Parallel Sparse Linear Solvers for Circuit Simulation

Achim Basermann, Fabienne Cortial-Goutaudier, Uwe Jaekel

NEC Europe Ltd.
C & C Research Laboratories, Sankt Augustin, Germany

E-mail: {basermann,jaekel,cortial}@ccrl-nece.de

Koutaro Hachiya

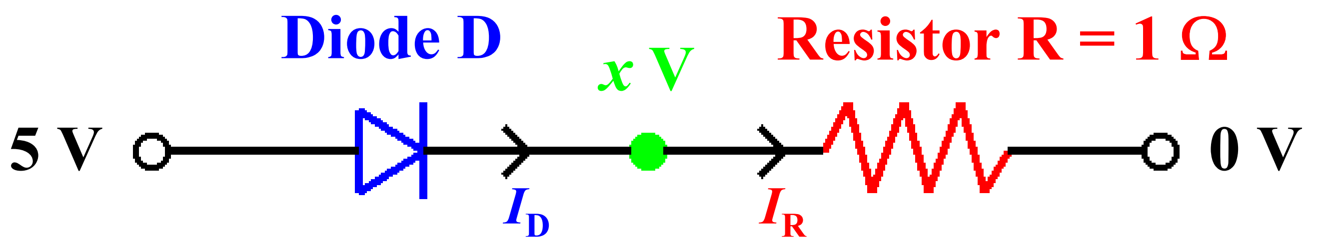
NEC Electron Devices, System LSI Design
Engineering Division, Tamagawa, Japan

E-mail: k-hachiya@ax.ho.nec.com

Survey

1. Introduction to circuit simulation
2. Problem of dense rows and columns
3. Distributed Schur Complement techniques
    a. Definitions
    b. Algorithm
    c. Preconditioning
    d. Repartitioning
    e. Reordering
4. Results
5. Conclusions

Introduction to circuit simulation (1)



$$I_D / \mathrm{A} = e^{100(5-x)} - 1 \qquad I_R / \mathrm{A} = \frac{x-0}{1} = x$$

Kirchoff current laws: $I_D = I_R$ →

$$f(x) = e^{100(5-x)} - 1 - x = 0$$

Nonlinear algebraic equation

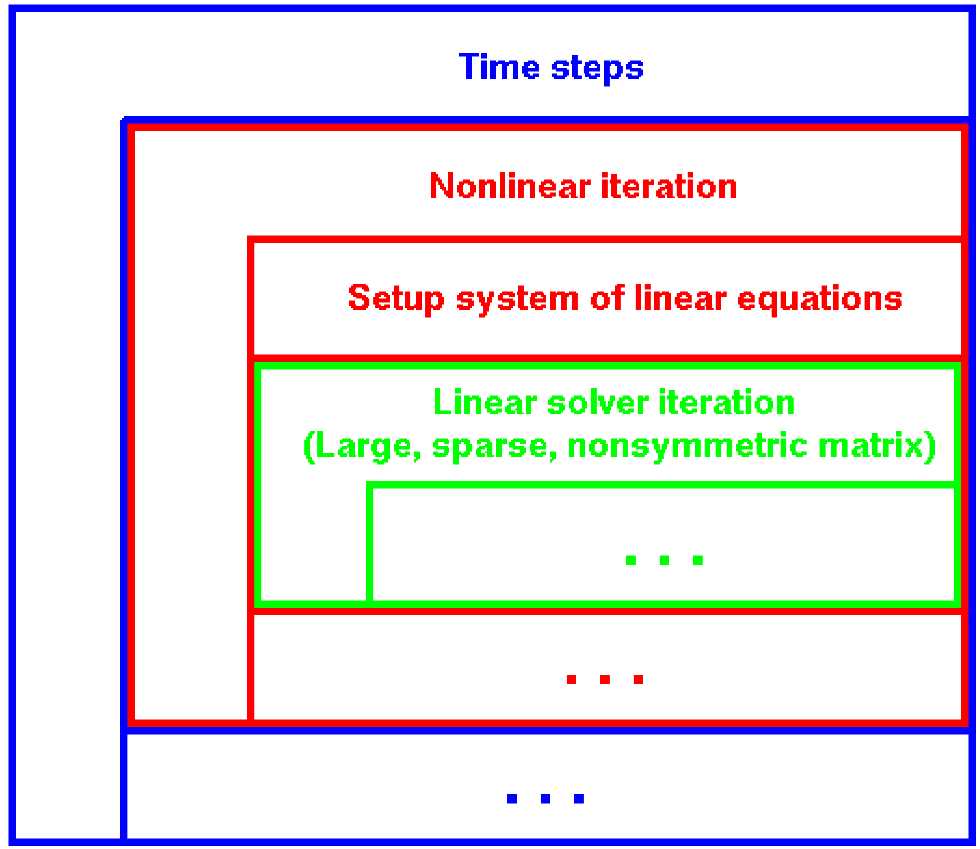
Introduction to circuit simulation (2)

Switches, time varying sources → **Transient analysis**

**In general:
Coupled system of nonlinear ordinary differential equations**

$$f(t, v(t), v'(t)) = 0$$

v(t): vector of nodal voltages at time *t*

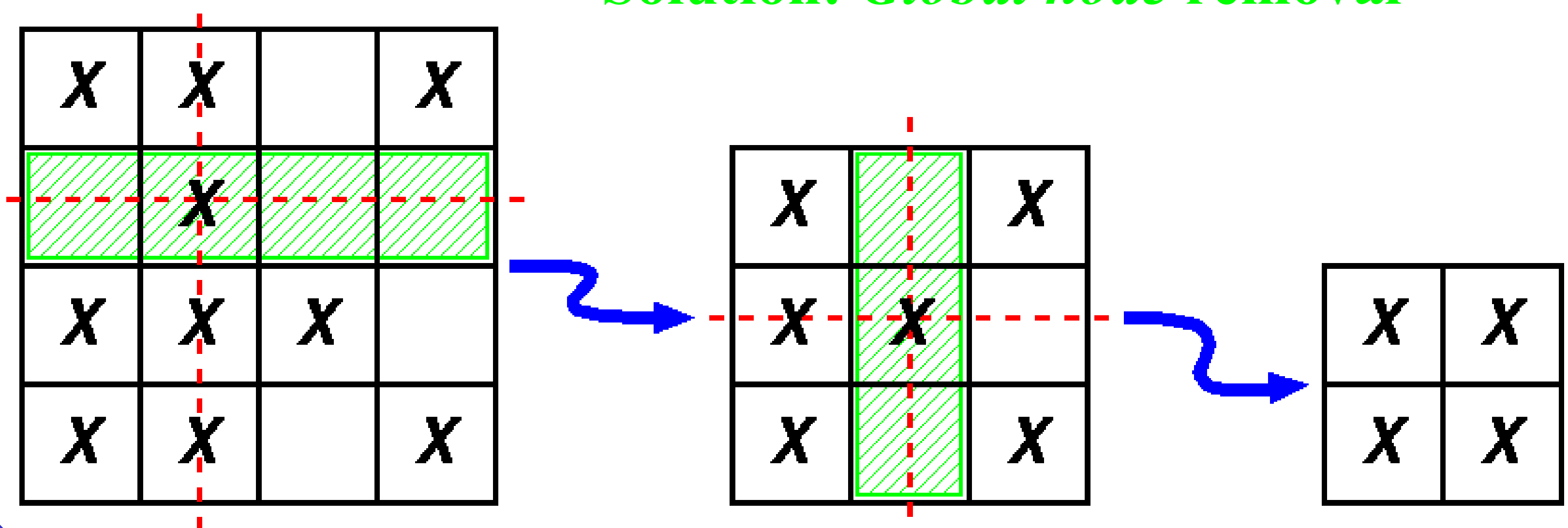


Problem of dense rows and columns (1)

Very sparse linear systems but some dense rows and columns

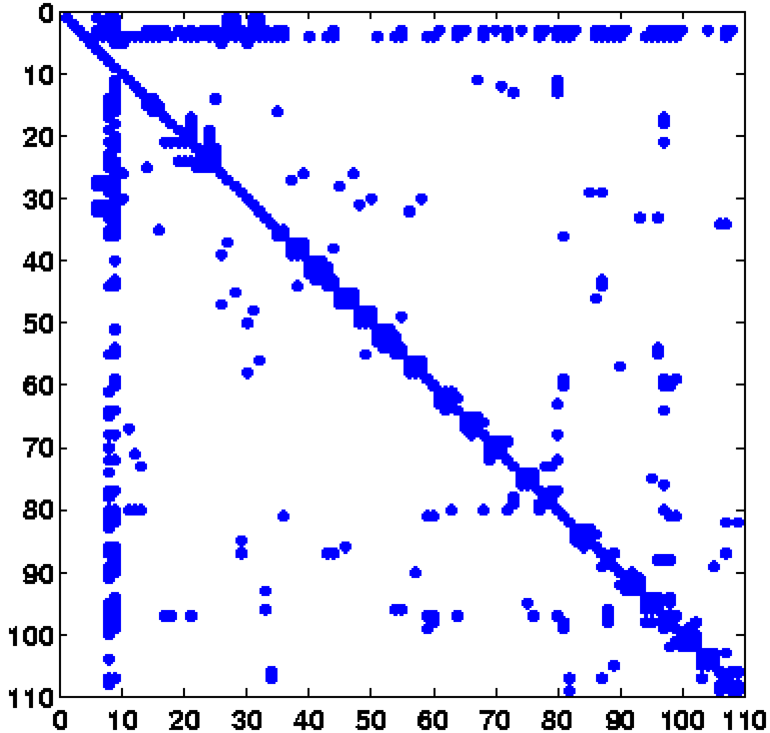
Problematic for: reordering, repartitioning, load balance, condition

