Wavelet algorithm for circuit simulation

Kai Bittner[1] Emira Dautbegovic[2]

[1]University of Wuppertal

[2]Infineon Technologies AG

Wavelet expansions

- Representation of $f : [a, b] \to \mathbb{R}$ as

$$f = \sum_{k \in \mathcal{I}} c_k \, \phi_k + \sum_{j=0}^{\infty} \sum_{k \in \Lambda_j} d_{jk} \, \psi_{jk}.$$

Wavelet expansions

- Representation of $f : [a, b] \to \mathbb{R}$ as

$$f = \sum_{k \in \mathcal{I}} c_k \, \phi_k + \sum_{j=0}^{\infty} \sum_{k \in \Lambda_j} d_{jk} \, \psi_{jk}.$$

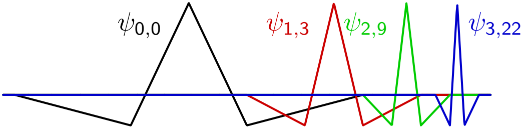
- Coarse approximation $\sum_{k \in \mathcal{I}} c_k \, \phi_k$

Wavelet expansions

- Representation of $f : [a, b] \to \mathbb{R}$ as

$$f = \sum_{k \in \mathcal{I}} c_k \, \phi_k + \sum_{j=0}^{\infty} \sum_{k \in \Lambda_j} d_{jk} \, \psi_{jk}.$$

- Coarse approximation $\sum_{k \in \mathcal{I}} c_k \, \phi_k$
- Details of increasing resolution j in terms of wavelets ψ_{jk}

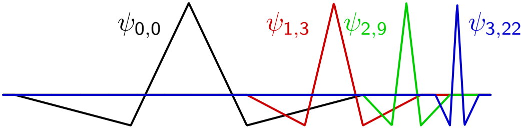


Wavelet expansions

- Representation of $f : [a, b] \to \mathbb{R}$ as

$$f = \sum_{k \in \mathcal{I}} c_k\, \phi_k + \sum_{j=0}^{\infty} \sum_{k \in \Lambda_j} d_{jk}\, \psi_{jk}.$$

- Coarse approximation $\sum_{k \in \mathcal{I}} c_k\, \phi_k$
- Details of increasing resolution j in terms of wavelets ψ_{jk}



- Infinite representation $\rightsquigarrow$ finite approximation needed

Wavelet approximation

Thresholding: $$f \approx f_\varepsilon := \sum_{k \in \mathcal{I}} c_k \, \phi_k + \sum_{(j,k):\ |d_{jk}| > \varepsilon} d_{jk} \, \psi_{jk}$$

Wavelet approximation

Thresholding: $$f \approx f_\varepsilon := \sum_{k \in \mathcal{I}} c_k \, \phi_k + \sum_{(j,k):\ |d_{jk}| > \varepsilon} d_{jk} \, \psi_{jk}$$

- Applicable only if all wavelet coefficients are known.

Wavelet approximation

Thresholding: $$f \approx f_\varepsilon := \sum_{k \in \mathcal{I}} c_k \, \phi_k + \sum_{(j,k):\ |d_{jk}| > \varepsilon} d_{jk} \, \psi_{jk}$$

- Applicable only if estimates for all wavelet coefficients are known.

Wavelet approximation

Thresholding: $$f \approx f_\varepsilon := \sum_{k \in \mathcal{I}} c_k \, \phi_k + \sum_{(j,k):\ |d_{jk}| > \varepsilon} d_{jk} \, \psi_{jk}$$

- Applicable only if estimates for all wavelet coefficients are known.
- Typically used for compression of inefficient approximations

Wavelet approximation

Thresholding: $$f \approx f_\varepsilon := \sum_{k\in\mathcal{I}} c_k\,\phi_k + \sum_{(j,k):\ |d_{jk}|>\varepsilon} d_{jk}\,\psi_{jk}$$

