Linear Solver Challenges in Large-Scale Circuit Simulation

MUMPS User Group Meeting

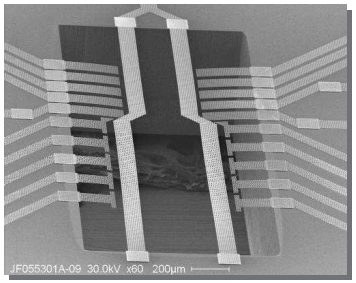
April 16th, 2010

Heidi K. Thornquist

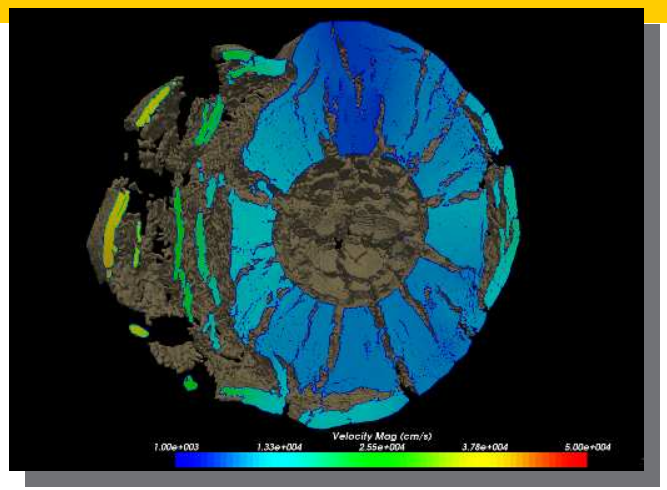
Sandia National Laboratories
Albuquerque, NM

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

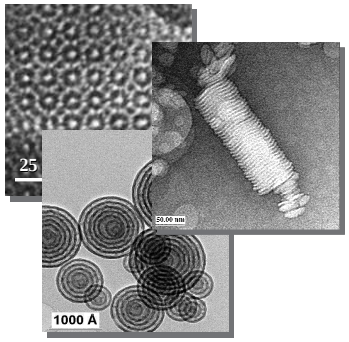
SNL has six core technical capabilities



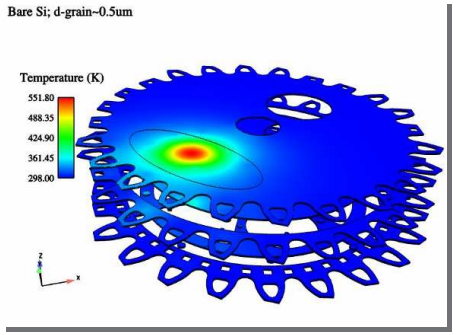
Microelectronics and Photonics



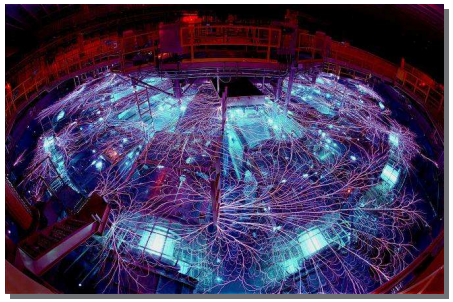
Computational & Informational Sciences



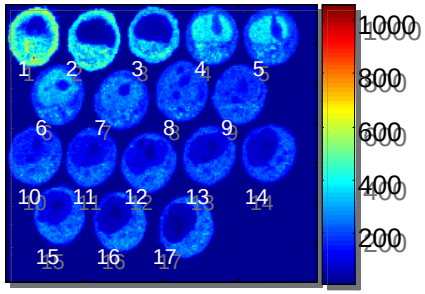
Materials Science & Technology



Engineering Sciences



Pulsed Power



Bioscience

ASC

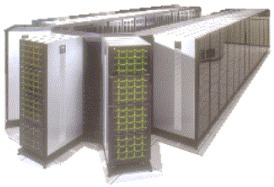
Sandia National Laboratories

CIS has a rich history in the development and maturation of high performance computing hardware and software technology