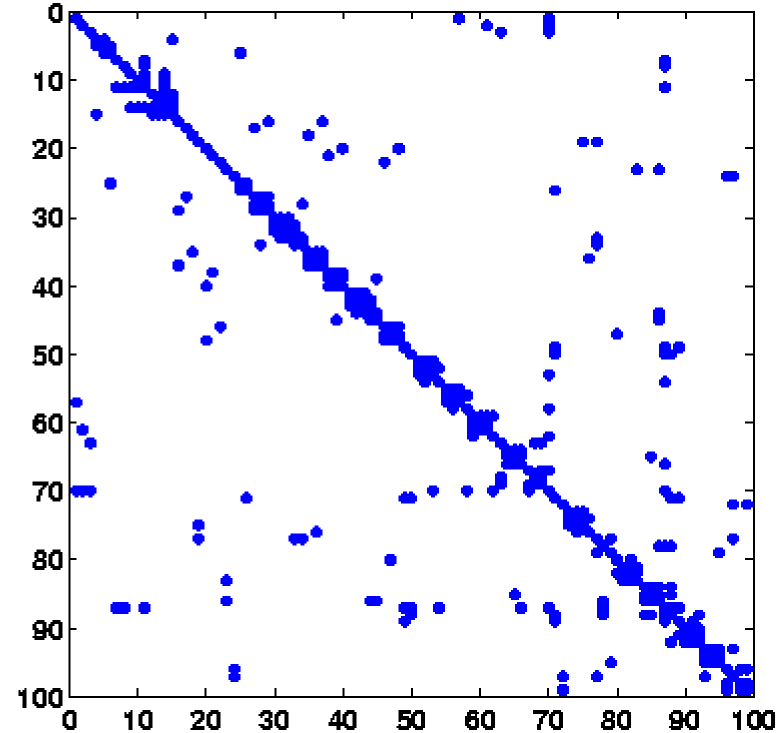
Solution: *Global node* removal



Problem of dense rows and columns (2)

Real, small buffer circuit

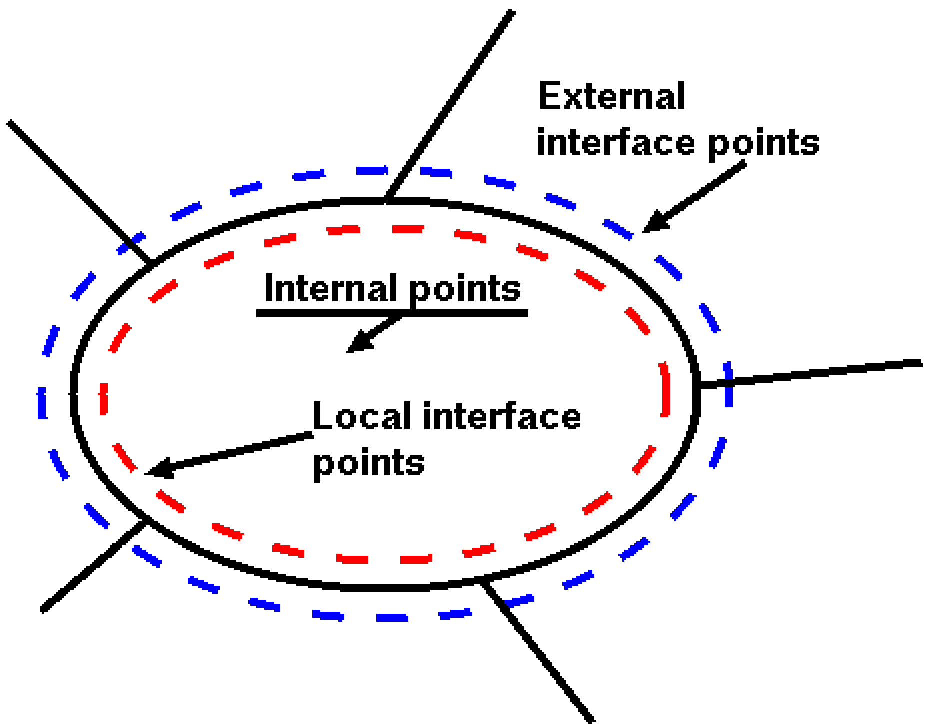




109 rows, 624 nonzeros
Condition: $8.3 \cdot 10^6$

99 rows, 385 nonzeros
Condition: $2.9 \cdot 10^3$

Distributed Schur Complement (DSC) techniques: Definitions (1)

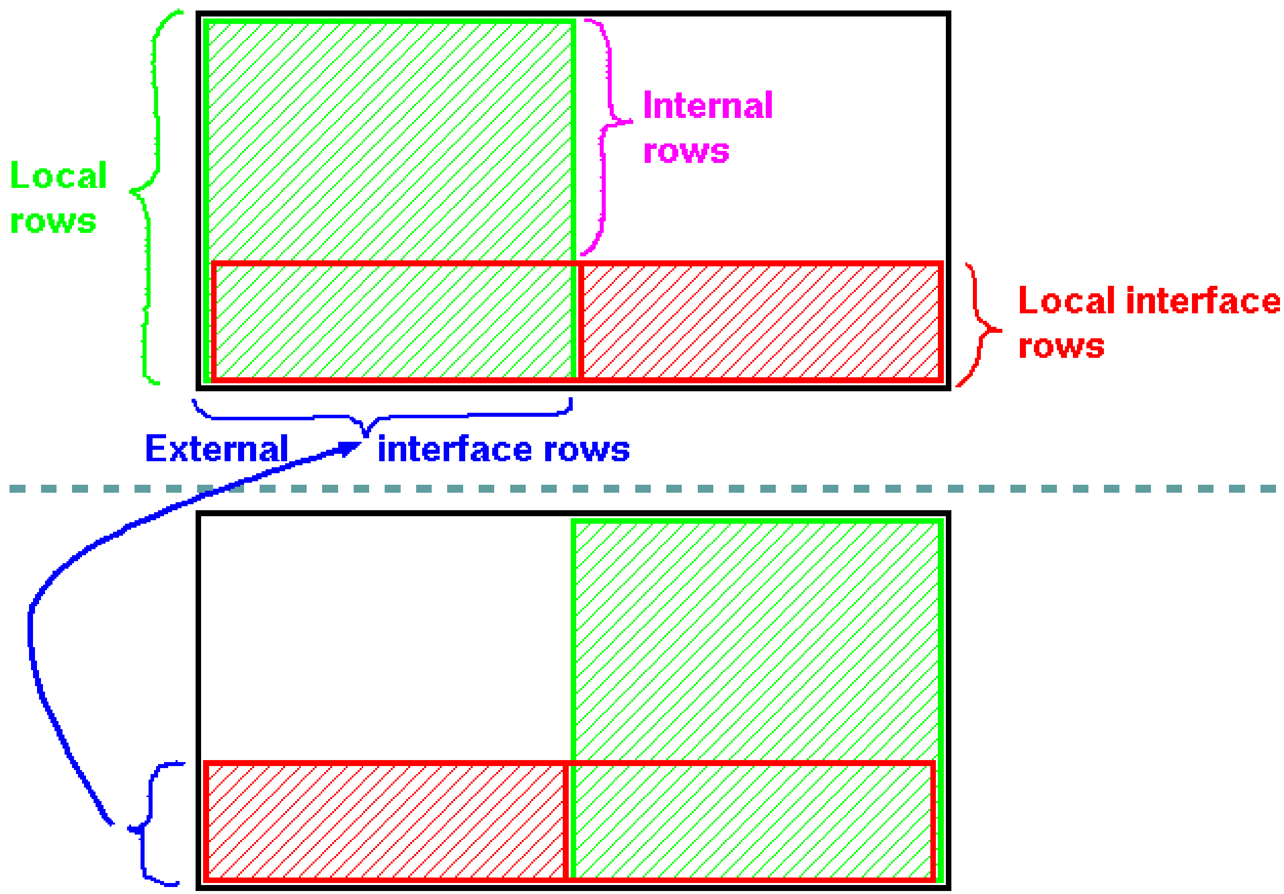


Distributed Schur Complement (DSC) techniques: Definitions (2)

