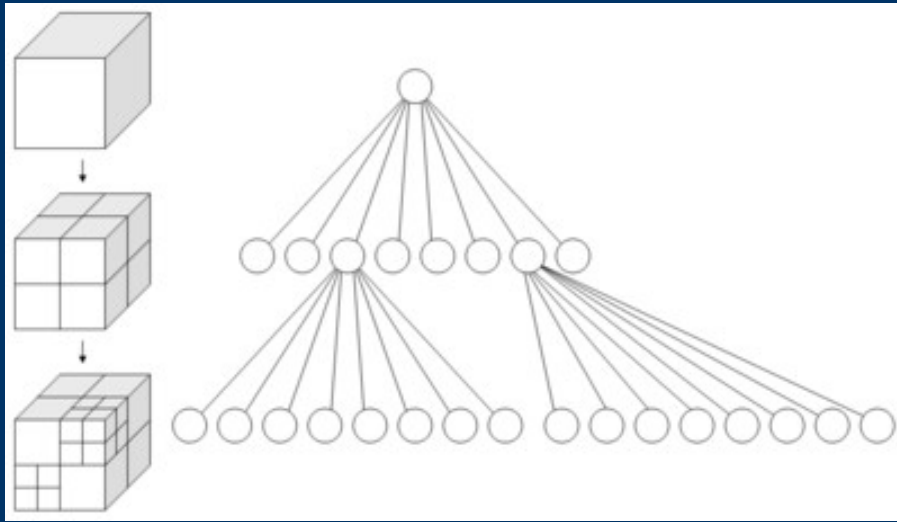


Preconditioning in Parallel

Vikram Aggarwal



Remedial 595



- Solving a Finite Difference equation over a volume.
- 3D volume is discretized using Octrees
- Equation is Poisson's Equation with variable coefficients.

Remedial 595 (lecture notes)

- Matrix properties:
 - At max, 11 non-zeros per row. (Sparse)
 - Non-symmetric structure.
 - Non-diagonals are negative. Diagonals are positive.
 - Large number of rows. Growth is quick.
 - Solution properties:
 - Iterative methods alone don't converge (GMRES/BiCGSTAB)
 - ILU (0) works remarkably well.
-
-

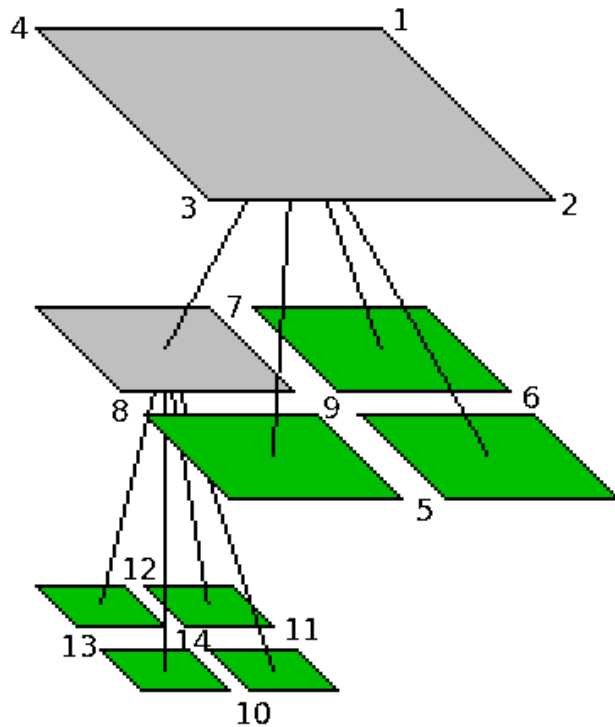
Problem and solution

- Want to solve the largest possible 3D systems.
 - Necessitates parallel computation (space and time limitations both)
 - ILU (0) with BiCGSTAB found to be the most promising.
 - But triangular solves do not parallelize too well.
 - Let's try parallelizing them in **this** case.
-
-

Existing Parallel Triangular solvers

- Dense case: lots of valuable research
 - Fan-out vector sum
 - Fan-in scalar product
 - Wavefront...
 - Sparse case:
 - Graph colorings (to find independent sub-problems)
 - Partitioned Inverses ($L = L_1 * L_2 * L_3 \dots$)
 - Hacks on the substitution algorithm: scheduling,
 - Special cases: Cholesky
-
-

hierarchy based ordering



- Nodes have hierarchy
- Corners hierarchy according to creation time.
- Allows for one traversal of corners.

