Wavelet-Based Simulation Technique in the Time Domain

Kai Bittner

University of Wuppertal
Circuit equations

\[ \frac{d}{dt} q(x(t)) + g(x(t), t) = 0, \quad t \in [0, T], \quad x(0) = x_0 \]
Petrov-Galerkin discretization

- Circuit equations

\[ \frac{d}{dt} q(x(t)) + g(x(t), t) = 0, \quad t \in [0, T], \quad x(0) = x_0 \]

- Approximate the solution as \( x(t) = \sum_i c_i \varphi_i(t) \) with suitable ansatz functions \( \varphi_i \).
Circuit equations

\[
\frac{d}{dt} q(x(t)) + g(x(t), t) = 0, \quad t \in [0, T], \quad x(0) = x_0
\]

Approximate the solution as \( x(t) = \sum_i c_i \varphi_i(t) \) with suitable ansatz functions \( \varphi_i \).

Discretization: \( \sum_{i=0}^n c_i \varphi_i(0) = x_0 \) and

\[
F_\ell(c) := \int_0^T \left( \frac{d}{dt} q(x(t)) + g(x(t), t) \right) \theta_\ell(t) \, dt = 0,
\]

for \( \ell = 1, \ldots, n \) and suitable test functions \( \theta_\ell \)
Wavelet approach

Evaluation of $q(x(t))$ and $g(x(t), t)$ requires efficient computation of $x(t)$.

Spline wavelets

Use B-spline representation for computation of $F^\ell(c)$ and Jacobian $F'\ell(c)$ in Newton iteration.

Test functions:

$\theta^\ell = \chi[\tau^\ell - 1, \tau^\ell]$

Wavelet representation for grid adaptation

Switch between representations by fast wavelet transform.
Wavelet approach

- Evaluation of \( q(x(t)) \) and \( g(x(t), t) \) requires efficient computation of \( x(t) \).  \( \rightsquigarrow \) Spline wavelets

Wavelet representation for grid adaptation

Switch between representations by fast wavelet transform.
Evaluation of $q(x(t))$ and $g(x(t), t)$ requires efficient computation of $x(t)$.  $\rightsquigarrow$ Spline wavelets

- Use B-spline representation for computation of $F_\ell(c)$ and Jacobian $F'_\ell(c)$ in Newton iteration.
Wavelet approach

- Evaluation of $q(x(t))$ and $g(x(t), t)$ requires efficient computation of $x(t)$. $\rightsquigarrow$ Spline wavelets
- Use B-spline representation for computation of $F_\ell(c)$ and Jacobian $F'_\ell(c)$ in Newton iteration.
- Test functions: $\theta_\ell = \chi[\tau_{\ell-1}, \tau_\ell]$

$$\chi_I(x) = \begin{cases} 1, & \text{if } x \in I, \\ 0, & \text{otherwise} \end{cases}$$
Evaluation of \( q(x(t)) \) and \( g(x(t), t) \) requires efficient computation of \( x(t) \).  

\( \implies \) Spline wavelets

Use B-spline representation for computation of \( F_\ell(c) \) and Jacobian \( F'_\ell(c) \) in Newton iteration.

Test functions: \( \theta_\ell = \chi[\tau_{\ell-1}, \tau_\ell] \)

Wavelet representation for grid adaptation
Wavelet approach

- Evaluation of $q(x(t))$ and $g(x(t), t)$ requires efficient computation of $x(t)$. ~⇒ Spline wavelets
- Use B-spline representation for computation of $F_\ell(c)$ and Jacobian $F'_\ell(c)$ in Newton iteration.
- Test functions: $\theta_\ell = \chi[\tau_{\ell-1}, \tau_\ell]$

- Wavelet representation for grid adaptation
- Switch between representations by fast wavelet transform.
Algorithm

**Input:** Initial grid $\mathcal{T}^{(0)}$, initial guess $\mathbf{c}^{(0)}$

- $\ell := 0$
Algorithm

Input: Initial grid $\mathcal{T}^{(0)}$, initial guess $c^{(0)}$

1. $\ell := 0$
2. DO
   - Solve by Newton’s method: $c^{(\ell)} \rightarrow \tilde{c}^{(\ell)}$
3. $\ell := \ell + 1$
4. UNTIL required accuracy is achieved

Output: $\mathcal{T}^{(\ell)}$, $c^{(\ell)}$
Algorithm

**Input:** Initial grid $\mathcal{T}^{(0)}$, initial guess $\mathbf{c}^{(0)}$

- $\ell := 0$
- DO
  1. Solve by Newton’s method: $\mathbf{c}^{(\ell)} \rightarrow \tilde{\mathbf{c}}^{(\ell)}$
  2. Wavelet refinement: $(\mathcal{T}^{(\ell)}, \tilde{\mathbf{c}}^{(\ell)}) \rightarrow (\mathcal{T}^{(\ell+1)}, \mathbf{c}^{(\ell+1)})$