- Applicable only if estimates for all wavelet coefficients are known.
- Typically used for compression of inefficient approximations

e.g., $$\sum_{k\in\mathcal{I}_J} f(t_{J,k})\,\phi_{J,k} = \sum_{k\in\mathcal{I}} c_k\,\phi_k + \sum_{j=0}^{J}\sum_{k\in\Lambda_j} d_{jk}\,\psi_{jk}.$$

Wavelet approximation

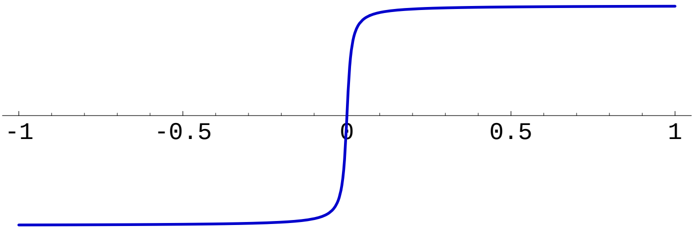
Thresholding: $$f \approx f_\varepsilon := \sum_{k\in\mathcal{I}} c_k\,\phi_k + \sum_{(j,k):\ |d_{jk}|>\varepsilon} d_{jk}\,\psi_{jk}$$

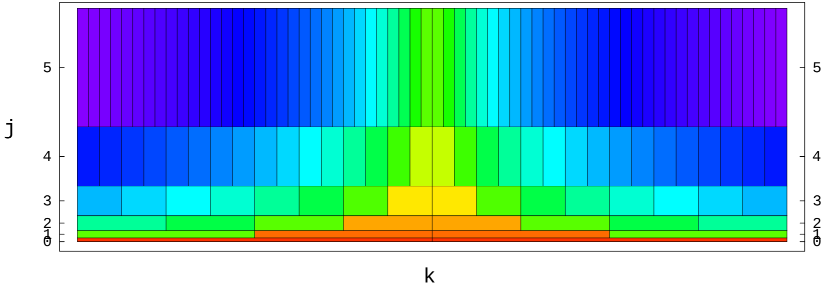
- Applicable only if estimates for all wavelet coefficients are known.
- Typically used for compression of inefficient approximations

e.g., $$\sum_{k\in\mathcal{I}_J} f(t_{J,k})\,\phi_{J,k} = \sum_{k\in\mathcal{I}} c_k\,\phi_k + \sum_{j=0}^{J}\sum_{k\in\Lambda_j} d_{jk}\,\psi_{jk}.$$

$\{f(t_{J,k})\} \to \{c_k\}, \{d_{jk}\}$ by fast wavelet transform.

Example of wavelet expansion





Magnitude of wavelet coefficients of function f

Important properties of Wavelets

- **Stability:** $\left\|(d_{j,k})\right\| \sim \|f\|$

Important properties of Wavelets

- **Stability:** $\left\|(d_{j,k})\right\| \sim \|f\|$
- **Polynomial exactness:**

$$p = \sum_{k \in \mathcal{I}} c_k(p)\, \phi_k, \quad \text{for } p \in \Pi_{m-1}$$

Important properties of Wavelets

- **Stability:** $\left\|(d_{j,k})\right\| \sim \|f\|$
- **Polynomial exactness:**

$$p = \sum_{k \in \mathcal{I}} c_k(p)\, \phi_k, \quad \text{for } p \in \Pi_{m-1}$$

- **Locality:** e.g., $\operatorname{supp} \psi_{jk} \subset 2^{-j}\big([a, b] + k\big).$