Leads to a numbering

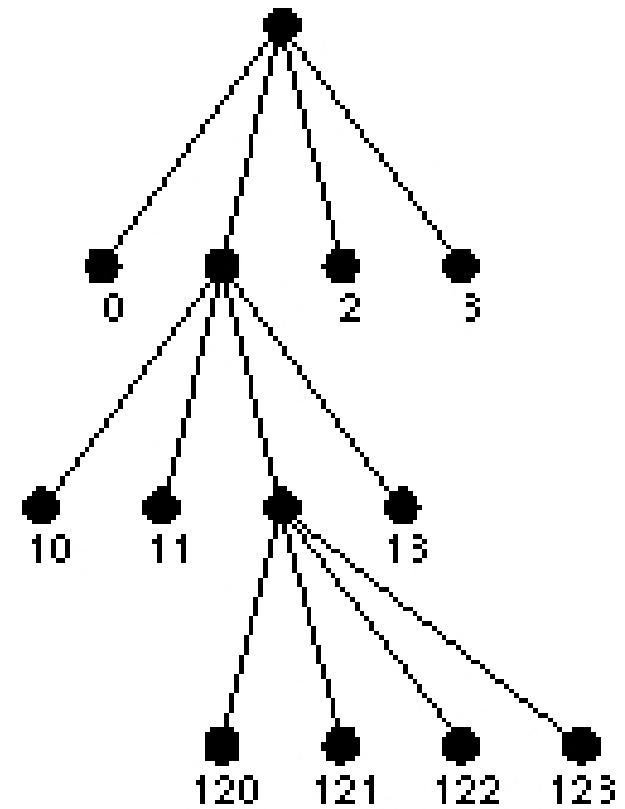
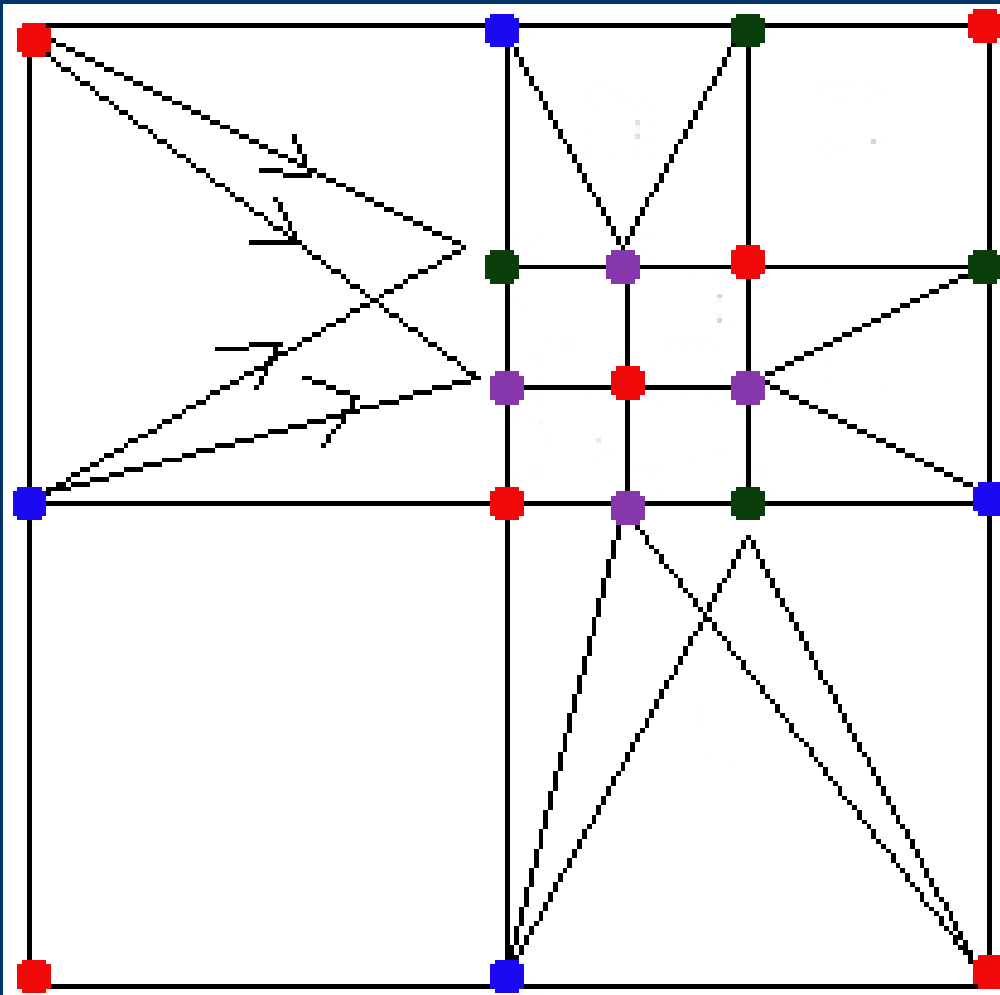
- We can use this ordering as a numbering for the corners.
 - Resulting matrix has fun properties:
 - Leads to small number of partitioned inverses
 - ILU (0) factors are not much worse than naive numbering.
 - And finally...
 - Un-symmetric edges are all on one triangle!
-
-

Radio sounds better in Color!