CM-2


nCUBE-2


iPSC-860


Paragon


ASCI Red


Cplant


Red Storm

1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007

Gordon Bell Prize

R&D 100
Parallel Software

Patent
Meshing

R&D 100
Dense Solvers

R&D 100
Storage

Gordon Bell Prize

World Record
Teraflops

Gordon Bell Prize

R&D 100
Allocator

R&D 100
Trilinos

R&D 100
Xyce

Gordon Bell Prize

R&D 100
Signal Processing

SC96 Gold Medal
Networking

Mannheim
SuParCup

Karp Challenge

World Record
281 GFlops

R&D 100
Aztec

Patent
Data Mining

R&D 100
3D-Touch

R&D 100
Meshing

World Record
143 GFlops

R&D 100
Salvo

Patent
Parallel Software

Patent
Paving

Patent
Decomposition

Fernbach
Award

Xyce Motivation

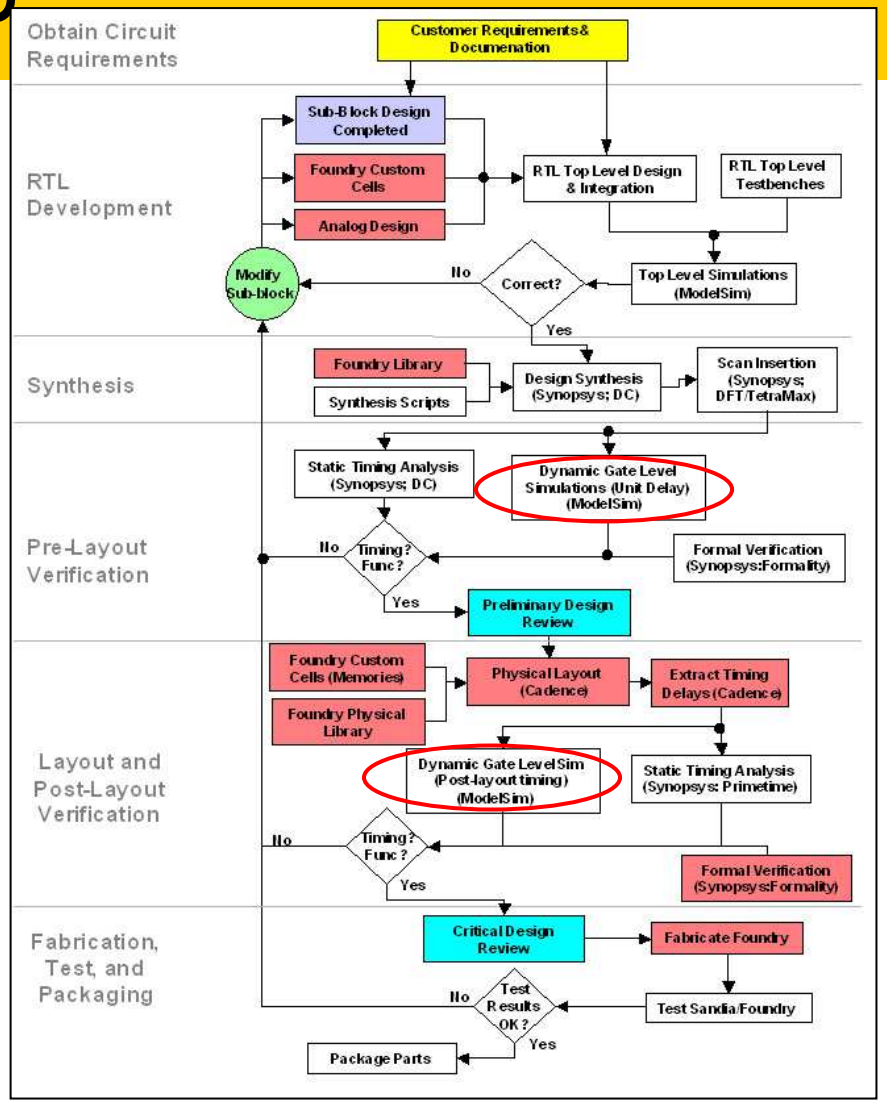
- Lack of NW testing:
  - Comprehensive Test Ban Treaty (CTBT), 1993
  - Advanced Simulation & Computing (ASC), 1995
  - Qualification Alternatives to SPR (QASPR), 2005

- Unique Requirements ➡ Differentiating capabilities
  - Full system simulation
  - Unique models: Radiation Effects
  - High fidelity: "true SPICE" level or higher
  - Large capacity: Massively-parallel

- IP: Sandia owns it, source-level access
- Commercial Tools are expensive ➡ $5K-$1M

Circuit Design Process

- Highly complex
  - Requires different tools for verifying different aspects of the circuit
- Cannot afford many circuit re-spins
  - Expense of redesign
  - Time to market
- Accurate / efficient / robust tools
  - Challenging for 45nm technology



Analog circuit simulator (SPICE compatible)

- Large scale (N>1e7) “flat” circuit simulation
  - solves set of coupled DAEs simultaneously
- Distributed memory parallel
  - threaded device loads
- Advanced solution techniques
  - Homotopy
  - Multi-level Formulation
  - Multi-time Partial Differential Equation (MPDE)
  - Parallel Iterative Matrix Solvers / Preconditioners
- 2008 R&D100 Award

Outline

- Simulation Challenges
  - Network Connectivity
  - Load Balancing / Partitioning
  - Efficient Parallel Linear Solvers
- Xyce & Trilinos
- Linear Solver Strategies
  - Results

Parallel Circuit Simulation Challenges

Analog simulation models network(s) of devices coupled via Kirchoff's current and voltage laws

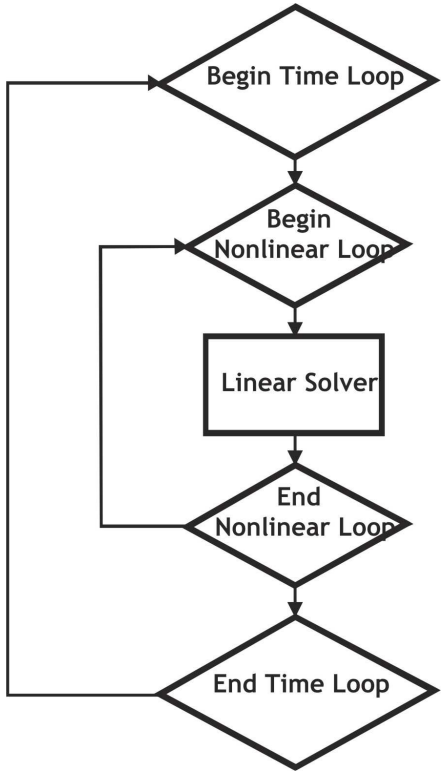
$$f(x(t)) + \frac{dq(x(t))}{dt} = b(t)$$

- Network Connectivity
  - Hierarchical structure rather than spatial topology
  - Densely connected nodes: O(n)
- Badly Scaled DAEs
  - Compact models designed by engineers, not numerical analysts!
  - Steady-state (DCOP) matrices are often ill-conditioned
- Non-Symmetric
  - Not elliptic and/or globally SPD
- Load Balancing / Partitioning
  - Balancing cost of loading Jacobian values unrelated to matrix partitioning for solves

Parallel Circuit Simulation Structure

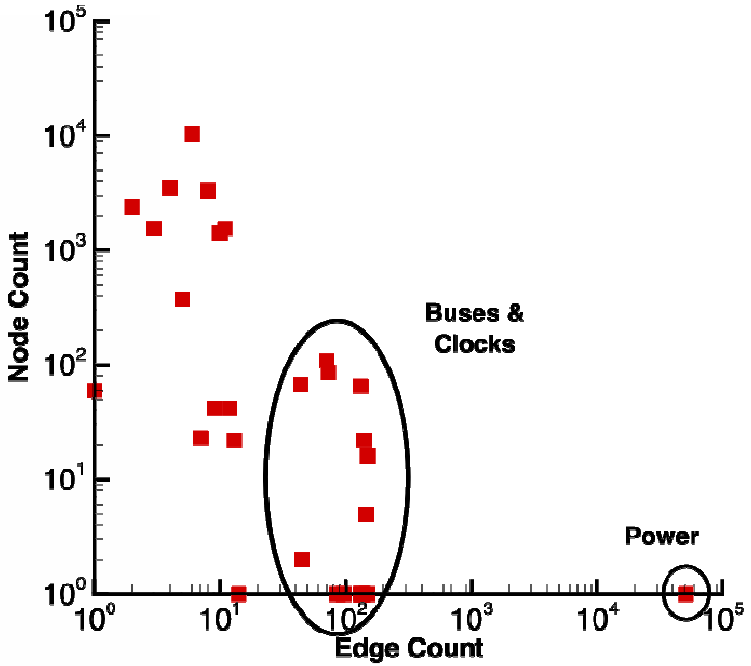
(Transient Simulation)