Distributed Matrix, 2 processors

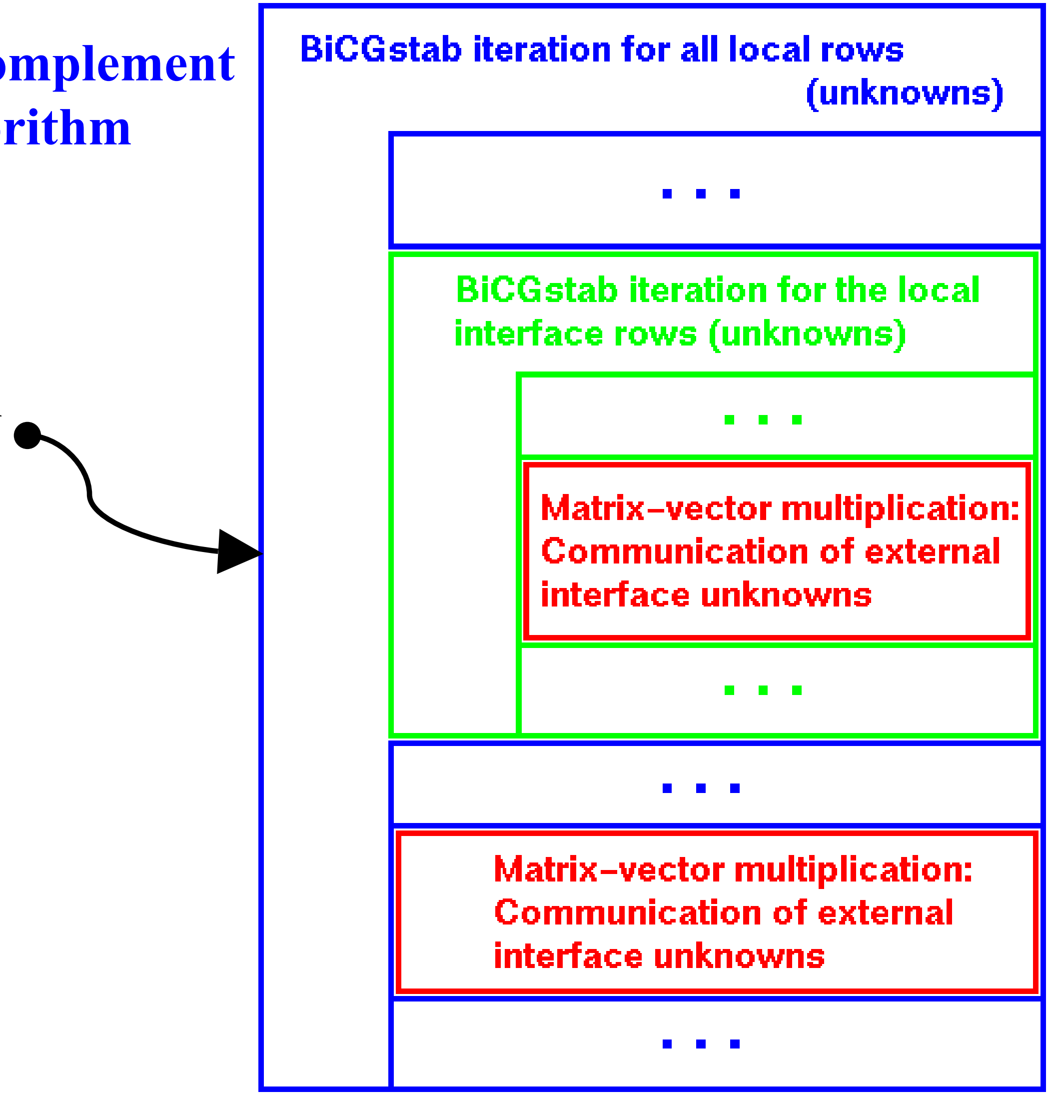
Processor 1

Processor 2



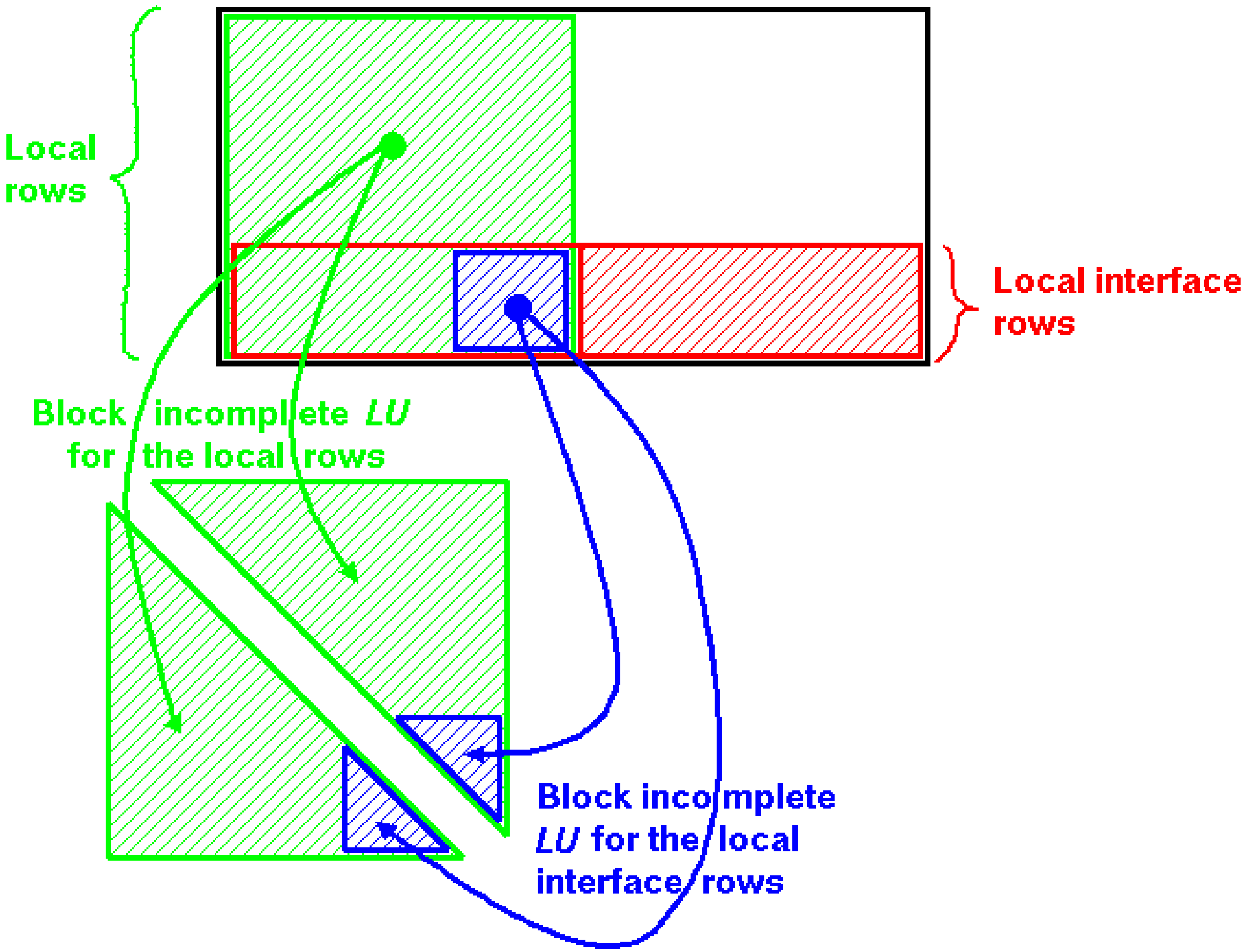
Distributed Schur Complement techniques: Algorithm

Schematic view on each processor



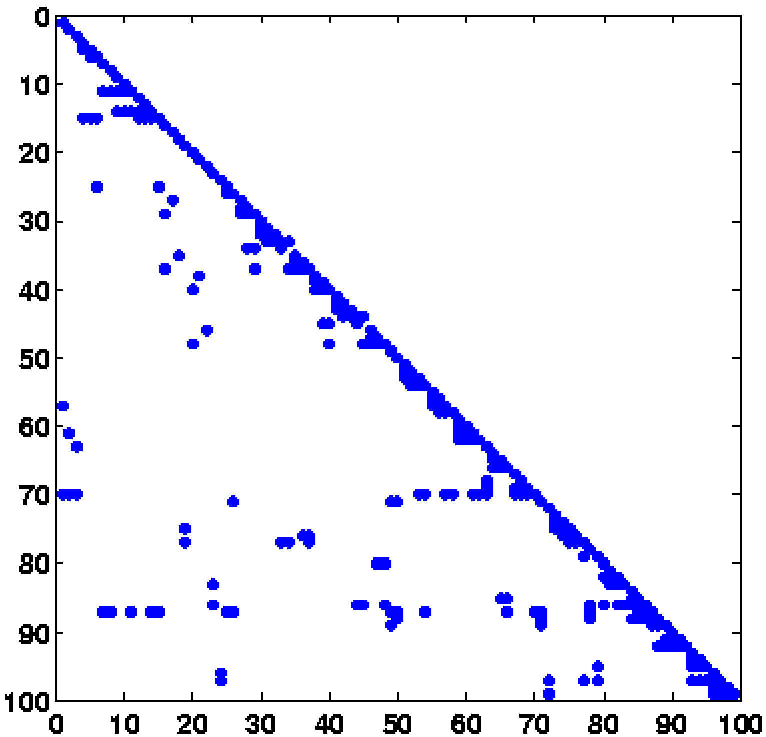
Distributed Schur Complement techniques: Preconditioning (1)

