Wavelet-Based Simulation Technique in the Time Domain

Kai Bittner

University of Wuppertal

Petrov-Galerkin discretization

- Circuit equations

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

Petrov-Galerkin discretization

- Circuit equations

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

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

Petrov-Galerkin discretization

- Circuit equations

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

- Approximate the solution as $x(t) = \sum_i c_i \, \varphi_i(t)$ with suitable ansatz functions φ_i.
- Discretization: $\sum_{i=0}^{n} c_i \, \varphi_i(0) = x_0$ and

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

for $\ell = 1, \ldots, n$ and suitable test functions θ_ℓ

Wavelet approach

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

Wavelet approach

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

Wavelet approach

- Evaluation of $q\big(x(t)\big)$ and $g\big(x(t), t\big)$ requires efficient computation of $x(t)$. $\rightsquigarrow$ Spline wavelets
- Use B-spline representation for computation of $F_\ell(\mathbf{c})$ and Jacobian $F'_\ell(\mathbf{c})$ in Newton iteration.

Wavelet approach

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

$$\chi_I(x) = \begin{cases} 1, & \text{if } x \in I, \\ 0 & \text{otherwise} \end{cases}$$

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 $F_\ell(\mathbf{c})$ and Jacobian $F'_\ell(\mathbf{c})$ in Newton iteration.
- Test functions: $\theta_\ell = \chi_{[\tau_{\ell-1}, \tau_\ell]}$
- Wavelet representation for grid adaptation

Wavelet approach

- Evaluation of $q\big(x(t)\big)$ and $g\big(x(t), t\big)$ requires efficient computation of $x(t)$. $\rightsquigarrow$ Spline wavelets
- Use B-spline representation for computation of $F_\ell(\mathbf{c})$ and Jacobian $F'_\ell(\mathbf{c})$ 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.

Adaptive Wavelet Solver

Algorithm

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

- $\ell := 0$

Adaptive Wavelet Solver

Algorithm

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

- $\ell := 0$
- DO
  1. Solve by Newton's method: $\mathbf{c}^{(\ell)} \longrightarrow \tilde{\mathbf{c}}^{(\ell)}$

Adaptive Wavelet Solver

Algorithm

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

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

Adaptive Wavelet Solver

Algorithm

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

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

  UNTIL required accuracy is achieved

Adaptive Wavelet Solver

Algorithm

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

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

Sub-interval wavelet method

- Large time interval $[0, T]$ may result in excessive computational cost.
- **Solution:** Split into small sub-intervals

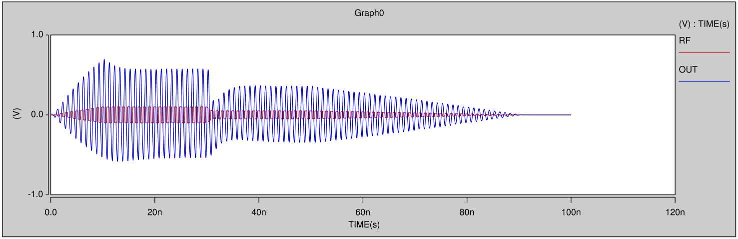
Sub-interval wavelet method

- 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.

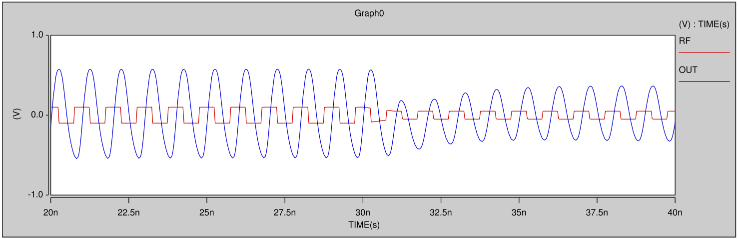
Sub-interval wavelet method

- 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.
  $\rightsquigarrow$ Speedup of computation

Example — Amplifier (transAmp)

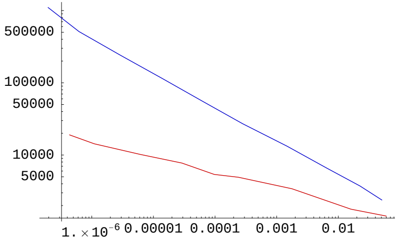


Input and output signal

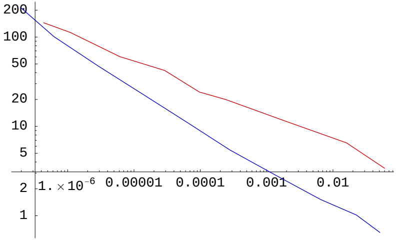


Detail

Example — Amplifier (transAmp)



Grid size versus error



CPU time versus error

for transient analysis and subinterval wavelet method

Adaptivity in the state variables

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

Adaptivity in the state variables

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

$$F_{\mu,\ell}(\mathbf{c}) := \int_0^T \Big(\tfrac{d}{dt} q_{\nu_\mu}(x(t)) + g_{\nu_\mu}(x(t), t) \Big) \theta_{\mu,\ell}(t)\, dt = 0,$$

Adaptivity in the state variables

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

$$F_{\mu,\ell}(\mathbf{c}) := \int_0^T \Big(\tfrac{d}{dt} q_{\nu_\mu}(x(t)) + g_{\nu_\mu}(x(t), t) \Big) \theta_{\mu,\ell}(t) \, dt = 0,$$

- Challenges:
  - Choose ν_μ

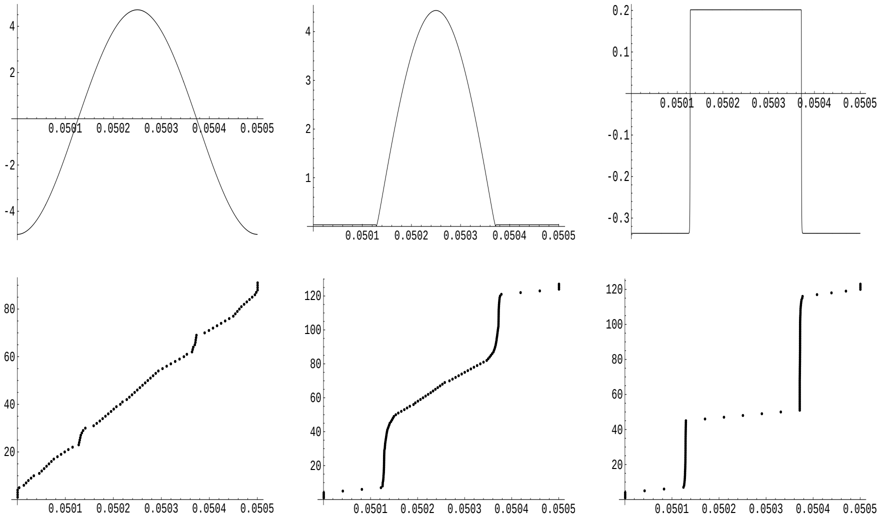
Adaptivity in the state variables

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

$$F_{\mu,\ell}(\mathbf{c}) := \int_0^T \Big(\tfrac{d}{dt} q_{\nu_\mu}(x(t)) + g_{\nu_\mu}(x(t), t) \Big) \theta_{\mu,\ell}(t) \, dt = 0,$$

- Challenges:
  - Choose ν_μ
  - Balance the refinement of single components

Example — Schmitt trigger



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?