- Simulation challenges create problems for linear solver
  - Direct solvers more robust
  - Iterative solvers have potential for better scalability
- Iterative solvers have previously been declared unusable for circuit simulation
  - Black box methods **do not** work!
  - Creation of effective preconditioner most important
- Leverage useful structure
  - Static graph
  - Highly connected nodes
  - Unidirectionality
  - Recycling solvers



Network Connectivity

(Singleton Removal)



Row Singleton: pre-process

$$\begin{bmatrix} & & a_{1j} & & \\ & & a_{2j} & & \\ & & \vdots & & \\ & & \vdots & & \\ 0 & \cdots & a_{ij} & \cdots & 0 \\ & & \vdots & & \\ & & a_{nj} & & \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ x_j \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

$$\Rightarrow x_j = b_i / a_{ij}$$

Column Singleton: post-process

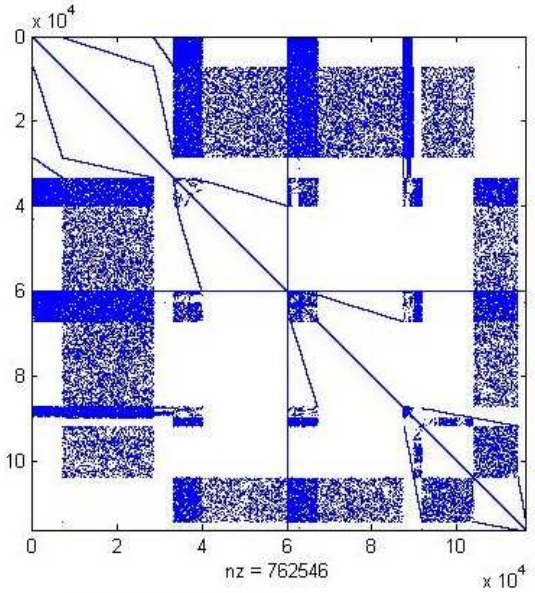
$$\begin{bmatrix} & & 0 & & \\ & & 0 & & \\ & & \vdots & & \\ & & \vdots & & \\ a_{i1} & \cdots & a_{ij} & \cdots & a_{in} \\ & & \vdots & & \\ & & 0 & & \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ x_j \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

$$\Rightarrow x_j = \left(b_i - \sum_{k \neq j} a_{ik} x_k \right) / a_{ij}$$

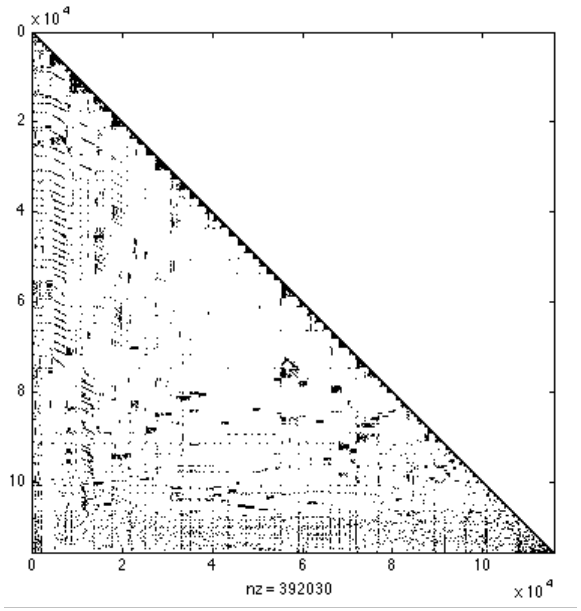
- Connectivity:
  - Most nodes very low connectivity -> sparse matrix
  - Power node generates very dense row (~0.9*N)
  - Bus lines and clock paths generate order of magnitude increases in bandwidth

Network Connectivity

(Hierarchical Structure)



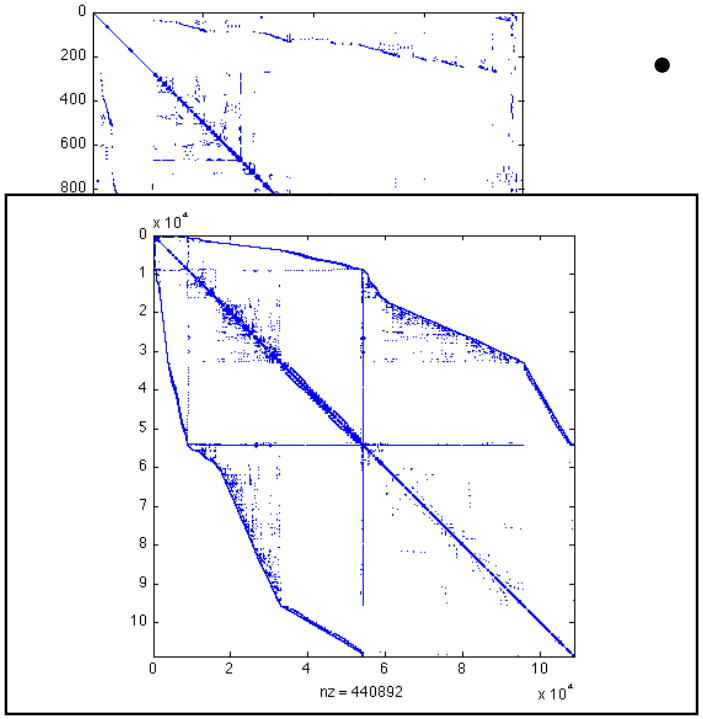
- Heterogeneous Matrix Structure
- Some circuits exhibit *unidirectionality*:
  - Common in CMOS Memory circuits
  - Not present in circuits with feedback (e.g. PLLs)
  - Block Triangular Form (BTF) via Dulmage-Mendelsohn permutation