- The ordering when used to color nodes gives surprising results
- The maximum in-degree is 10.
- 10 colors should be sufficient to color the entire 3D graph.



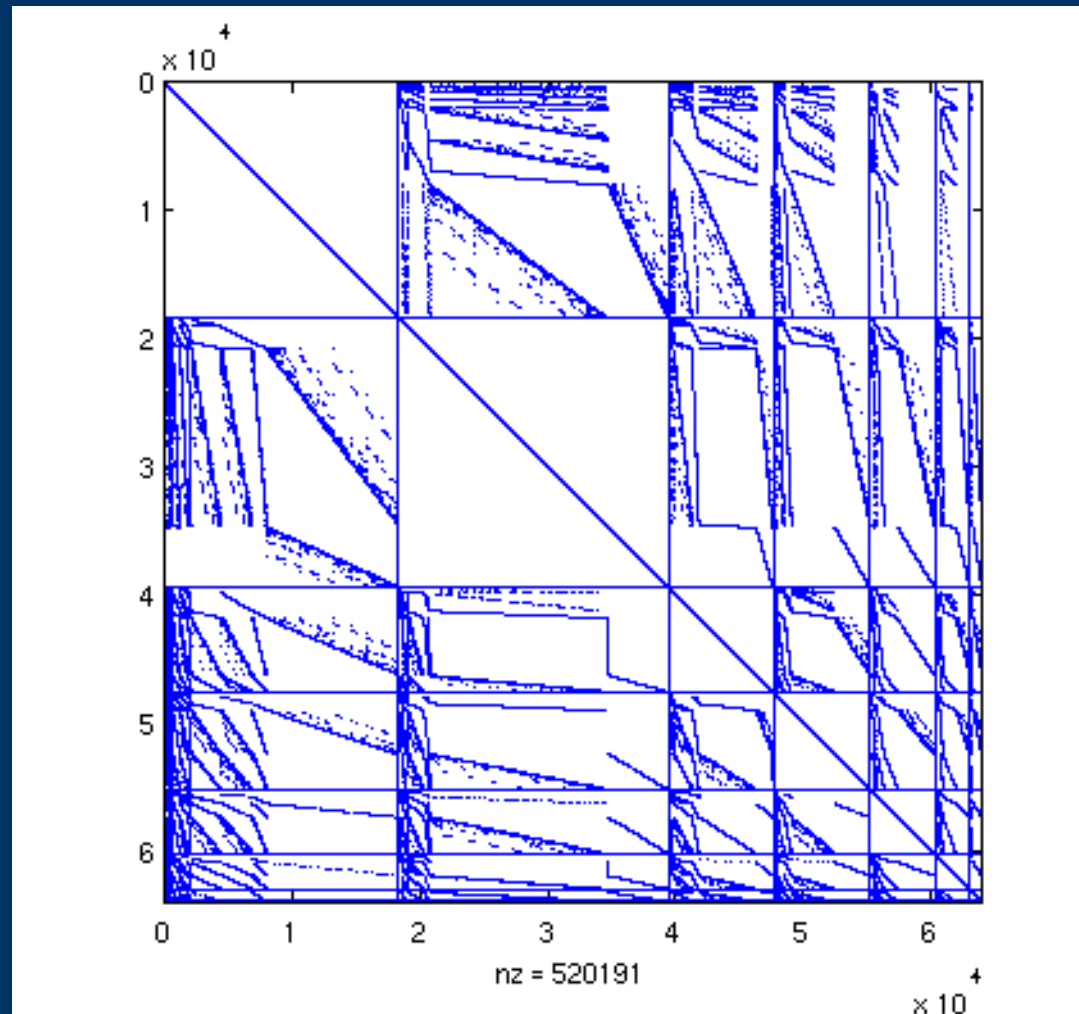
Example



What next...?

- Colors lead to a numbering. Reds are put together, then blues...
 - Leads to matrix with large diagonal blocks
 - Colors can be solved independently, in parallel.
 - After one color is solved (in parallel), nodes must communicate the value obtained.
 - Using Fan-out vector-sum algorithm.
 - Quality of ILU (0) factor is not diminished.
-
-

Matrix structure



Steps (please comment)

- Using HyPRE for parallel iterative methods.
 - I'm done writing HyPRE hooks for putting in a parallel triangular solver.
 - Matrix generation code (GeomOct) already modified to generate these matrices.
 - Colored nodes need to be spread evenly among CPUs. This is nearly written.
 - Parallel triangular solve (in MPI), outside of HyPRE written. Debugging now.
-
-

Steps...

- Generate Matrix in GeomOct. Permuted according to the color-hierarchy numbering.
 - Construct ILU (0) factors, and split factors for parallel computation.
 - HyPRE reads the matrix, and ILU (0) factors, and performs calculations in parallel.
 - Verify that iterations are small.
 - Verify that time spent is much less.
-
-