Using nested dissection to solve problems in electrostatics

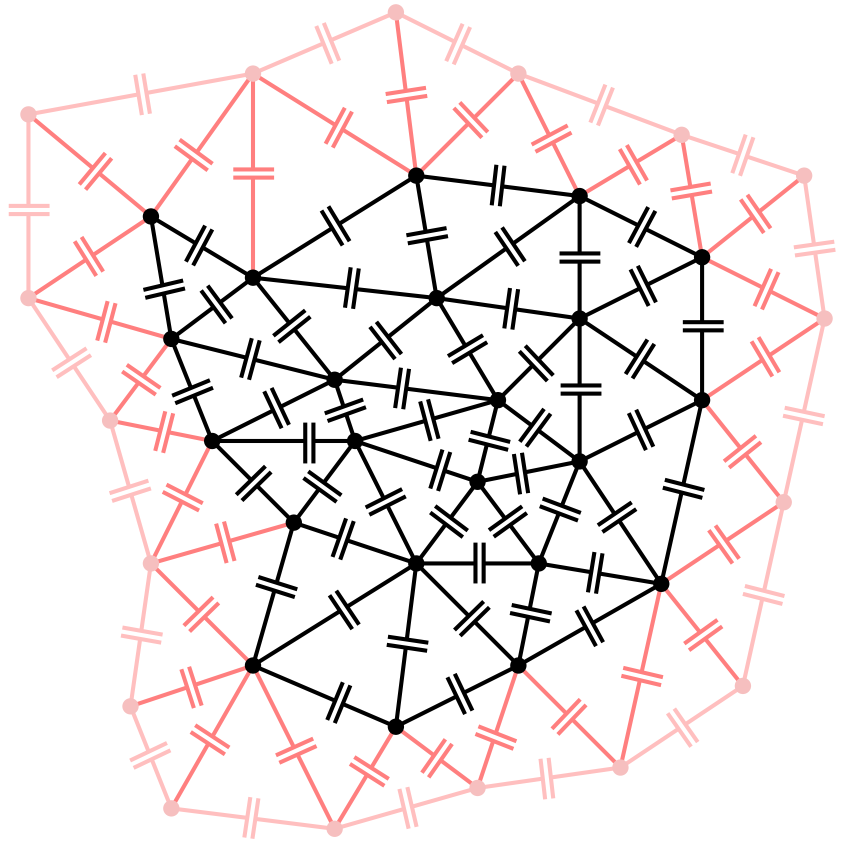
Shawn Eastwood

Introduction

- The “nested dissection” of linear systems remains a powerful tool for solving linear systems that cover large computational meshes.
- The example application that will be addressed here will be the problem of computing the voltages at all points within a volume given the voltage at each point on the surface.
- The focus will be on solving the linear system as opposed to the setup of the linear system.
- The computational complexity of the nested dissection approach for a 3D computational mesh will be analyzed and compared with other approaches.
- Other benefits of nested dissection will also be addressed.

Electrostatics and the finite element method

- Over a volume Ω, solve Laplace's equation:
  $$\nabla^2 V = 0$$
  where V is known at each point on the surface of Ω.
- From finite element modeling, the linear system to be solved is:
  $$C_{int}\vec{V}_{int} + C_{bound}\vec{V}_{bound} = \vec{0}$$
  where $\vec{V}_{bound}$ is known and $\vec{V}_{int}$ is sought.
- The entries of C_{int} and C_{bound} are measures of capacitance.

