dt} + G x(t) = u(t)$$

$$\tilde{C} = \begin{bmatrix} C & -C_{pq} \\ \hline & & & \end{bmatrix}$$

$$(p - 1) \times j + 1$$

$$\tilde{G} = \begin{bmatrix} (q - 1) \times j + 1 \hline \hline 0 & 1 & \ldots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & 1 \\ -\frac{\alpha_0}{\alpha_j} & \ldots & -\frac{\alpha_{j-1}}{\alpha_j} \\ \hline \hline \end{bmatrix}$$

$$p \times j$$

$$q \times j$$

$$\alpha_k = (-1)^{(j)} \frac{(i+j-k)!j!}{i!(i+j)!(j-k)!}$$
The Proposed HiSPICE Algorithm

A background on previous approaches to HiSPICE

The block structure of the augmented matrices \( \tilde{C}, \tilde{G} \in \mathbb{R}^{Nj \times Nj} \)

\[
C \frac{dx(t)}{dt} + Gx(t) = u(t)
\]
A-Stable High-order SPICE (HiSPICE):

The main concept

- Order? $i + j$

- Stability?
  - Necessary and sufficient condition for A-stability is $j - 2 \leq i < j$
  - Necessary and sufficient condition for L-stability is $j - 2 \leq i < j$
What kind of Stability in HiSPICE

Laplace s-domain \( i = j = 1 \)

HiSPICE

order 2

A-Stable
Stability in DE solvers

Laplace s-domain \( i = 1, j = 2 \)

HiSPICE

order 3

L-Stable
What kind of Stability in HiSPICE

Laplace s-domain $i = j = 4$

HiSPICE order 8

A-Stable

Stable

Unstable
A-Stable High-order SPICE (HiSPICE):

A sample of the results

<table>
<thead>
<tr>
<th>Order</th>
<th># time points</th>
<th>Speedup</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classical SPICE (2)</td>
<td>1771</td>
<td>-</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>HiSPICE (4)</td>
<td>101</td>
<td>6.5</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>HiSPICE (6)</td>
<td>36</td>
<td>10.3</td>
<td>&lt; 0.001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order</th>
<th># time points</th>
<th>Speedup</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classical SPICE (2)</td>
<td>5751</td>
<td>-</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>HiSPICE (4)</td>
<td>185</td>
<td>12</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>HiSPICE(6)</td>
<td>59</td>
<td>21.4</td>
<td>&lt; 0.0001</td>
</tr>
</tbody>
</table>
A-Stable High-order SPICE (HiSPICE):
A sample of the results

- High-Order, 8, $i=j=4$
- Classical SPICE, order 2, TR
A-Stable High-order SPICE (HiSPICE):

- To Summarize
  - At the expense of solving a larger system of equations, compared to the low-order method, we are able to save on the CPU time significantly, in addition to solving the difficult problem of stability.
  - This is previous work that was published (1),(2)


This Work

Presents a new way to reduce the computational cost per time step for linear circuits.
Agenda

- The troubles with classical SPICE
  - The dilemma of Order and Stability
- A-stable and L-stable High-order SPICE (HiSPICE), based on Obreshkov methods
  - Background, Concept, Advantages
- The Proposed MultiCore HiSPICE
- Simulation Results
- Conclusion
The Proposed HiSPICE Algorithm

Proposed Multi-Core approach to HiSPICE

1. Problem Reformulation
2. Problem Permutation
3. Multi-Core Implementation
The Proposed HiSPICE Algorithm

Proposed Multi-Core approach to HiSPICE

\[ V (g_{pq} I + c_{pq} \lambda) V^{-1} \]

- \( V \) is common to all the blocks in the matrix
- \( \lambda \) is a diagonal matrix

\[ \tilde{G} + \tilde{C} = \tilde{V} j \tilde{V}^{-1} \]

\[ \tilde{V} = \begin{bmatrix} V & 0 & \ldots & 0 \\ 0 & V & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & V \end{bmatrix} \]

\[ \tilde{j} = \begin{bmatrix} \tilde{G} + \tilde{C} \end{bmatrix} \]

Block Diagonal

Diagonal blocks

\( N j \times N j \) matrix
The Proposed HiSPICE Algorithm

Proposed Multi-Core approach to HiSPICE

\[
\left( \tilde{C} + \tilde{G} \right) \tilde{x}_n = \tilde{u}_n
\]

Using a change of variables

\[
\tilde{\rho}_n \leftarrow \hat{V}^{-1} \tilde{x}_n
\]

\[
\tilde{J} \tilde{\rho}_n = \hat{V}^{-1} \tilde{u}_n
\]

\[\hat{J} = \begin{pmatrix}
\cdot & \cdot & \cdots & \cdot \\
\cdot & \cdot & \cdots & \cdot \\
\vdots & \vdots & \ddots & \vdots \\
\cdot & \cdot & \cdots & \cdot \\
\end{pmatrix}\]

Diagonal blocks
The Proposed HiSPICE Algorithm

Proposed Multi-Core approach to HiSPICE

\[ \tilde{J} \tilde{\rho} = \hat{V}^{-1} \hat{\tilde{u}} \]

Problem Reformulation

Problem Permutation

Multi-Core Implementation

Diagonal blocks

\[ \tilde{J} = \begin{bmatrix} \text{Diagonal blocks} \end{bmatrix} \]

Sparse \( N \times N \) block

Block Diagonal

j blocks
The Proposed HiSPICE Algorithm

Proposed Multi-Core approach to HiSPICE

Problem Reformulation

Problem Permutation

Multi-Core Implementation

- Blocks can be factorized using separate threads executed in parallel
- All the blocks have identical structures
  - Sparse ordering is done only once
- Some of the blocks are complex, some are real
- Complex blocks are conjugate pairs
  - Needs complex factorization libraries (klu_zl_xxx)
  - Conjugate-ness is exploited to save computations

Sparse $N \times N$ block

Block Diagonal
Agenda

• The troubles with classical SPICE
  • The dilemma of Order and Stability
• A-stable and L-stable High-order SPICE (HiSPICE), based on Obreshkov methods
  • Background, Concept, Advantages
• The Proposed MultiCore HiSPICE
• Simulation Results
• Conclusion
Simulation Results

A circuit with 48 coupled conductor transmission lines

- Line 1
- Line 2
- Line 3
- Line 48

Parameters:
- $w = 0.5 \text{ mm}$
- $h = 1 \text{ mm}$
- $\rho = 1.8 \text{ mm}$
- $\varepsilon_r = 4$
<table>
<thead>
<tr>
<th>Order</th>
<th>CPU (seconds)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classical SPICE (2)</td>
<td>145.17</td>
<td>-</td>
</tr>
<tr>
<td>HiSPICE (10)</td>
<td>11.20</td>
<td>13</td>
</tr>
</tbody>
</table>
The CPU of a single time point

- Proposed Multi-Core HiSPICE
- Previous HiSPICE

CPU (Sec.)

Order

Classical SPICE
Conclusion

- High-Order A-stable and L-stable SPICE (HiSPICE)
  - Taking advantage of multicore architecture
- Cost per time point:
  - Independent from the order
  - Comparable to Low Order Cost
- The algorithm is developed for linear circuits arising in package models.
Thank You