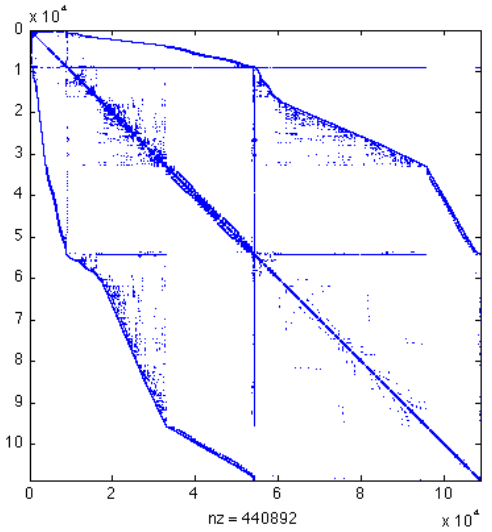
- BTF benefits both direct and preconditioned iterative methods
- Used by Tim Davis's KLU in Trilinos/AMESOS (The "Clark Kent" of Direct Solvers)

Network Connectivity

(Parasitics/PLLs)



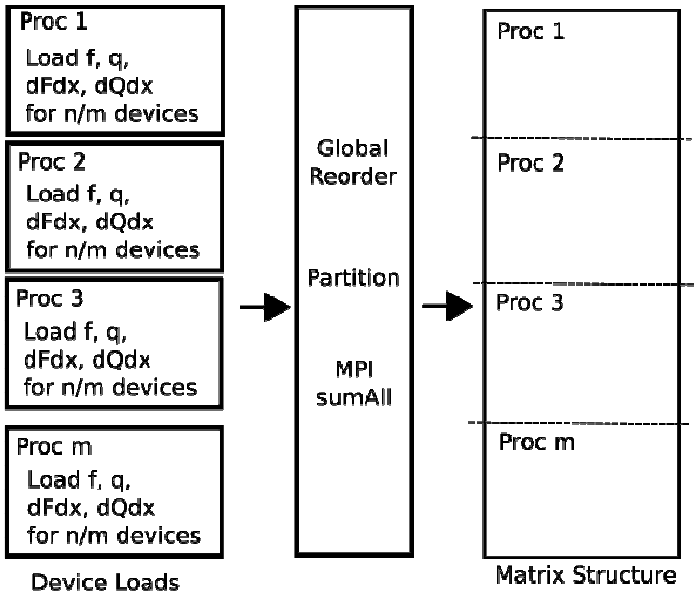
- Other circuits **do not** exhibit *unidirectionality*:
  - Common in phase-locked loops (PLLs)
  - Common in post-layout circuits
    - circuits with parasitics
    - important for design verification
    - often **much** larger than original circuit



- Dulmage-Mendelsohn permutation results in large irreducible block

Load Balancing / Partitioning

- Balancing Jacobian loads with matrix partitioning for iterative solvers
  - Use different partitioning for Jacobian loads and solves
  - Simple distribution of devices across processors
- Matrix partitioning more challenging:
  - Graph
    - Assumes symmetric structure
    - Robust software available (ParMETIS, etc.)
  - Hypergraph
    - Works on rectangular, non-symmetric matrices
    - Newer algorithms (Zoltan, etc.)
    - More expensive to compute
    - More accurately measures communication volume



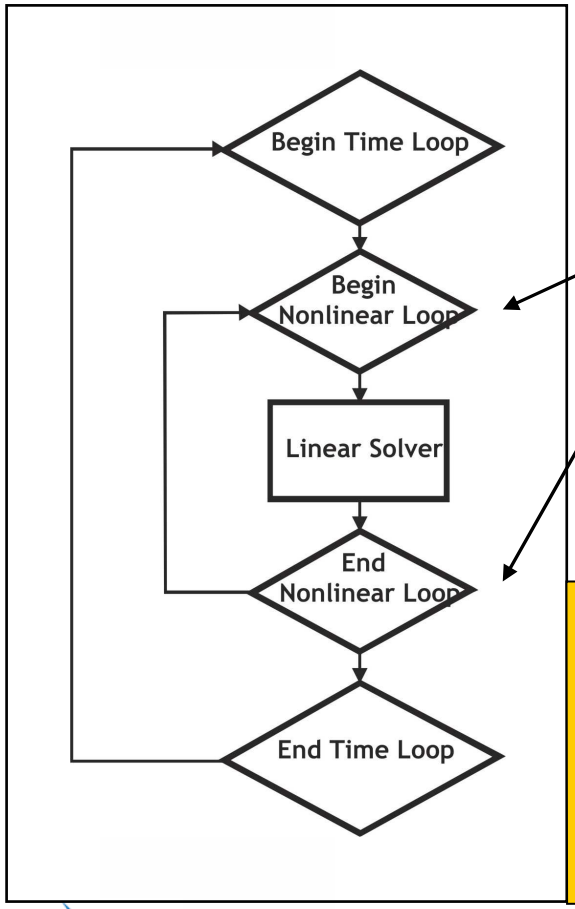
- Trilinos is an evolving framework to support large-scale simulation codes:
  - Fundamental atomic unit is a *package*
  - Includes core set of vector, graph and matrix classes (Epetra/Tpetra packages)
  - Provides a common abstract solver API (Thyra package)
  - Provides a ready-made package infrastructure:
    - Source code management (git, gitk)
    - Build tools (cmake)
    - Automated regression testing (ctest / cdash)
    - Communication tools (mailman mail lists)
  - Specifies requirements and suggested practices to address ASC SQA/SQE requirements
- Trilinos allows the separation of efforts:
  - Efforts best done at the Trilinos level (useful to most or all packages)
  - Efforts best done at a package level (peculiar or important to a package)
  - **Allows package developers to focus only on things that are unique to their package**

Trilinos Package Summary

	Objective	Package(s)
Discretizations	Spatial Discretizations (FEM,FV,FD)	Intrepid
	Time Integration	Rythmos
Methods	Automatic Differentiation	Sacado
	Mortar Methods	Moertel
Core	Linear algebra objects	Epetra, Jpetra, Tpetra
	Abstract interfaces	Thyra, Stratimikos, RTOp
	Load Balancing	Zoltan, Isorropia
	"Skins"	PyTrilinos, WebTrilinos, Star-P, ForTrilinos
	C++ utilities, (some) I/O	Teuchos, EpetraExt, Kokkos, Triutils
Solvers	Iterative (Krylov) linear solvers	AztecOO, Belos, Komplex
	Direct sparse linear solvers	Amesos
	Direct dense linear solvers	Epetra, Teuchos, Pliris
	Iterative eigenvalue solvers	Anasazi
	ILU-type preconditioners	AztecOO, IFPACK, TIFPACK
	Multilevel preconditioners	ML, CLAPS
	Block preconditioners	Meros
	Nonlinear system solvers	NOX, LOCA
	Optimization (SAND)	MOOCHO, Aristos