Processor 1

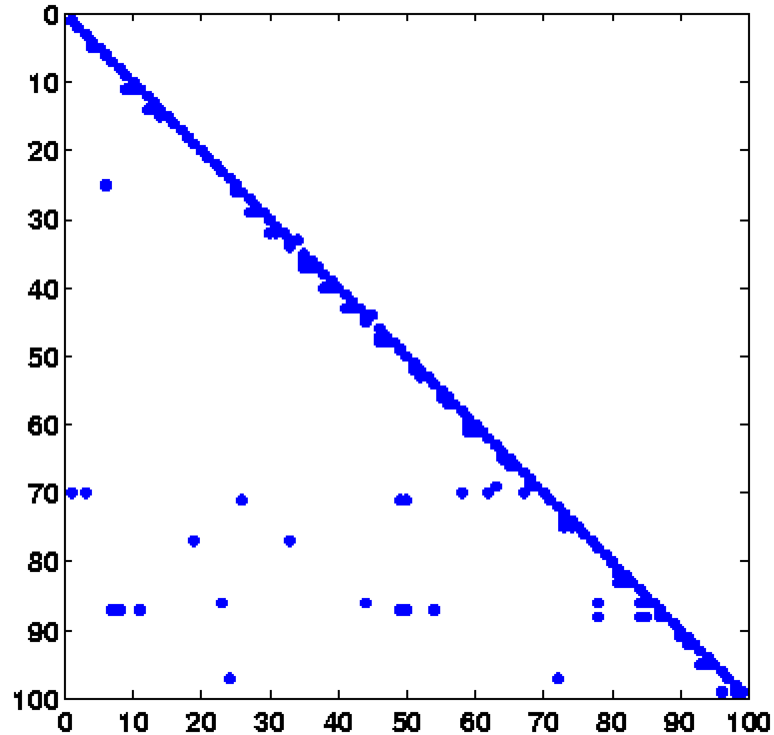


Distributed Schur Complement techniques: Preconditioning (2)

Incomplete *LU* factorization with threshold (ILUT), small buffer circuit



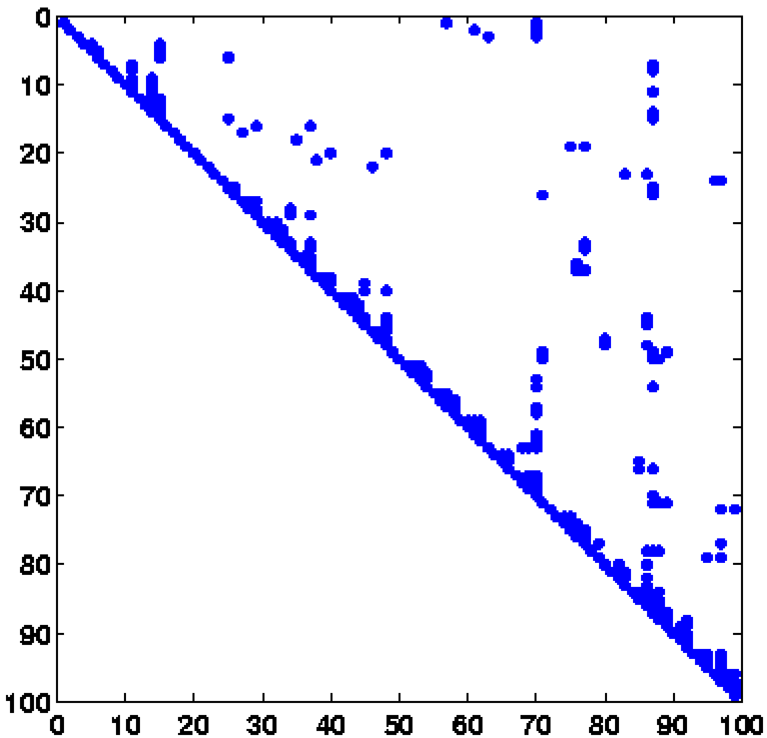
L, complete
288 nonzeros



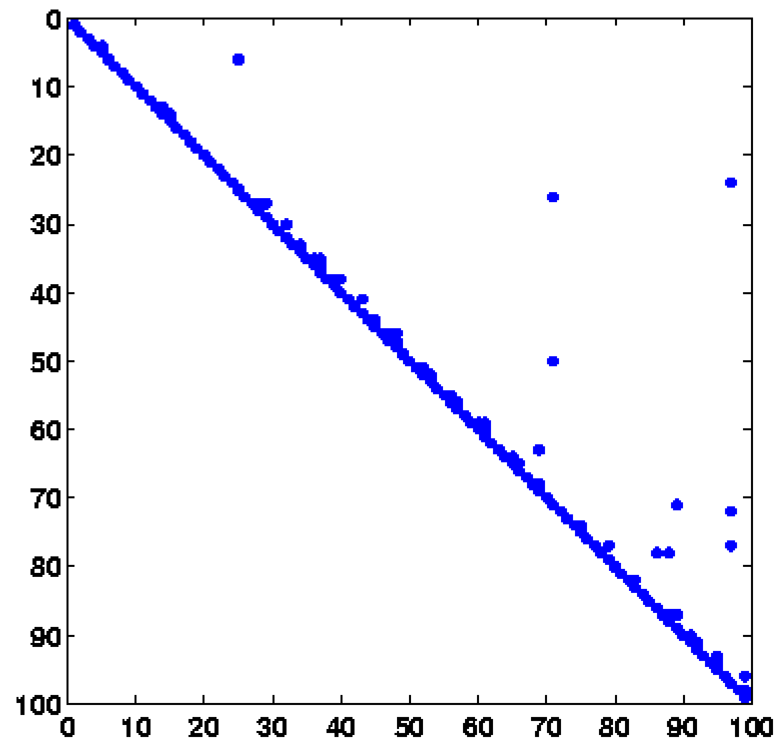
L, incomplete, threshold 10^{-1}
178 nonzeros

Distributed Schur Complement techniques: Preconditioning (3)

Incomplete *LU* factorization with threshold (ILUT), small buffer circuit



U, complete
290 nonzeros



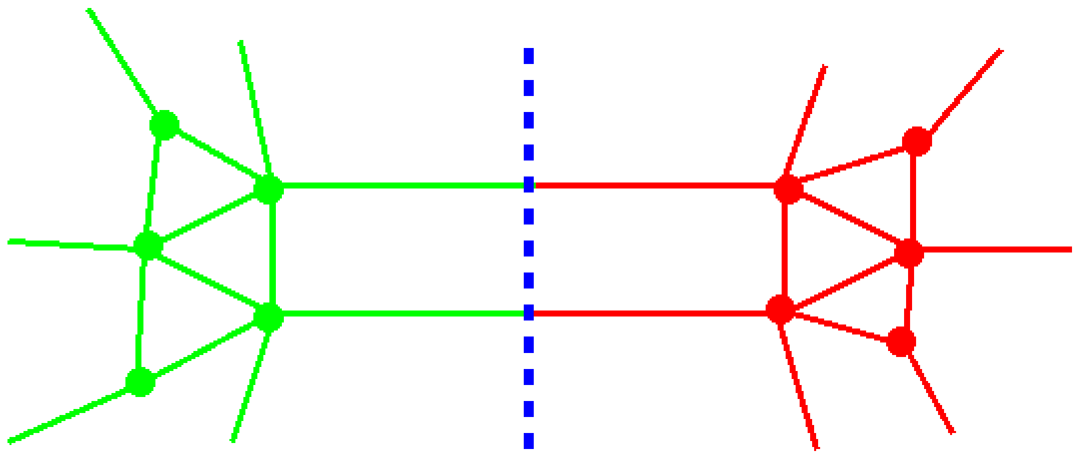
U, incomplete, threshold 10^{-1}
147 nonzeros

Distributed Schur Complement techniques: Repartitioning (1)

Graph partitioning: ParMETIS (University of Minnesota)

Goal:

