SAND2018-0384C

Using the Basker Linear Solvers in Xyce

Heidi K. Thornquist, Ting Mei,
Siva Rajamanickam, and Nathan Ellingwood

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. SAND NO. 2011-XXXXP

Xyce

PARALLEL ELECTRONIC SIMULATOR

- Originally designed to be a massively parallel “transistor-level” circuit simulator
  - Circa 1999 ➔ distributed memory parallelism (MPI)
  - Supports NW-specific circuit analysis
  - Enables full system parallel transient simulation for large integrated circuits
- Can also simulate networks of biological / neural / power grid devices

$$f(x(t)) + \frac{dq(x(t))}{dt} = b(t)$$

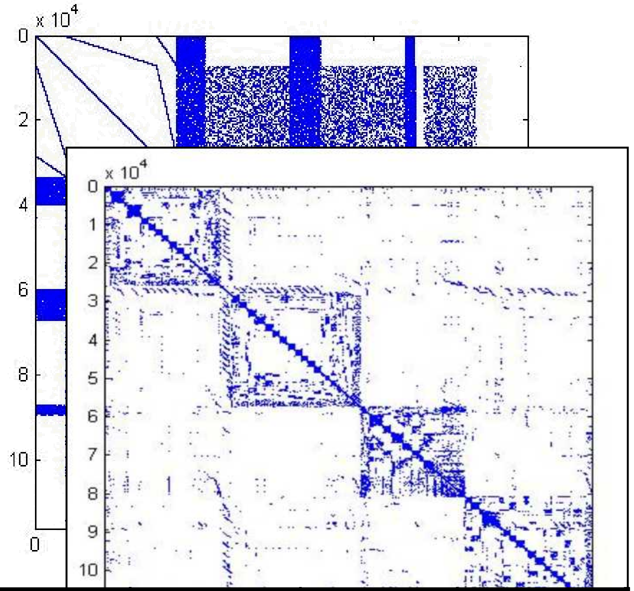
- Defining characteristics that effect performance:
  - Hierarchical structure rather than spatial topology (ODEs vs. PDEs)
  - Densely connected nodes (clock signals, power rails, ground rails)
  - Badly scaled nonlinear differential algebraic equations (DAEs)
  - Multiple objectives for load balancing (device evaluation vs. linear solves)
  - Strong scaling and robustness is the key challenge!

Xyce Circuit Analysis

- Time-domain simulation was the original focus of Xyce
  - *Motivated by stockpile stewardship efforts (ALTs/LEPs/surveillance)*
  - DC [steady-state]
  - Transient
    - Challenging because customer circuits have diversity in scale
- Frequency-domain simulation has become an increasing focus
  - *Motivated by RF designers, microwave / satellite applications*
  - AC [small signal]
    - linear response of the circuit to stimuli or noise generating elements
  - Harmonic Balance (HB)
    - Analysis of high-frequency nonlinear circuits such as mixers, amplifiers, and oscillators
    - This capability is in high-demand and is challenging to efficiently implement, especially in parallel

Transient Linear Solver Scaling

- Initially (circa 1999), Xyce used available PDE-based preconditioning techniques
  - Incomplete LU factorization on subdomains
  - Limited scaling / robustness
- For small scale circuits, the Dulmage-Mendelsohn permutation (BTF) was leveraged in KLU (2004)
- In 2008, BTF structure was leveraged to create a new preconditioned iterative method
  - Great for CMOS memory circuits
  - Circuits with parasitics are more challenging
- In 2010, ShyLU was developed in Trilinos
  - First "hybrid-hybrid" linear solver in Xyce
  - Improve robustness
  - Scaling limitations



Preconditioning Method	Residual	GMRES Iters	Solver Time (s.)
Local AMD ILUT ParMETIS	3.43e-01 (FAIL)	500	302.573
BTF Block Jacobi Hypergraph	3.47e-10	3	0.139

Harmonic Balance in Xyce

- Harmonic Balance analysis = Newton's method in the frequency-domain
- However, device models are evaluated in the time domain
  - This has been reduced to nonlinear device models in FY16 for efficiency
- Implementation requires transformation of time-domain DAE

$$\frac{dq(x)}{dt} + f(x) = b(t),$$

using Fourier series expansions for the input and solution variables

$$x(t) = \sum_{i=-\infty}^{\infty} X_i e^{ji\frac{2\pi}{T}t}, \quad b(t) = \sum_{i=-\infty}^{\infty} B_i e^{ji\frac{2\pi}{T}t}$$

after substitution and truncation, the frequency-domain system is

$$H_{HB} = \Omega Q(X) + F(X) - B = 0 \qquad \mathbf{\Omega} = \begin{pmatrix} -Mj\omega_0 & & \\ & \ddots & \\ & & Mj\omega_0 \end{pmatrix}, \omega_0 = \frac{2\pi}{T}$$

- But, we can't evaluate the device models in the frequency-domain …

Harmonic Balance Jacobian

- So, expanding the frequency-domain equations using the discrete Fourier transform matrices, D and D^{-1}.