Parallel Circuit Simulation Structure

(Transient Simulation)



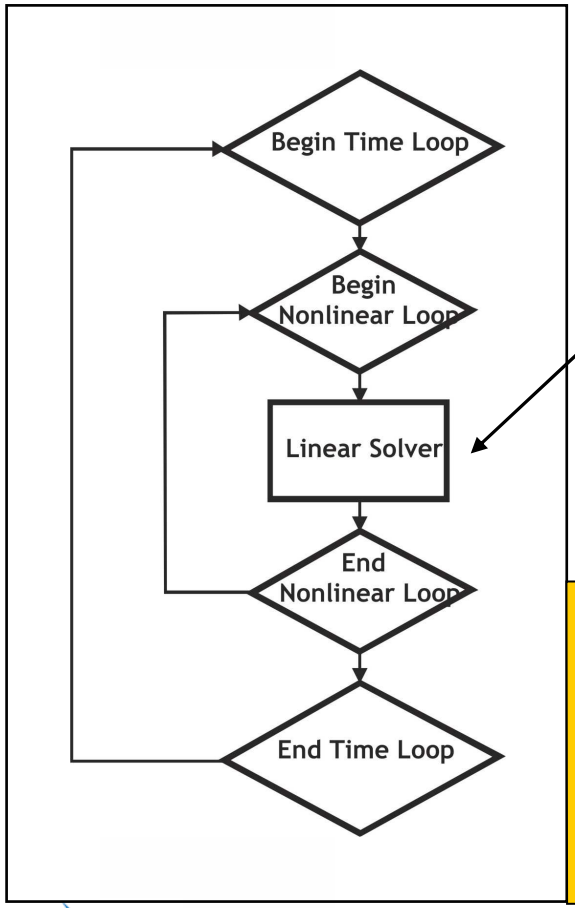
- NOX
  - Suite of nonlinear solution methods
  - Globalizations: Line Search and Trust Region
  - Parallel, OO C++, independent of linear algebra
- LOCA
  - Library of continuation algorithms
  - Zero-order, first-order, arc length, etc.
- Parallel, OO C++, independent of linear algebra

Xyce Interface

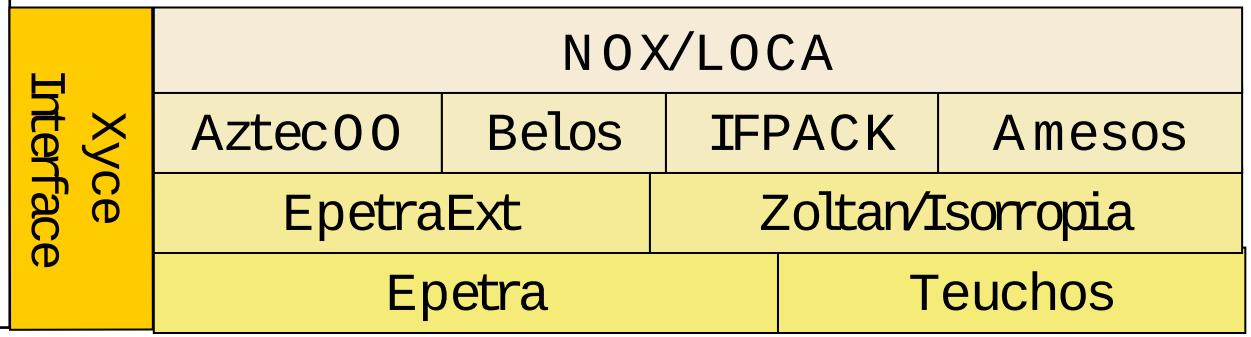
NOX/LOCA			
AztecOO	Belos	IFPACK	Amesos
EpetraExt		Zoltan/Isorropia	
Epetra		Teuchos	

Parallel Circuit Simulation Structure

(Transient Simulation)

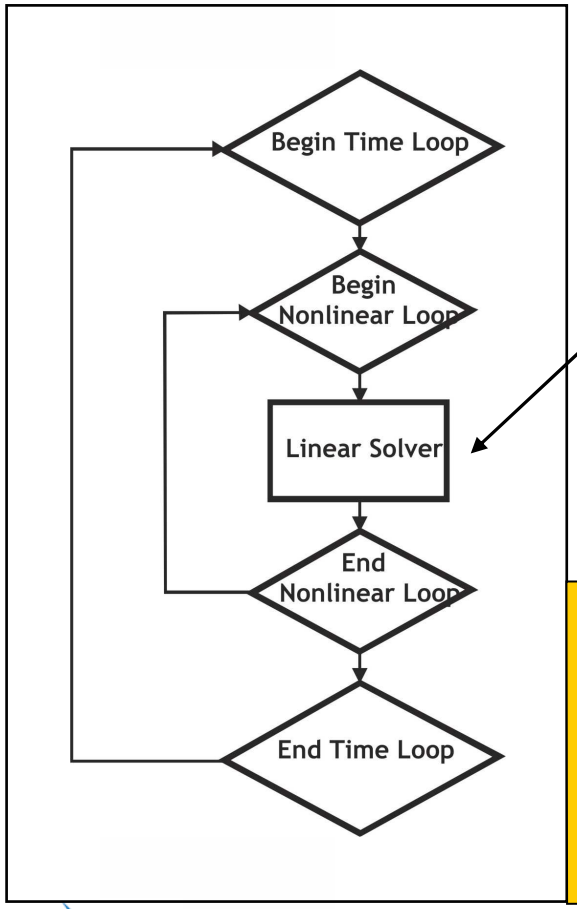


- Amesos
  - Interface to direct solvers for distributed sparse linear systems (KLU, SuperLU, MUMPS, PARDISO, …)
- AztecOO
  - Krylov subspace solvers: CG, GMRES, …
- IFPACK
  - Overlapping Schwarz preconditioners with incomplete factorizations, block relaxations, block direct solves