Minimize the number of edges cut ⟷ number of interface unknowns

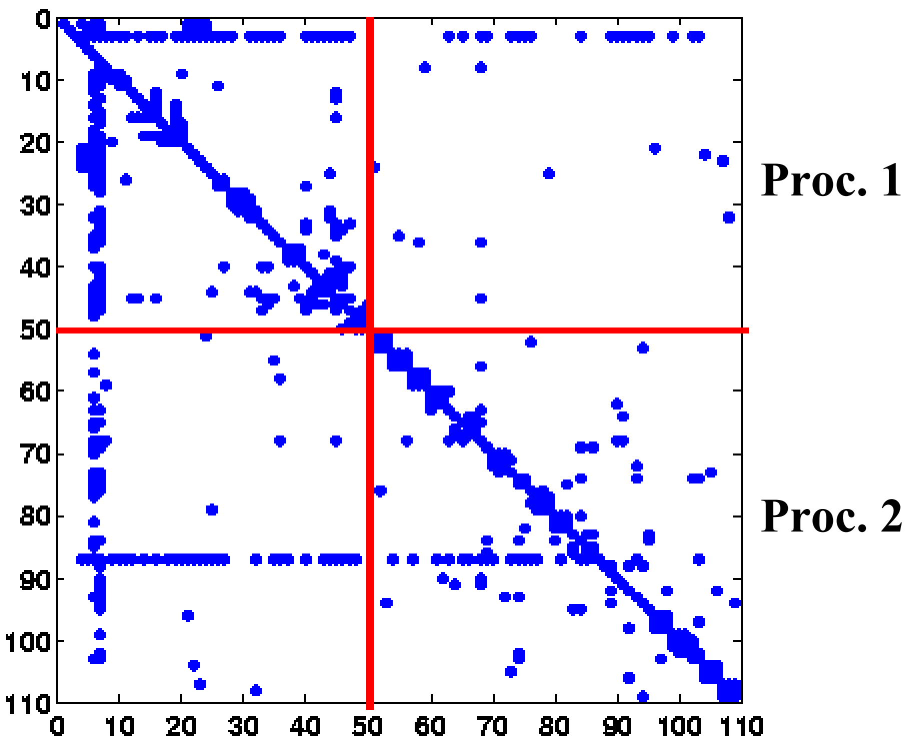


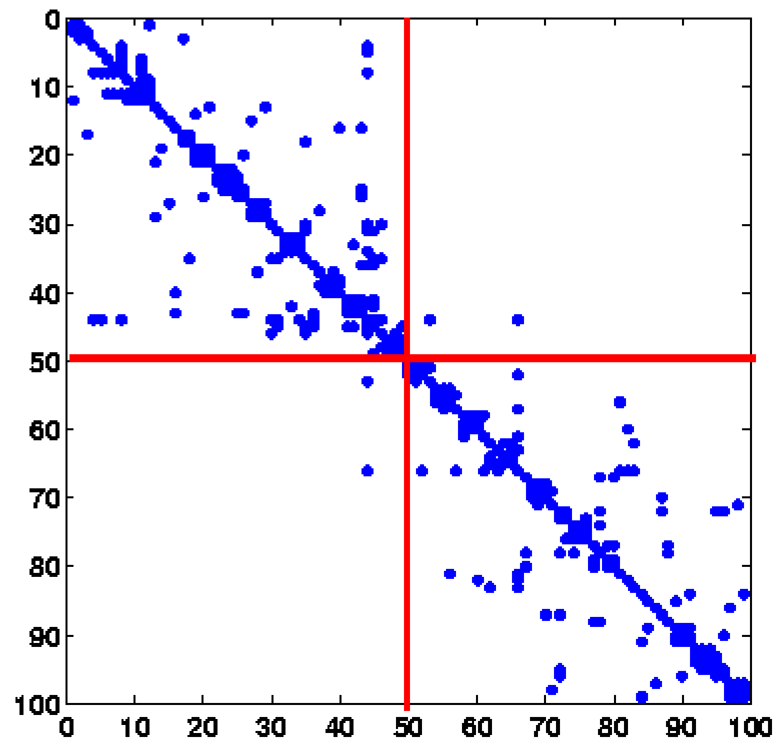
Undirected graph ➡ Symmetrize the matrix pattern

A. Basermann
http://www.ccrl-nece.de
NEC Europe Ltd.
C & C Research Laboratories

Distributed Schur Complement techniques: Repartitioning (2)

ParMETIS partitioning for 2 processors, small buffer circuit



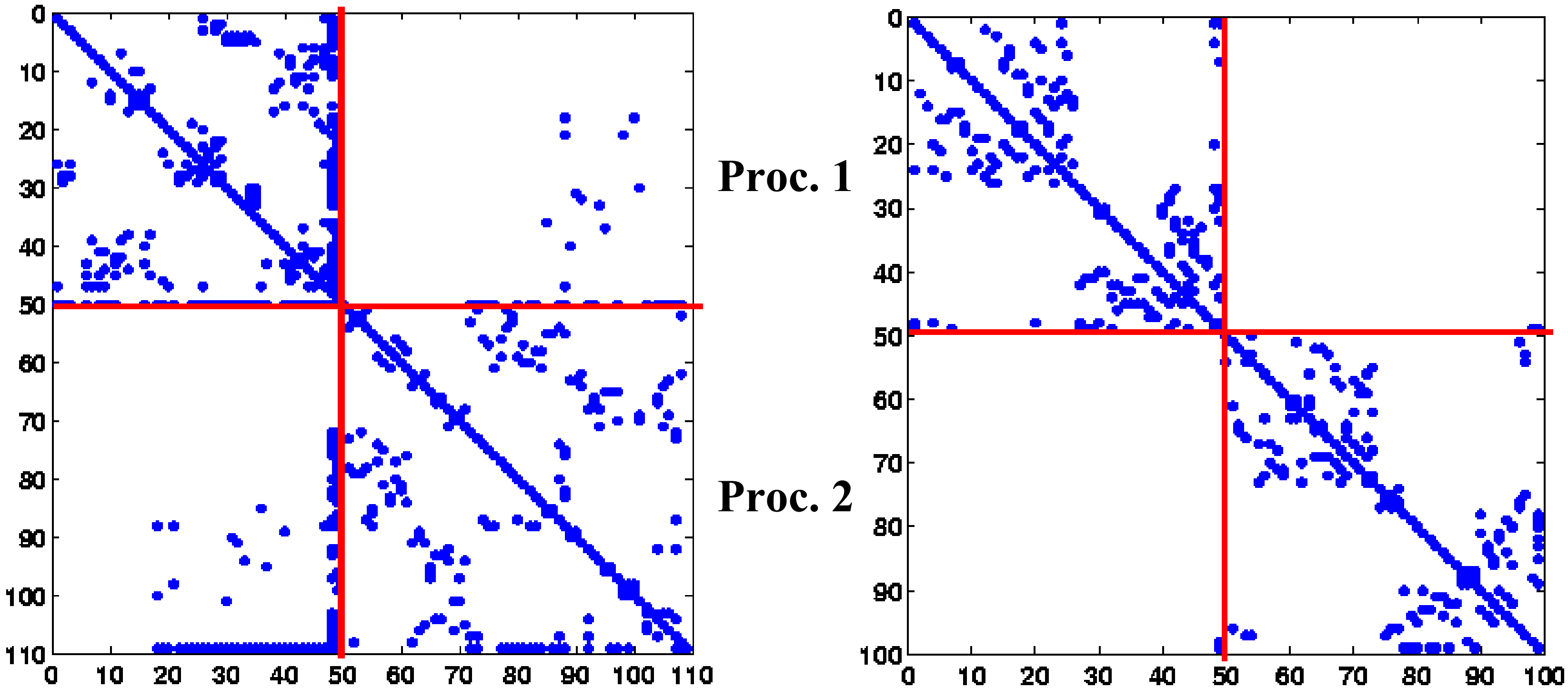


Original matrix

Matrix without global nodes

Distributed Schur Complement techniques: Reordering

METIS nested dissection reordering, 2 processors, small buffer circuit
Goal: fill-in reduction for ILUT (threshold 10^{-4} below)



Original matrix:
Total fill-in 573 ➡ 561

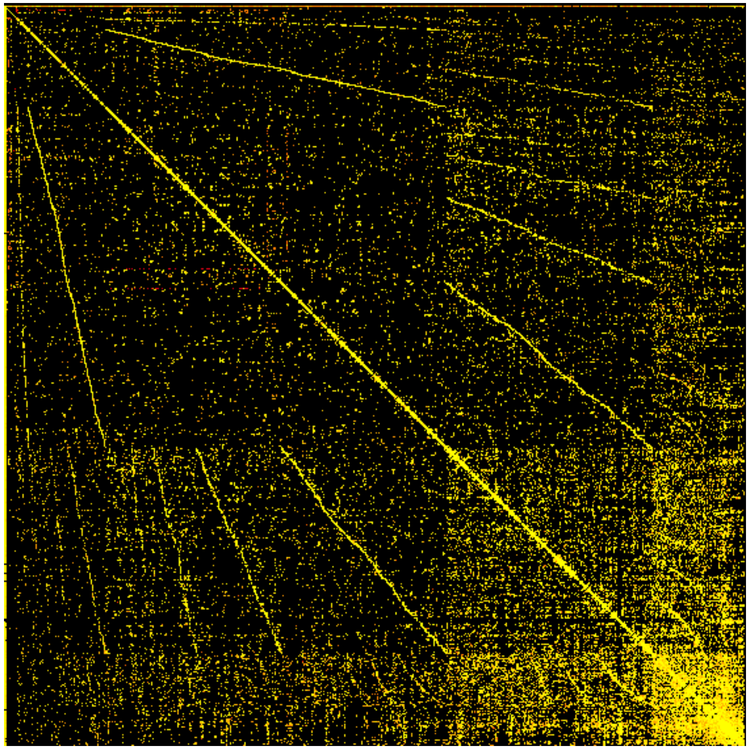
Matrix without global nodes:
Total fill-in 407 ➡ 395

A. Basermann
http://www.ccrl-nece.de
NEC Europe Ltd.
C & C Research Laboratories

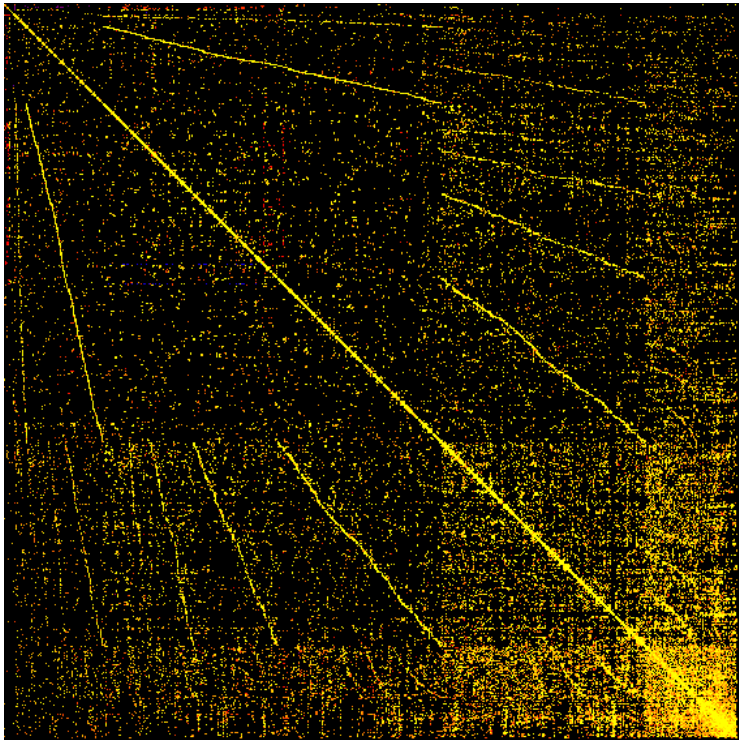
NEC

Results (1)