Important properties of Wavelets

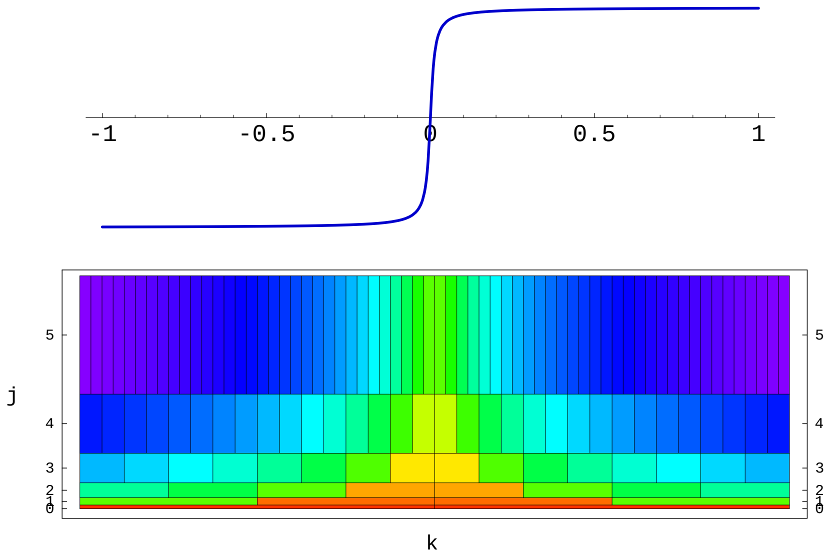
- **Stability:** $\left\|(d_{j,k})\right\| \sim \|f\|$
- **Polynomial exactness:**

$$p = \sum_{k \in \mathcal{I}} c_k(p)\, \phi_k, \quad \text{for } p \in \Pi_{m-1}$$

- **Locality:** e.g., $\operatorname{supp} \psi_{jk} \subset 2^{-j}\big([a,b]+k\big)$.

$$\rightsquigarrow \quad \left|d_j^k\right| \le C\, 2^{-j\alpha} \left\| f|_{\operatorname{supp} \psi_{jk}} \right\|_{W_p^\alpha}, \quad \alpha < m$$

Example of wavelet expansion



Magnitude of wavelet coefficients of function f

Adaptive Wavelet Solver

- Expand solution $x(t)$ of charge oriented MNA circuit equation

$$\frac{d}{dt}q\big(x(t)\big) + g\big(x(t)\big) = s(t)$$

in terms of spline wavelets

Adaptive Wavelet Solver

- Expand solution $x(t)$ of charge oriented MNA circuit equation

$$\frac{d}{dt}q\big(x(t)\big) + g\big(x(t)\big) = s(t)$$

  in terms of spline wavelets

- Petrov-Galerkin discretization $\rightsquigarrow$ nonlinear system

Adaptive Wavelet Solver

- Expand solution $x(t)$ of charge oriented MNA circuit equation

$$\frac{d}{dt}q\left(x(t)\right) + g\left(x(t)\right) = s(t)$$

  in terms of spline wavelets

- Petrov-Galerkin discretization $\rightsquigarrow$ nonlinear system
- Wavelet based adaptive refinements during Newton iteration

Adaptive Wavelet Solver

- Expand solution $x(t)$ of charge oriented MNA circuit equation

$$\frac{d}{dt}q(x(t)) + g(x(t)) = s(t)$$

in terms of spline wavelets

- Petrov-Galerkin discretization $\rightsquigarrow$ nonlinear system
- Wavelet based adaptive refinements during Newton iteration
- Efficient implementation

Adaptive Wavelet Solver

- Expand solution $x(t)$ of charge oriented MNA circuit equation

$$\frac{d}{dt}q\big(x(t)\big) + g\big(x(t)\big) = s(t)$$

  in terms of spline wavelets

- Petrov-Galerkin discretization $\rightsquigarrow$ nonlinear system
- Wavelet based adaptive refinements during Newton iteration
- Efficient implementation due to
  - fast wavelet transform

Adaptive Wavelet Solver

- Expand solution $x(t)$ of charge oriented MNA circuit equation

$$\frac{d}{dt} q\big(x(t)\big) + g\big(x(t)\big) = s(t)$$

  in terms of spline wavelets

- Petrov-Galerkin discretization $\rightsquigarrow$ nonlinear system
- Wavelet based adaptive refinements during Newton iteration
- Efficient implementation due to
  - fast wavelet transform
  - adaptive representation $\rightsquigarrow$ minimal size of problem

Implementation and numerical tests

- Implemented in the framework of an industrial circuit simulator.