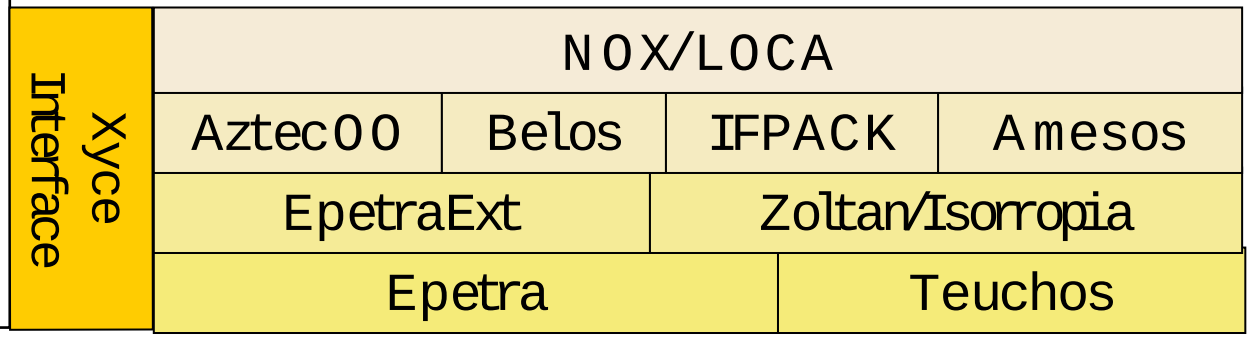


Parallel Circuit Simulation Structure

(Transient Simulation)

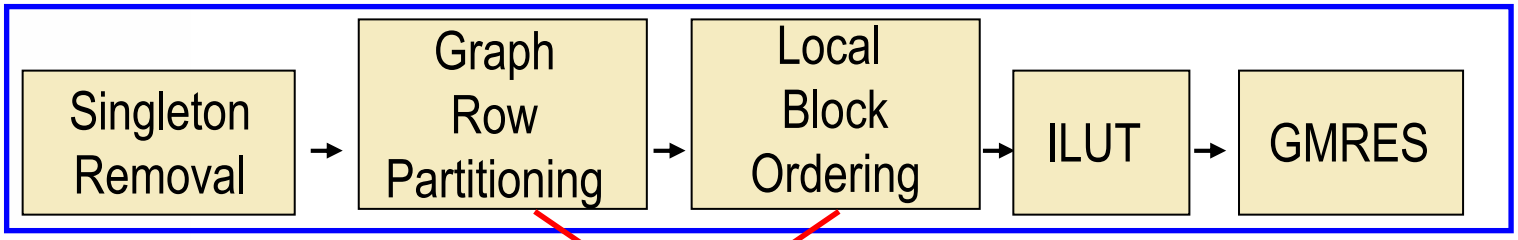


- Epetra
  - Petra provides a “common language” for distributed linear algebra objects (operator, matrix, vector)
  - Restricted to real, double precision arithmetic
  - Uses stable core subset of C++ (circa 2000)
- EpetraExt
  - Extensions to Epetra; linear transformations
- Isorropia
  - Interface from linear algebra objects to partitioners

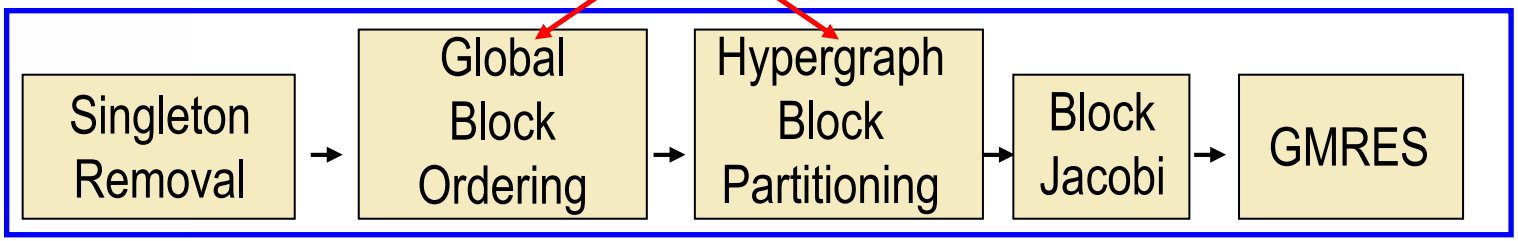


Linear Solver Strategies
~ Iterative and Direct ~

- Strategy 1: (DD)



- Strategy 2: (BTF)



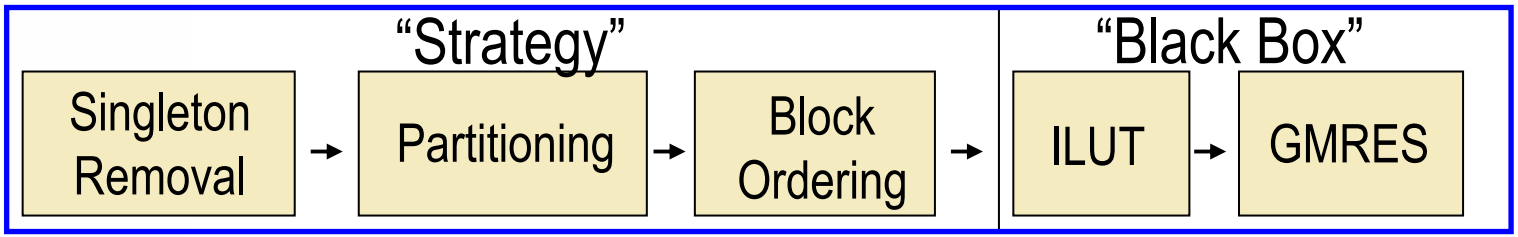
- Direct Solver Strategies:
  - KLU (serial), SuperLU-Dist (parallel)

Linear Solver Strategies
~ Strategy 1 ~

- Assertion:

  Black box methods won't work!