Matrix from a large memory circuit simulation



Original matrix:
89,556 rows; 760,630 nonzeros

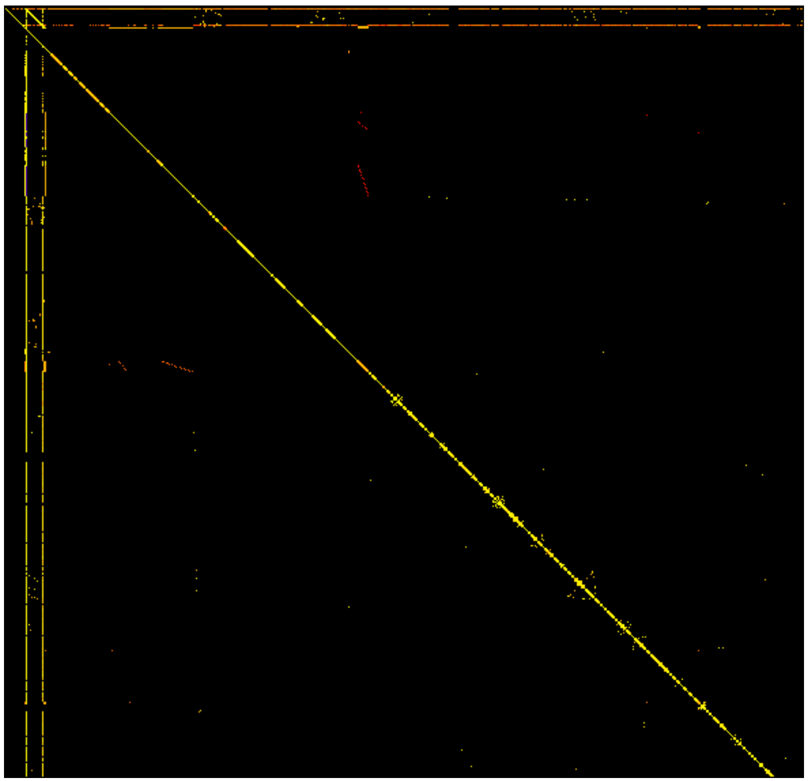


Matrix without global nodes:
89,378 rows; 603,753 nonzeros

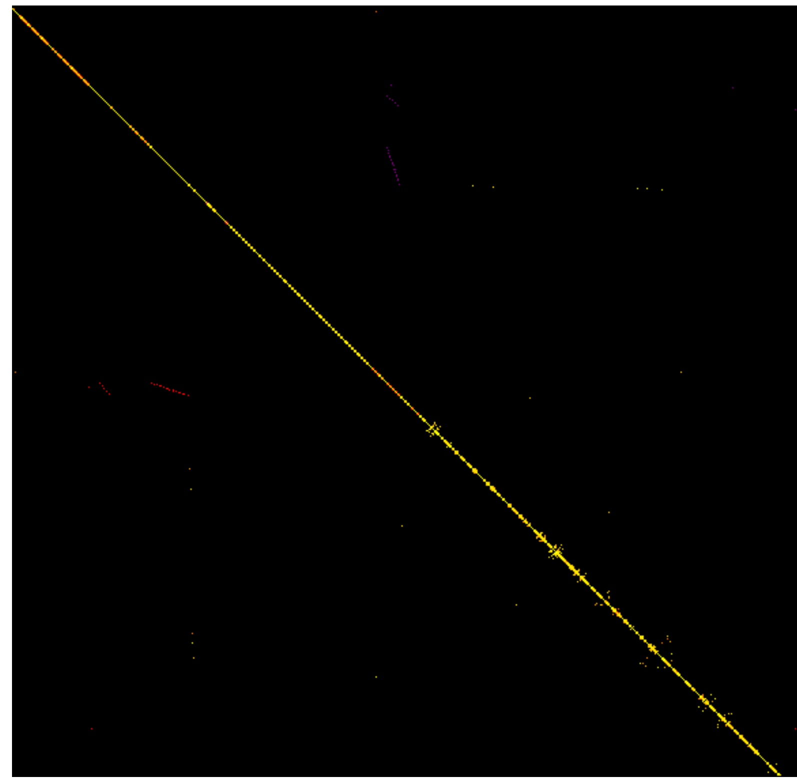
Iterative solver: threshold 10^{-4} ; residual norm / initial residual norm < 10^{-11}

Results (2)

Matrix from a large memory circuit simulation, zoomed in left above



Original matrix

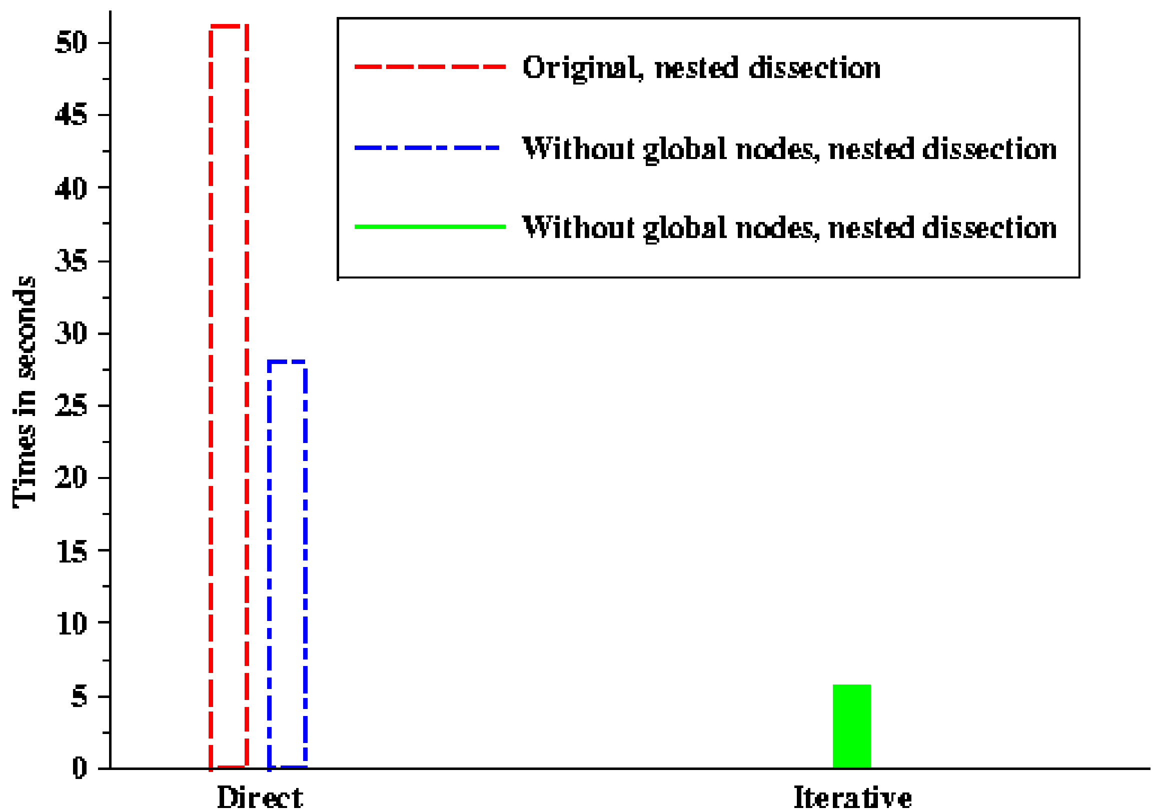


Matrix without global nodes

A. Basermann
http://www.ccrl-nece.de
NEC Europe Ltd.
C & C Research Laboratories

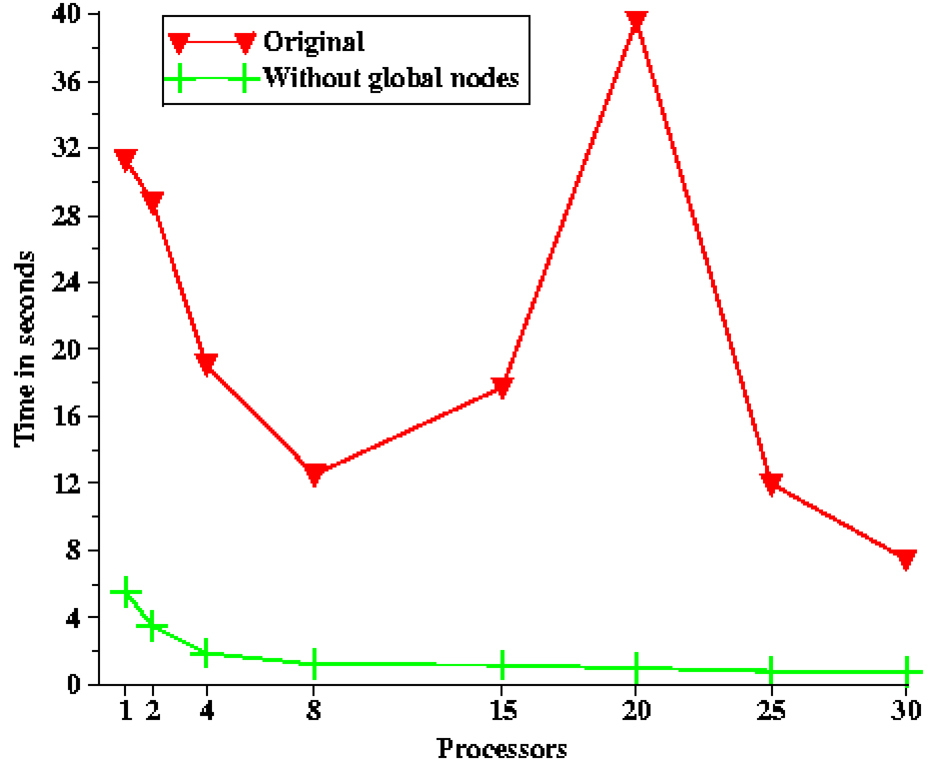
Results (3)