Implementation and numerical tests

- Implemented in the framework of an industrial circuit simulator.
  - Coded in C

Implementation and numerical tests

- Implemented in the framework of an industrial circuit simulator.
  - Coded in C
  - Special care needed to connect to existing code

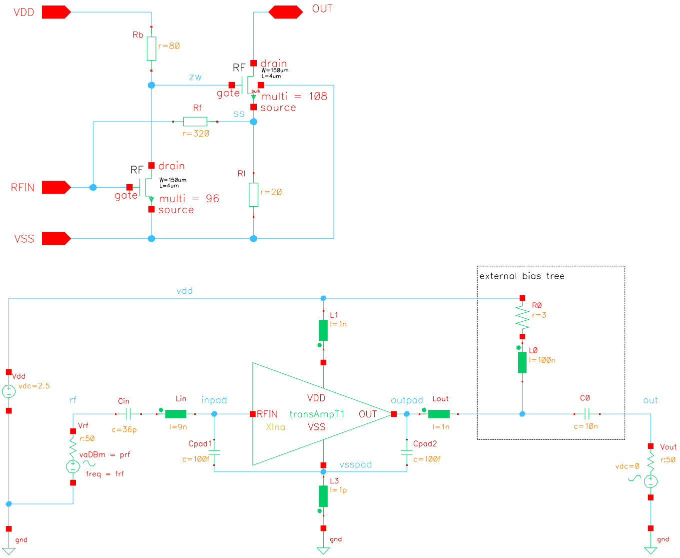
Implementation and numerical tests

- Implemented in the framework of an industrial circuit simulator.
  - Coded in C
  - Special care needed to connect to existing code
- Accuracy and performance tested against highly optimized transient analysis.

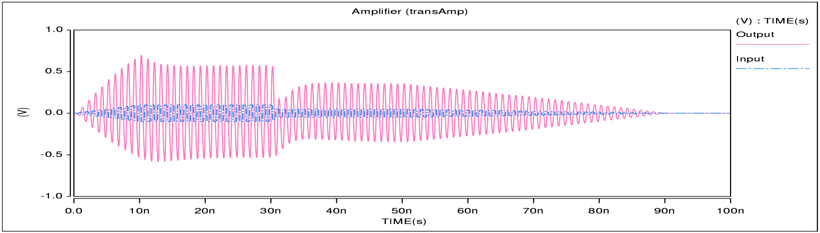
Implementation and numerical tests

- Implemented in the framework of an industrial circuit simulator.
  - Coded in C
  - Special care needed to connect to existing code
- Accuracy and performance tested against highly optimized transient analysis.
  - Amplifier
  - Mixer
  - Oscillator