$$\mathbf{H_{HB}} = \Omega D \mathbb{Q}(.) D^{-1} X + D \mathbb{F}(.) D^{-1} X - B$$

$$\mathbf{J_{HB}} = \Omega D \mathbb{C} D^{-1} + D \mathbb{G} D^{-1},$$

$$\mathbb{C} = \begin{pmatrix} C(t_0) & & \\ & \ddots & \\ & & C(t_{2M}) \end{pmatrix}, C(t_i) = \frac{dq}{dx}|_{x(t_i)} \qquad \mathbb{G} = \begin{pmatrix} G(t_0) & & \\ & \ddots & \\ & & G(t_{2M}) \end{pmatrix}, G(t_i) = \frac{df}{dx}|_{x(t_i)}$$

- Linear solvers for the HB Jacobian matrix are challenging
  - Large matrix (n x N) with block structure, complex-valued, possibly dense
  - Iterative methods are most often used because they avoid matrix construction
    - For highly-nonlinear circuits effective preconditioners are difficult to construct
  - Convergence failure for HB is often blamed on the linear solver …

Basker HB Linear Solver

- In FY17, a new direct solver was developed in Xyce, using the templated Basker direct solver in Amesos2
  - Enables the use of “block matrices” as a scalar type
- Motivated by the Berkeley Design Automation (BDA) direct solver

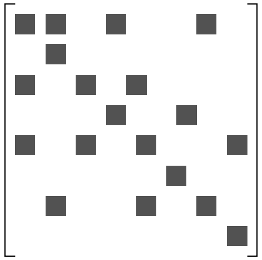


Figure 1: Sparsity structure of HB Jacobian

  - *BDA Observation:* Block structure of HB Jacobian is dependent upon ordering
  - *Our Observation*: Dense blocks are created when voltage nodes are connected to nonlinear devices, all other nodes create diagonal blocks (easy!)
  - As of FY16, Xyce separates linear and nonlinear devices and variables, so we have a conservative estimate of the dense blocks in the HB Jacobian

ASC Weekly News Notes

HB Solver Enhancement in Xyce.

Brief: A sparse-direct linear solver is now available for performing Harmonic Balance (HB) simulation in Xyce, providing a more robust set of solver capabilities.

A sparse-direct linear solver is now available for performing Harmonic Balance (HB) simulation in Xyce , Sandia's high-performance circuit simulator. Previously, only iterative linear solvers were available for performing the steady state solve using an unassembled HB Jacobian. Iterative solvers were prone to failure for highly nonlinear circuits and bad initial conditions because of ineffective preconditioners; direct solvers provide a more robust set of solver capabilities for highly nonlinear circuits. The new direct solver assembles the HB Jacobian, after which it is sent either to LAPACK (dense solver approach), or to the templated Basker solver that is provided by Trilinos. Two approaches are taken with the Basker solver, using a block form (which is the Berkeley Design Automation approach), or in a non-block form. The block form currently fails for some nonlinear circuits, but a likely source of the bug has been identified for future work. This capability provides a robust linear solver approach for HB, but is not scalable. Furthermore, it lays the groundwork for improvements to Xyce's multi-tone HB capability that is useful for microwave, electromagnetic (EM), and Trust-based NW applications. (POC: Heidi Thornquist, hkthorn@sandia.gov)

New HB Solver Impact

- Yes, I’m excited about a serial direct solver …
  - For mission-critical impact, robustness is just as worthy of a goal as scalability
- This has enabled us to address convergence issues other than linear solver failure

Belos with block Jacobi preconditioner

```
***** Solution Summary *****
        Number Successful Steps Taken:          430
        Number Failed Steps Attempted:          11
        Number Jacobians Evaluated:             1777
        Number Iteration Matrix Factorizations: 0
        Number Linear Solves:                   690
        Number Failed Linear Solves:            0
        Number Linear Solver Iterations:        1078
        Number Residual Evaluations:            1131
        Number Nonlinear Convergence Failures:  0
        Total Residual Load Time:               2.2480e-02 seconds
        Total Jacobian Load Time:               8.0790e-01 seconds
        Total Linear Solution Time:             1.8122e+00 seconds
```

Basker HB solver

```
***** Solution Summary *****
        Number Successful Steps Taken:          430
        Number Failed Steps Attempted:          11
        Number Jacobians Evaluated:             683
        Number Iteration Matrix Factorizations: 0
        Number Linear Solves:                   690
        Number Failed Linear Solves:            0
        Number Residual Evaluations:            1131
        Number Nonlinear Convergence Failures:  0
        Total Residual Load Time:               2.4401e-02 seconds
        Total Jacobian Load Time:               2.2535e-03 seconds
        Total Linear Solution Time:             7.9714e-01 seconds
```

- Looking forward, this new solver provides a jumping off point
  - Multi-threaded improvements to current Basker HB solver
  - Use Basker HB solver on each node in parallel
  - Algorithmic interpolation between Basker and block-Jacobi