Sequential results, direct versus iterative solver, PC cluster



Results (4)

Parallel results, iterative solver for original and reduced matrix, PC cluster

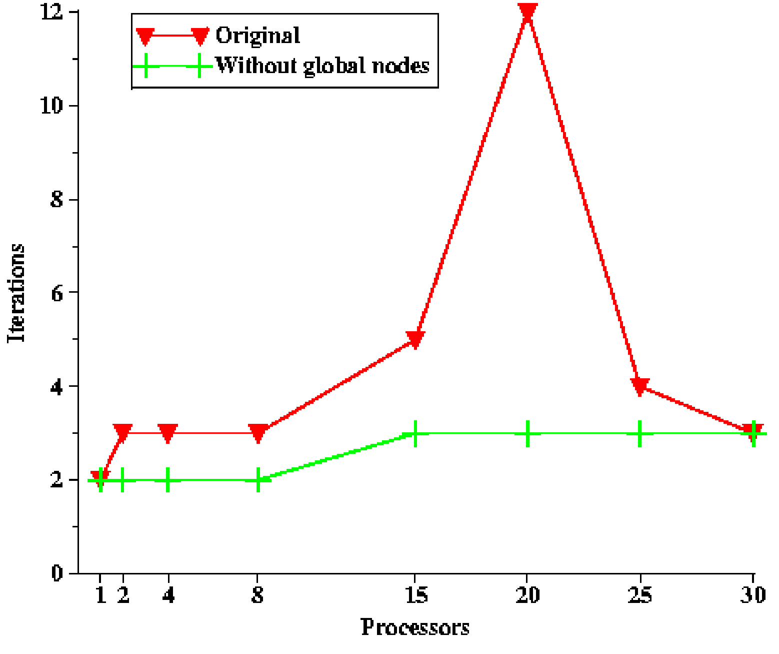


A. Basermann
http://www.ccrl-nece.de
NEC Europe Ltd.
C & C Research Laboratories

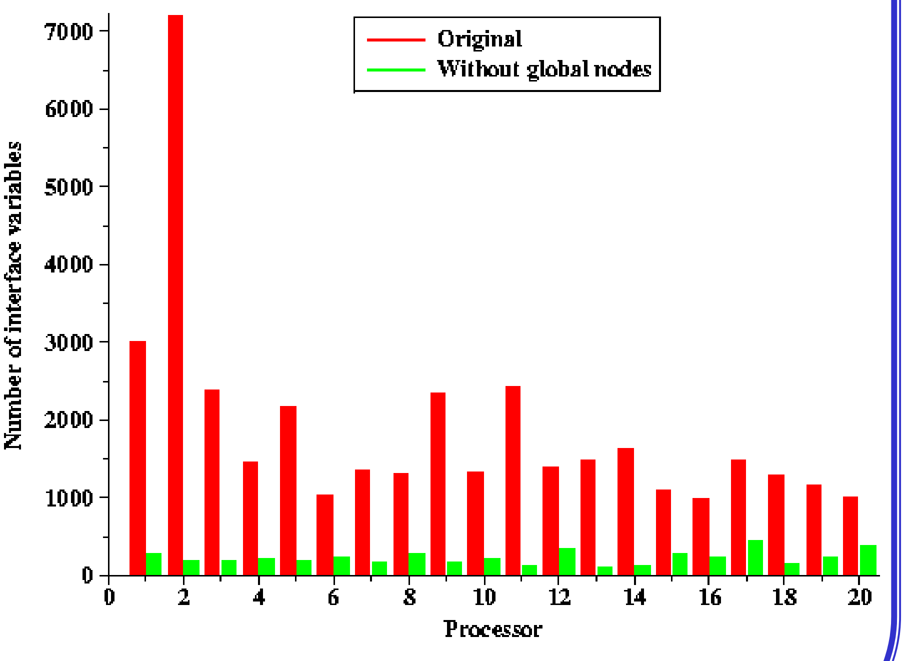
Results (5)

Parallel results, iterative solver for original and reduced matrix, PC cluster

Number of iterations for 1-30 processors

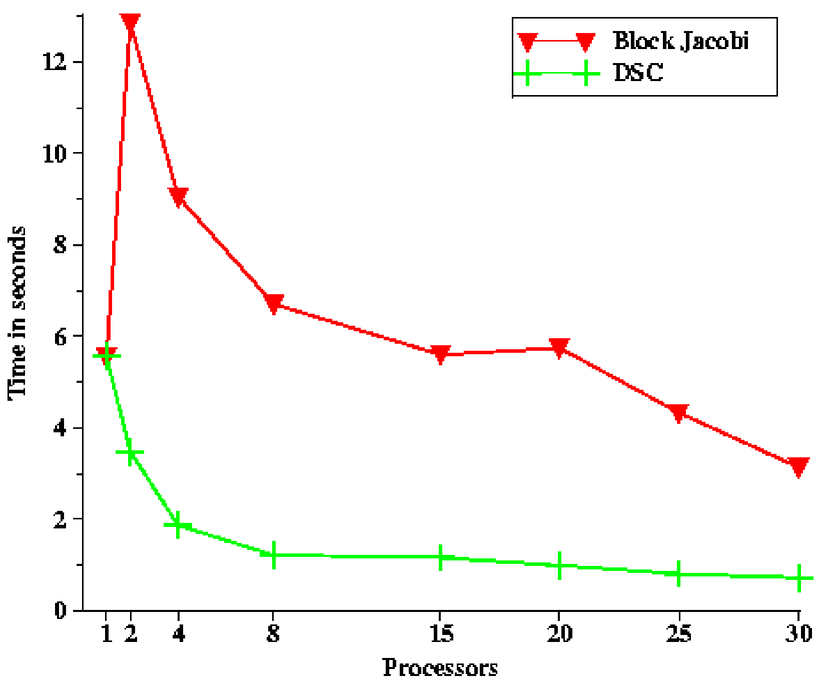


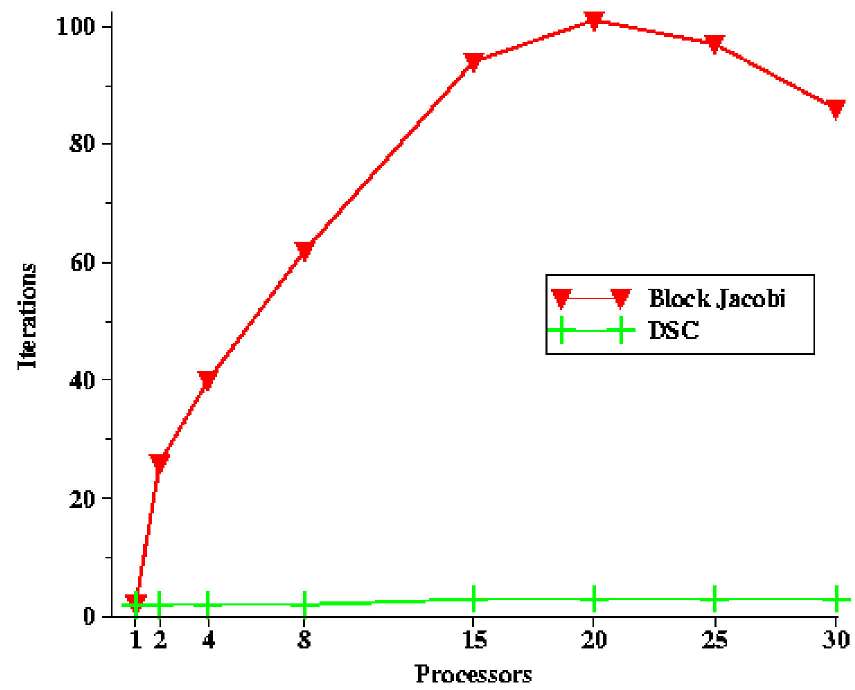
Number of interface variables per processor (20 processors)



Results (6)