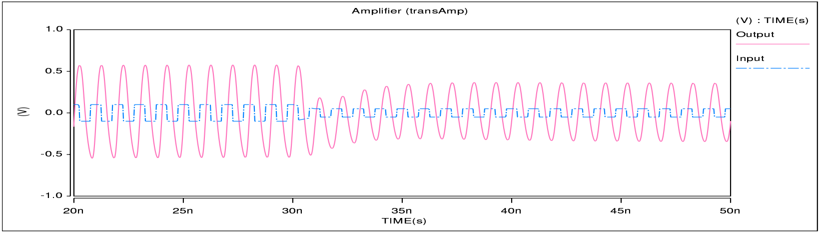
Example — Amplifier



Example — Amplifier

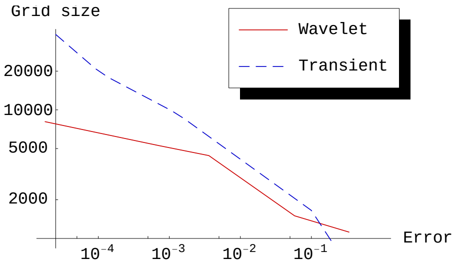


Input and output signal

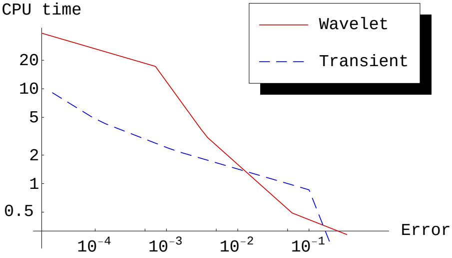


Detail

Example — Amplifier

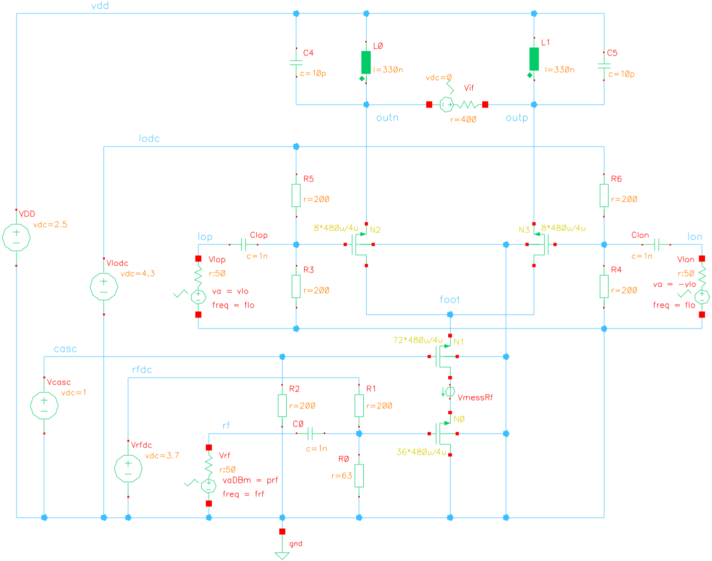


Grid size versus error

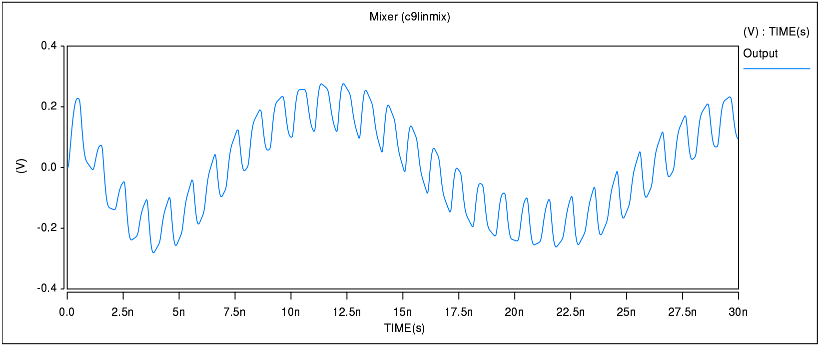


CPU time versus error

Example — Mixer

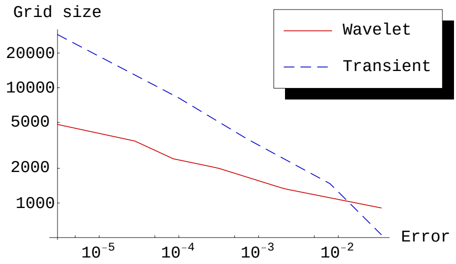


Example — Mixer

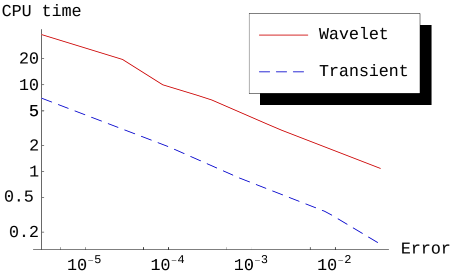


Output for $f_{rf} = 950\text{MHz}$ and $f_{lo} = 1\text{GHz}$

Example — Mixer

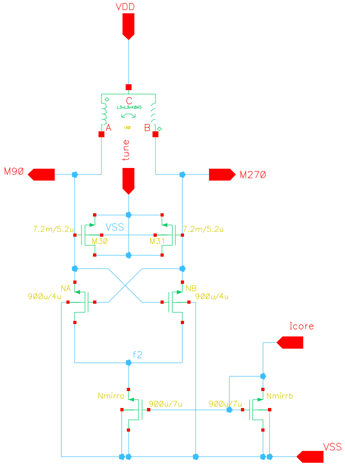


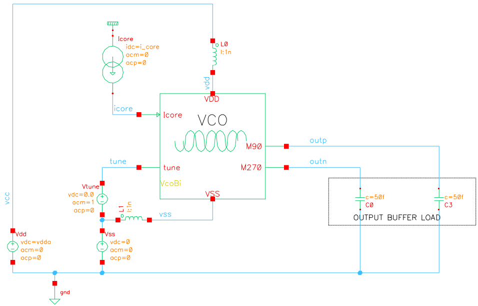
Grid size versus error



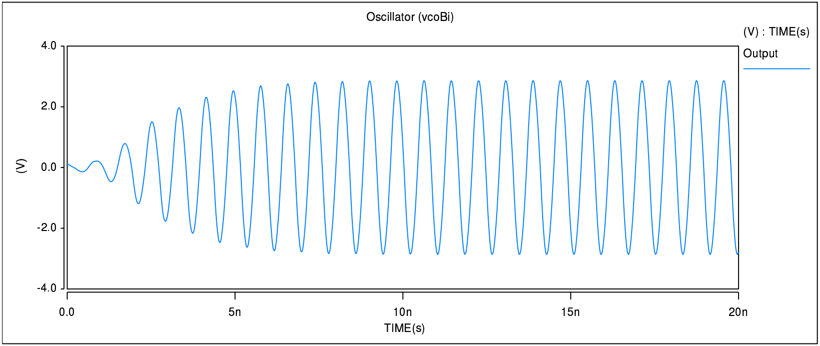
CPU time versus error

Example — Oscillator



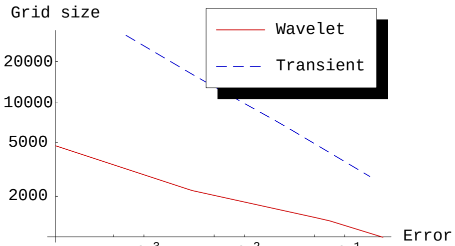


Example — Oscillator

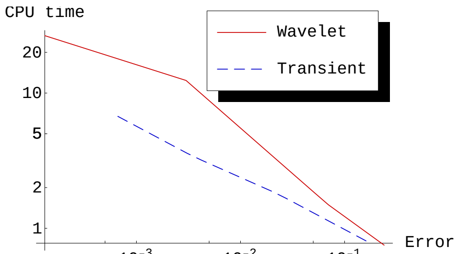


Output

Example — Oscillator



Grid size versus error



CPU time versus error

Conclusions

- New wavelet method developed and implemented

Conclusions

- New wavelet method developed and implemented
- Our tests showed that a wavelet algorithm can reach the performance of traditional methods

Conclusions

- New wavelet method developed and implemented
- Our tests showed that a wavelet algorithm can reach the performance of traditional methods
- $\Rightarrow$ Potential for productive exploitation

Conclusions

- New wavelet method developed and implemented
- Our tests showed that a wavelet algorithm can reach the performance of traditional methods
- $\Rightarrow$ Potential for productive exploitation
- Further optimization needed before released to end-user

Thank you

Thank you for your attention!

Questions?