Strategy	Precond	N	Total Cuts	Condition #	GMRES Iters	Solve Time
Black Box	ILUT	1220	~1000	3.00E+05	500	4.7
Strategy 1	SR+Zoltan+ AMD+ILUT	1054	68	1.00E+04	127	0.43

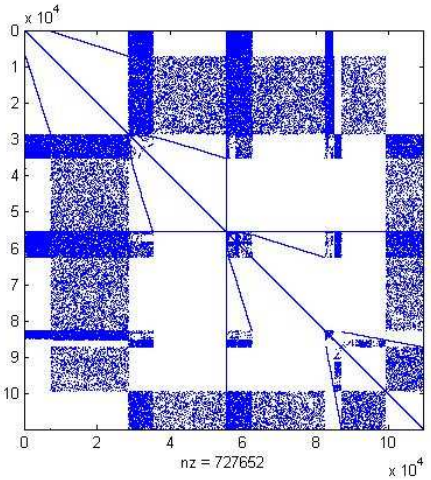


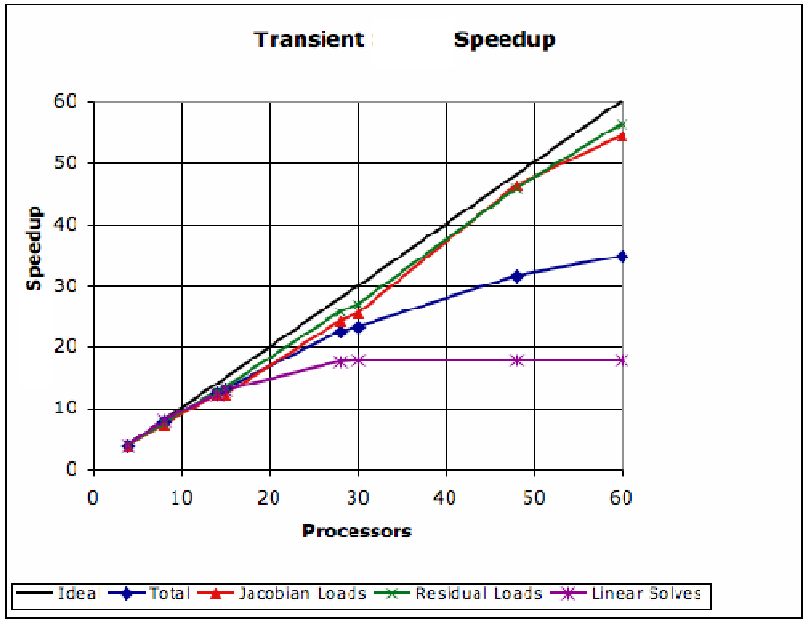
“Strategy 1” Solver Performance

- Assertion:

  Solver strategy is problem dependent!

  - Ex: 100K transistor IC problem



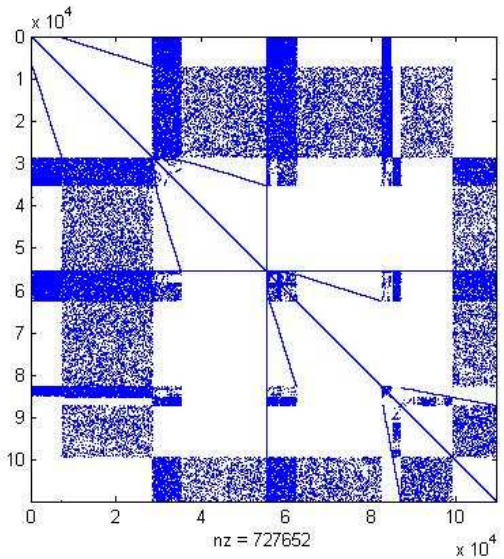


Strategy	Method	Residual	GMRES Iters	Solve Time
1 (4 procs)	SR+Zoltan+ AMD+ILUT	3.425e-01 (FAIL)	500	302.573

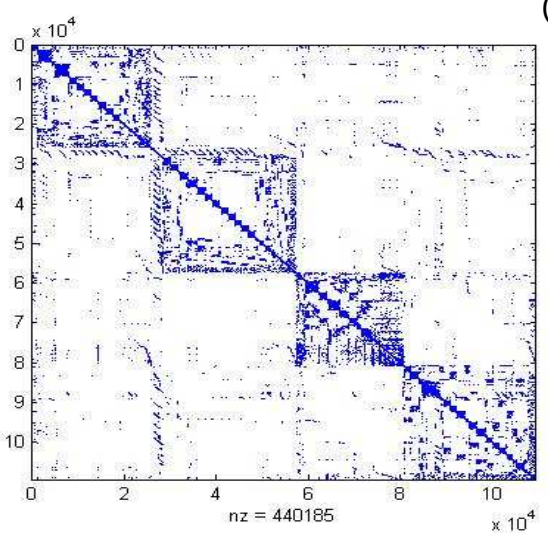
Linear Solver Strategies
~ Strategy 2 ~

Strategy	Method	Residual	GMRES Iters	Solve Time
1	SR+Zoltan+ AMD+ILUT	3.425e-01 (FAIL)	500	302.573
2	SR+BTF+ Hypergraph+KLU	3.473e-10	3	0.139

Original



BTF+Hypergraph
(4 procs)



Test Circuits

Circuit	N	Capacitors	MOSFETs	Resistors	Voltage Sources	Diodes
ckt1	688838	93	222481	175	75	291761
ckt2	434749	161408	61054	276676	12	49986
ckt3	116247	52552	69085	76079	137	0
ckt4	63761	208236	11732	51947	56	0
ckt5	46850	21548	18816	0	21	0
ckt6	32632	156	13880	0	23	0
ckt7	25187	0	71097	0	264	0
ckt8	17788	14274	7454	0	15	0
ckt9	15622	7507	10173	11057	29	0
ckt10	10217	460	4243	1	23	0

Results - 4 Cores

Circuit	Task	KLU (serial)	SLUD	DD	BTF	Speedup (KLU/BTF)
ckt3	Setup	131	56	F2	57	**2.3x**
	Load	741	568	F2	562	**1.3x**
	Solve	6699	2230	F2	255	**26.2x**
	Total	**7983**	**2903**	**F2**	**923**	**8.6x**
ckt4	Setup	552	58	F2	F1	-
	Load	153	44	F2	F1	-
	Solve	106	157	F2	F1	-
	Total	**840**	**274**	**F2**	**F1**	-
ckt10	Setup	3	6	1	F1	-
	Load	606	300	339	F1	-
	Solve	323	3049	2460	F1	-
	Total	**989**	**3381**	**2827**	**F1**	-

