1. Abstract

SPICE-accurate simulation of present-day large-scale nonlinear integrated circuit (IC) systems with millions of linear/nonlinear components can be prohibitively expensive, and thus extremely challenging. In this paper, we present a novel support-circuit preconditioning (SCP) technique for tackling large-scale nonlinear circuit simulations by exploiting sparsified graphs of a given circuit network. By extracting support graphs (SGs) from the original linear circuit networks, and combining them with nonlinear devices, support-circuit preconditioners can be efficiently computed using existing matrix solvers, allowing for on-the-fly updates during transient simulations when adopted in Krylov-subspace iterative solvers. Experimental results for a variety of large-scale circuit designs show that the proposed method achieves up to 22X speedups in solving the matrices involved in DC and transient (TR) simulations, and up to 8X reduction in memory usage, when compared with the simulator powered by the state-of-the-art direct solver KLU.

2. Motivation and Challenges

- **Motivation**
  - Integrated circuit (IC) system that involves billions of transistors and interconnect components needs to be accurately modeled and analyzed.

- **Challenges in large-scale nonlinear circuit simulation**
  - Applying direct solver is not runtime or memory efficient during system matrix factorization.
  - Finding good and reliable preconditioners for fast iterative solver is very difficult.

3. Problem Formulation and Numerical Method

- **Problem formulation**
  - Nonlinear differential equations
  - $f$ and $q$ denote the static and dynamic nonlinearities, respectively
  \[
  F(x) = f(x(t)) + \int q(x(t)) \, dt = 0
  \]

- **General techniques rely on Newton-Raphson (NR) method**
  - Linearize the nonlinear devices (transistors, etc)
  - Iteratively approach the final solution
  \[
  x^{k+1} = x^k - \left(\frac{\partial F(x)}{\partial x}\right)^{-1} F(x)
  \]

4. Support-Circuit Preconditioner (SCP)

- **Support circuit construction**
  - STEP 1. Input the circuit netlist
  - STEP 2. Extract the interconnect networks
  - STEP 3. Compute the support graphs
  - STEP 4. Combine support graphs with nonlinear devices

- **How to find a better support graph**
  1. Spanning Tree Support Graph
     - Convergence of support graph preconditioner
       - The convergence is determined by the condition number of matrix $L P$
     - The support of preconditioner is defined as: $G_k^P = \text{min}\{ |(G_{k+1}P)_{ij} = 0, all\}$
  2. Support Graph Approximation
     - A weighted graph $G \text{–} P$ approximates a weighted graph $A$
     - $(G_{(k+1)P})_{ij} = \frac{1}{\text{max}\{ |(G_{k+1}P)_{ij} = 0, all\}}$
     - Better approximation
       - In linear network, $L_{(k+1)P}$ always approximates the power distribution of the network.
       - A spanning tree of a graph only retains $n-1$ edges, thus the power distribution on the support graph may be more intuitive than the original one.
       - If a preconditioner can approximate not only the eigenvalues but also the power distribution of the original system, the preconditioner can be more effective.

5. 5.C.5-based GMRES Solver

- **Adaptive preconditioner updating**
  - To further improve the runtime efficiency
    - System Jacobian matrices between two NR steps may not change too much
    - The previous preconditioner can be reused such that the matrix factorization can be saved
    - General techniques rely on Newton
      - Motivation
        - Adult nodes of the original network
        - Nonlinear circuit simulation
      - Motivation
        - Adult nodes of the original network
        - Nonlinear circuit simulation
      - Motivation
        - Adult nodes of the original network
        - Nonlinear circuit simulation

6. Experimental Results

- **Performance comparison for matrix factorization**
  - Ultra-compact power dissipation
  - Motivation
    - Adult nodes of the original network
    - Nonlinear circuit simulation
    - Motivation
    - Adult nodes of the original network
    - Nonlinear circuit simulation
    - Motivation
    - Adult nodes of the original network
    - Nonlinear circuit simulation

7. Conclusion and Future Work

1. **Conclusion**
   - **Conclusion**
     - Extracting ultra-sparsifier support graphs from the linear networks
     - Combine them with the nonlinear analog or digital circuit blocks
     - Our experimental results show that SCP can attain:
       - Up to 22X speedups in DC and transient simulations
       - Reduce up to 80% memory consumption

2. **Future Work**
   - **Future Work**
     - Construct the support circuit based on linearized circuit instead of relying on the topology of the circuit
     - General support-circuit preconditioners

8. Notes

- **Notes**
  - Number of unknowns
  - Number of resistors
  - Number of capacitors
  - Memory
  - Number of current sources
  - Number of current sources
  - Origin: original system
  - Tree: spanning tree support graph
  - Ultra: ultra-sparse support graph
  - Legend: SCP: Support Circuit Preconditioner
  - A: Accurate
  - Non-ADC: Non-Approximate DC
  - ADC: Approximate DC
  - Note: A: Accurate
  - Non-ADC: Non-Approximate DC
  - ADC: Approximate DC
  - Note: A: Accurate
  - Non-ADC: Non-Approximate DC
  - ADC: Approximate DC

- **Motivation**
  - Integrated circuit (IC) system that involves billions of transistors and interconnect components needs to be accurately modeled and analyzed.

- **Challenges in large-scale nonlinear circuit simulation**
  - Applying direct solver is not runtime or memory efficient during system matrix factorization.
  - Finding good and reliable preconditioners for fast iterative solver is very difficult.

- **Performance comparison for matrix factorization**
  - Ultra-compact power dissipation
  - Motivation
    - Adult nodes of the original network
    - Nonlinear circuit simulation
    - Motivation
    - Adult nodes of the original network
    - Nonlinear circuit simulation
    - Motivation
    - Adult nodes of the original network
    - Nonlinear circuit simulation

- **Experimental Setup**
  - Motivation
    - Adult nodes of the original network
    - Nonlinear circuit simulation
    - Motivation
    - Adult nodes of the original network
    - Nonlinear circuit simulation
    - Motivation
    - Adult nodes of the original network
    - Nonlinear circuit simulation

- **Performance comparison for transient simulation**
  - Ultra-compact power dissipation
  - Motivation
    - Adult nodes of the original network
    - Nonlinear circuit simulation
    - Motivation
    - Adult nodes of the original network
    - Nonlinear circuit simulation
    - Motivation
    - Adult nodes of the original network
    - Nonlinear circuit simulation

- **Notes**
  - Number of unknowns
  - Number of resistors
  - Number of capacitors
  - Memory
  - Number of current sources
  - Number of current sources
  - Origin: original system
  - Tree: spanning tree support graph
  - Ultra: ultra-sparse support graph
  - Legend: SCP: Support Circuit Preconditioner
  - A: Accurate
  - Non-ADC: Non-Approximate DC
  - ADC: Approximate DC
  - Note: A: Accurate
  - Non-ADC: Non-Approximate DC
  - ADC: Approximate DC

- **Notes**
  - Number of unknowns
  - Number of resistors
  - Number of capacitors
  - Memory
  - Number of current sources
  - Number of current sources
  - Origin: original system
  - Tree: spanning tree support graph
  - Ultra: ultra-sparse support graph
  - Legend: SCP: Support Circuit Preconditioner
  - A: Accurate
  - Non-ADC: Non-Approximate DC
  - ADC: Approximate DC
  - Note: A: Accurate
  - Non-ADC: Non-Approximate DC
  - ADC: Approximate DC