**Output:** $\mathcal{T}^{(\ell)}$, $\mathbf{c}^{(\ell)}$
Algorithm

Input: Initial grid $\mathcal{T}^{(0)}$, initial guess $c^{(0)}$

\begin{itemize}
  \item $\ell := 0$
  \item DO
    \begin{enumerate}
      \item Solve by Newton’s method: $c^{(\ell)} \rightarrow \tilde{c}^{(\ell)}$
      \item Wavelet refinement: $(\mathcal{T}^{(\ell)}, \tilde{c}^{(\ell)}) \rightarrow (\mathcal{T}^{(\ell+1)}, c^{(\ell+1)})$
      \item $\ell := \ell + 1$
    \end{enumerate}
  \item UNTIL required accuracy is achieved
\end{itemize}
Algorithm

**Input:** Initial grid $\mathcal{T}^{(0)}$, initial guess $\mathbf{c}^{(0)}$

- $\ell := 0$
- **DO**
  1. Solve by Newton’s method: $\mathbf{c}^{(\ell)} \rightarrow \tilde{\mathbf{c}}^{(\ell)}$
  2. Wavelet refinement: $(\mathcal{T}^{(\ell)}, \tilde{\mathbf{c}}^{(\ell)}) \rightarrow (\mathcal{T}^{(\ell+1)}, \mathbf{c}^{(\ell+1)})$
  3. $\ell := \ell + 1$

**UNTIL** required accuracy is achieved

**Output:** $\mathcal{T}^{(\ell)}$, $\mathbf{c}^{(\ell)}$
Sub-interval wavelet method

- Large time interval $[0, T]$ may result in excessive computational cost.

Solution:
- Split into small sub-intervals
- Quasi-periodic behavior permits the reuse of a sub-interval solution and grid as initial guess for the next subinterval.

$\Rightarrow$ Speedup of computation
Large time interval $[0, T]$ may result in excessive computational cost.

**Solution:** Split into small sub-intervals
Sub-interval wavelet method

- Large time interval \([0, T]\) may result in excessive computational cost.

- **Solution**: Split into small sub-intervals

- Quasi-periodic behavior permits the reuse of a sub-interval solution and grid as *initial guess for the next subinterval*. 

K. Bittner  
Wavelet-Based Simulation Technique in the Time Domain
Large time interval $[0, T]$ may result in excessive computational cost.

**Solution:** Split into small sub-intervals

Quasi-periodic behavior permits the reuse of a sub-interval solution and grid as initial guess for the next subinterval.

$\Rightarrow$ Speedup of computation
Example — Amplifier (transAmp)

Input and output signal

Detail
Grid size versus error

CPU time versus error

for transient analysis and subinterval wavelet method
Adaptivity in the state variables

\[ x = \sum_{i=1}^{n} c_i \varphi_i \] each component \( x_\mu \) of \( x \) has the same representation
Adaptivity in the state variables

- In $x = \sum_{i=1}^{n} c_i \varphi_i$ each component $x_\mu$ of $x$ has the same representation.
- Different signal behavior $\xrightarrow{\sim}$ different representations $x_\mu = \sum_{i=1}^{n_\mu} c_{\mu,i} \varphi_{\mu,i}$ are more efficient.
Adaptivity in the state variables

- In $x = \sum_{i=1}^{n} c_i \varphi_i$ each component $x_\mu$ of $x$ has the same representation.

- Different signal behavior $\Rightarrow$ different representations $x_\mu = \sum_{i=1}^{n_\mu} c_{\mu,i} \varphi_{\mu,i}$ are more efficient.

- Discretization:

$$F_{\mu,\ell}(c) := \int_{0}^{T} \left( \frac{d}{dt} q_{\nu \mu}(x(t)) + g_{\nu \mu}(x(t), t) \right) \theta_{\mu,\ell}(t) dt = 0,$$
Adaptivity in the state variables

- In $x = \sum_{i=1}^{n} c_i \varphi_i$ each component $x_\mu$ of $x$ has the same representation
- Different signal behavior $\Rightarrow$ different representations $x_\mu = \sum_{i=1}^{n_\mu} c_{\mu,i} \varphi_{\mu,i}$ are more efficient
- Discretization:

$$F_{\mu,\ell}(c) := \int_{0}^{T} \left( \frac{d}{dt} q_{\nu_{\mu}}(x(t)) + g_{\nu_{\mu}}(x(t), t) \right) \theta_{\mu,\ell}(t) dt = 0,$$

- Challenges: Choose $\nu_{\mu}$
Adaptivity in the state variables

- In $x = \sum_{i=1}^{n} c_i \varphi_i$ each component $x_\mu$ of $x$ has the same representation.
- Different signal behavior $\Rightarrow$ different representations $x_\mu = \sum_{i=1}^{n_\mu} c_{\mu,i} \varphi_{\mu,i}$ are more efficient.
- Discretization:

$$F_{\mu,\ell}(c) := \int_{0}^{T} \left( \frac{d}{dt} q_{\nu_{\mu}}(x(t)) + g_{\nu_{\mu}}(x(t), t) \right) \theta_{\mu,\ell}(t) \, dt = 0,$$

- Challenges:
  - Choose $\nu_{\mu}$
  - Balance the refinement of single components
Input, intermediate voltage and output, with corresponding grid
Conclusions

- New spline wavelet method developed and implemented
Conclusions

- New spline wavelet method developed and implemented
- Wavelet algorithm can reach performance of traditional methods
Conclusions

- New spline wavelet method developed and implemented
- Wavelet algorithm can reach performance of traditional methods
- Adaptivity in state variables under development
Thank you

Questions?