Parallel results, block Jacobi versus DSC preconditioning, PC cluster

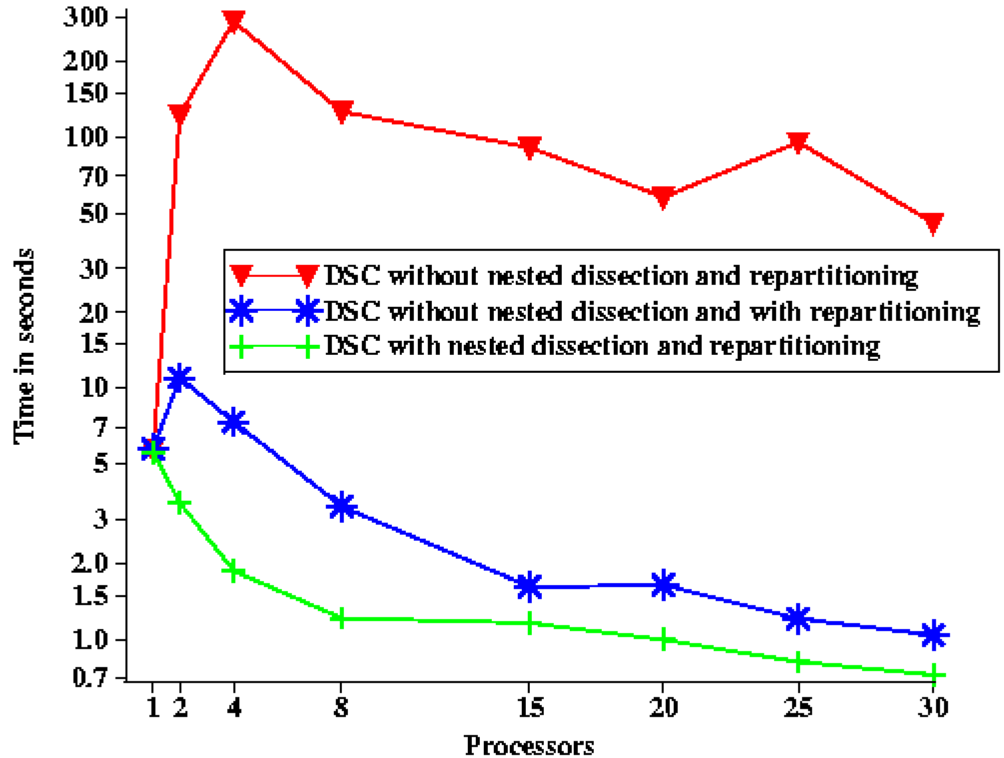




A. Basermann
http://www.ccrl-nece.de
NEC Europe Ltd.
C & C Research Laboratories

Results (7)

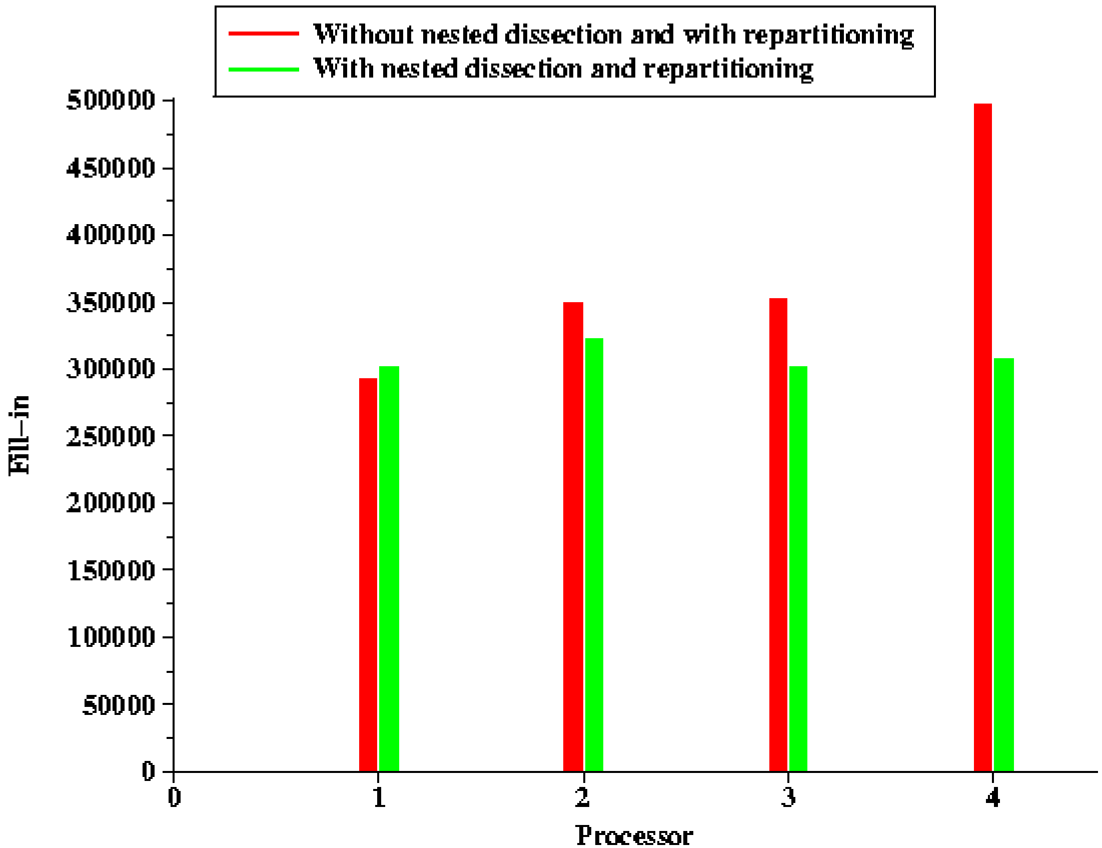
Parallel results, effect of reordering and repartitioning, PC cluster



Results (8)

Parallel results, effect of nested dissection reordering, PC cluster

ILUT fill-in per processor (4 processors)

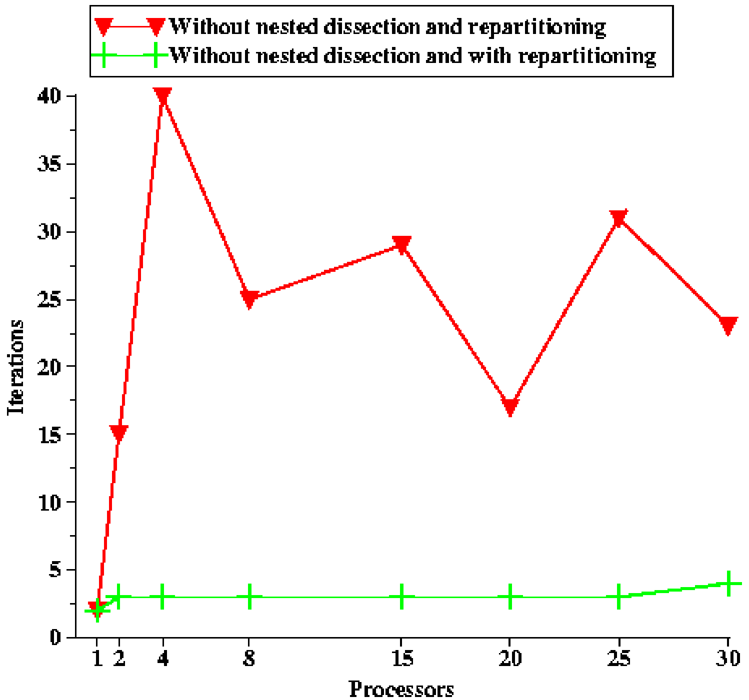


A. Basermann
http://www.ccrl-nece.de
NEC Europe Ltd.
C & C Research Laboratories

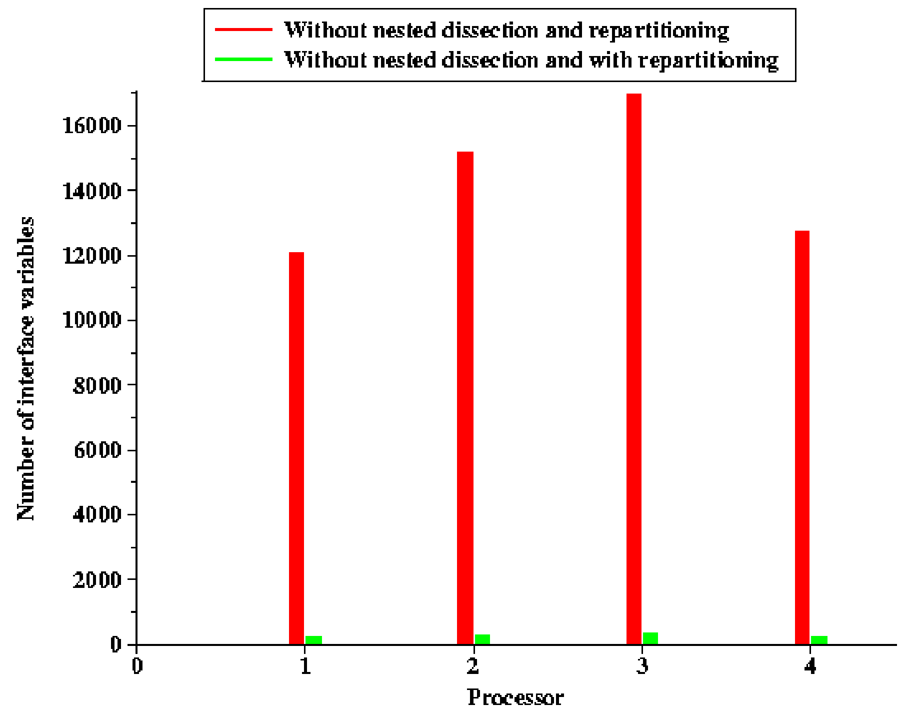
Results (9)

Parallel results, effect of repartitioning, PC cluster

Number of iterations for 1-30 processors



Number of interface variables per processor (4 processors)



Conclusions

- Removal of dense rows and columns necessary
- Iterative solver distinctly superior to direct solution
- DSC preconditioner markedly superior to block Jacobi
- Reordering and repartitioning important for DSC
  - Reordering improves load balance
  - Repartitioning keeps interface system small
- DSC solver well suited for circuit simulation