F1 = BTF large irreducible block
F2 = Newton convergence failure

Results - 16 Cores

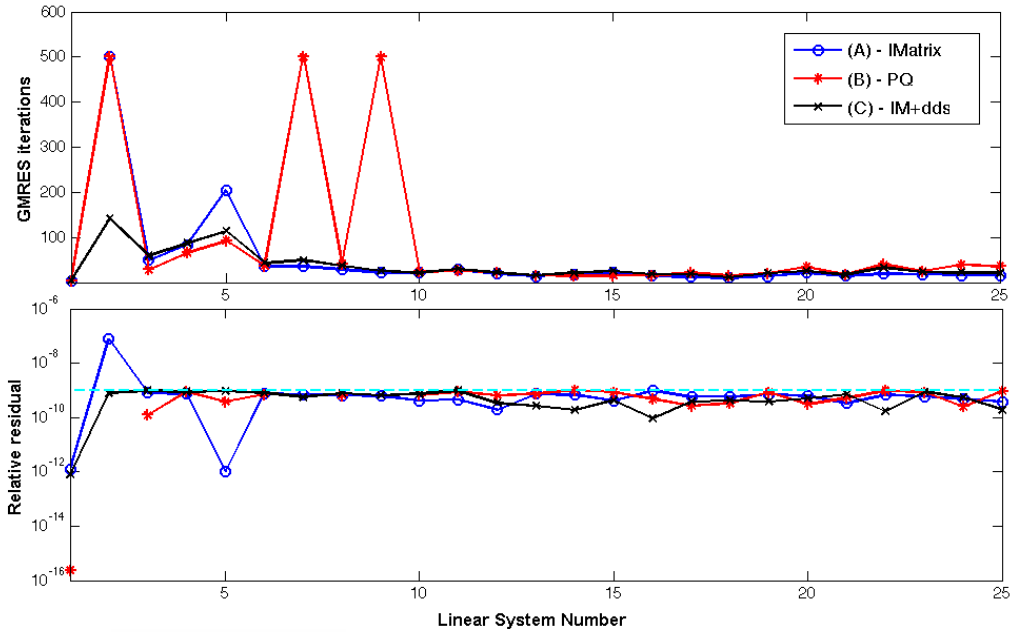
Circuit	Task	KLU (serial)	SLUD	DD	BTF	Speedup (KLU/BTF)
ckt1	Setup	2396	F3	207	199	12.0x
	Load	2063	F3	194	180	11.4x
	Solve	1674	F3	3573	310	5.4x
	Total	**6308**	**F3**	**4001**	**717**	**8.8x**
ckt3	Setup	131	29	F2	29	4.5x
	Load	741	181	F2	175	4.2x
	Solve	6699	1271	F2	84	79.8x
	Total	**7983**	**1470**	**F2**	**306**	**26.1x**
ckt4	Setup	552	32	F2	F1	-
	Load	153	21	F2	F1	-
	Solve	106	133	F2	F1	-
	Total	**840**	**192**	**F2**	**F1**	-

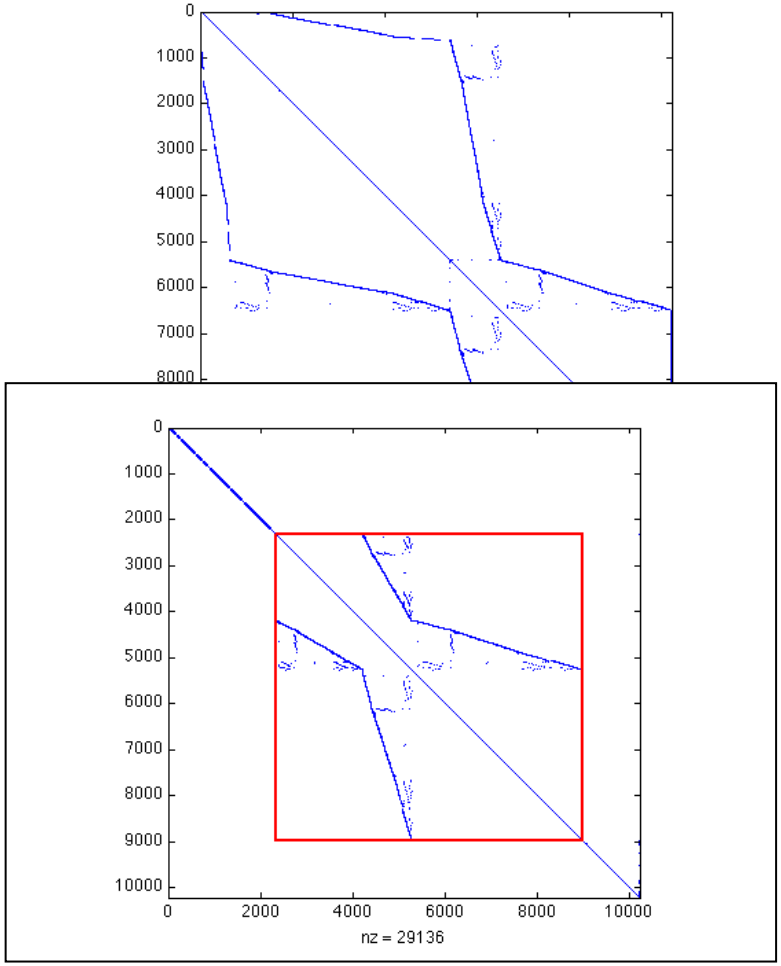
F1 = BTF large nonreducible block
F2 = Newton convergence failure

F3 = Out of memory

Preconditioning Directions:
Multilevel ILU

- ckt10 : CircuitSim90 - Voter circuit
- Needs more efficient preconditioner





[ILU++, Mayer, J.]

Other Solver Strategies

- The SPICE industry standard is Markowitz ordering
  - BTF structure is known but not highly regarded
- Preconditioned iterative methods presented before
  - C. W. Bomhof and H.A. van der Vorst [NLAA, 2000]
    - Requires doubly bordered block diagonal matrix partition
  - A. Basermann, U. Jaekel, and K. Hachiya [SIAM LA 2003 proc.]
    - Requires ParMETIS to give good initial ordering
  - H. Peng and C.K. Cheng [DATE 2009 proc.]
    - Domain decomposition approach, requires knowledge of device boundaries
- Paraklete: parallel KLU [in the works, Tim Davis]

Conclusions

- Iterative linear solvers can enable scalable circuit simulation
  - Dependent upon choosing correct preconditioning strategy
- BTF preconditioning strategy has been successful
  - Great for CMOS memory circuits (ckt3)
  - Performs better than standard strategy on Xyce 680k ASIC (ckt1)
- But it is still not a silver bullet …
  - Circuits with feedback (PLLs) are more challenging (ckt4)
- Multilevel techniques are a positive research direction
  - Can help to more efficiently precondition some circuits with large irreducible blocks (ckt10)

Questions?

- Xyce
  - http://xyce.sandia.gov
- Trilinos
  - http://trilinos.sandia.gov
  - Trilinos User Group (TUG) Meeting 2010