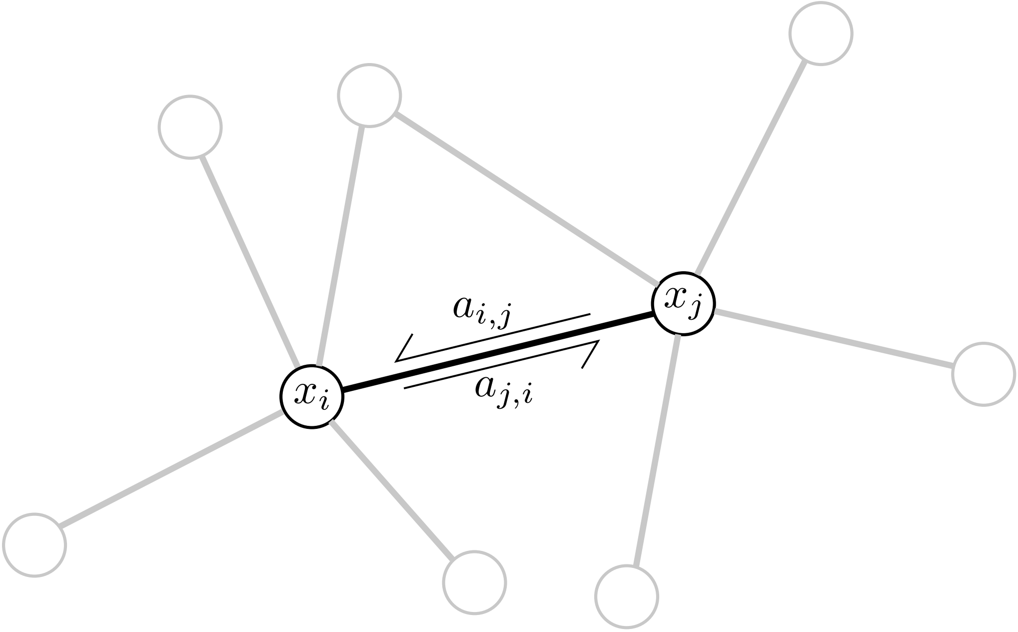


The straightforward approach

- Mesh dimensions are $O(N)$.
- $\vec{V}_{int}$ has $O(N^3)$ entries.
- $\vec{V}_{bound}$ has $O(N^2)$ entries.
- C_{int} has dimensions of $O(N^3) \times O(N^3)$. This is also a **sparse** matrix.
- C_{bound} has dimensions of $O(N^3) \times O(N^2)$. This is also a **sparse** matrix.
- The goal is to compute the $O(N^3) \times O(N^2)$ solution matrix A such that:
$$\vec{V}_{int} = A\vec{V}_{bound}$$
- Without any exploitation of the sparse structure, this requires $O(N^9)$ flops.

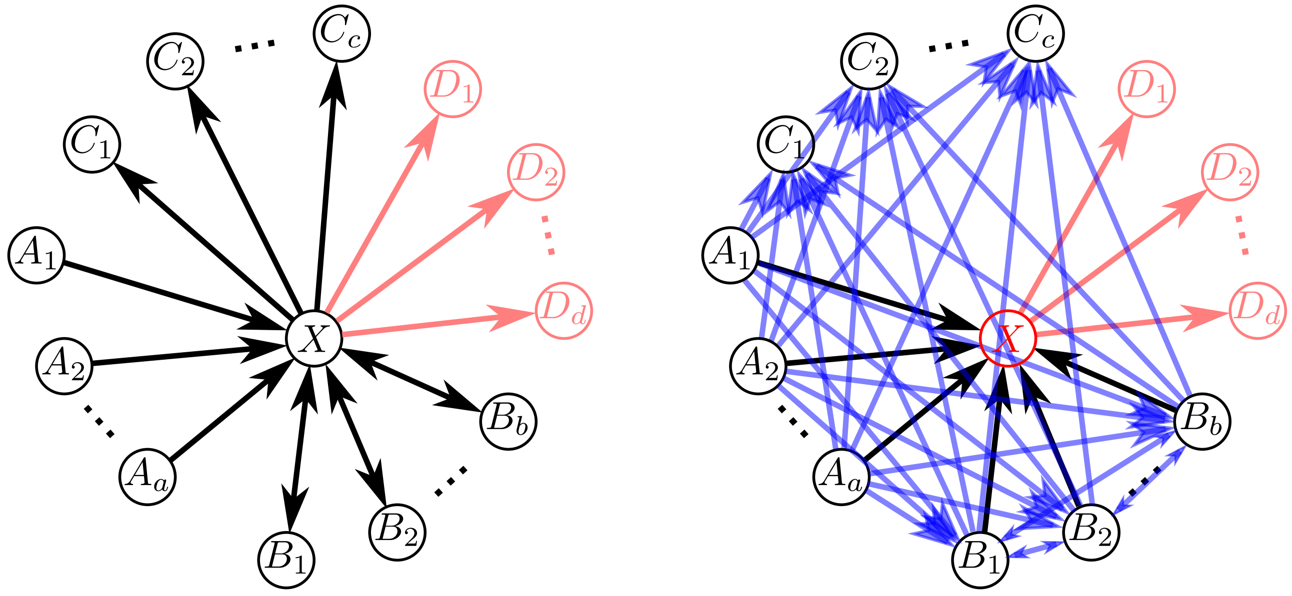
Sparse matrices and computational meshes

- Consider solving the linear system
$$M\vec{x} = N\vec{y}$$
- There are 2 types of variable: "unknown" and "conditioned" variables.
  - Unknown variables from $\vec{x}$ are each associated with a linear equation.
  - Conditioned variables from $\vec{y}$ have no equation and can be set to any arbitrary value.
- A directed edge runs from x_j to x_i if x_j appears in the equation associated with x_i. This also means that the (i, j) entry from the matrix $[M, N]$ is non-zero.

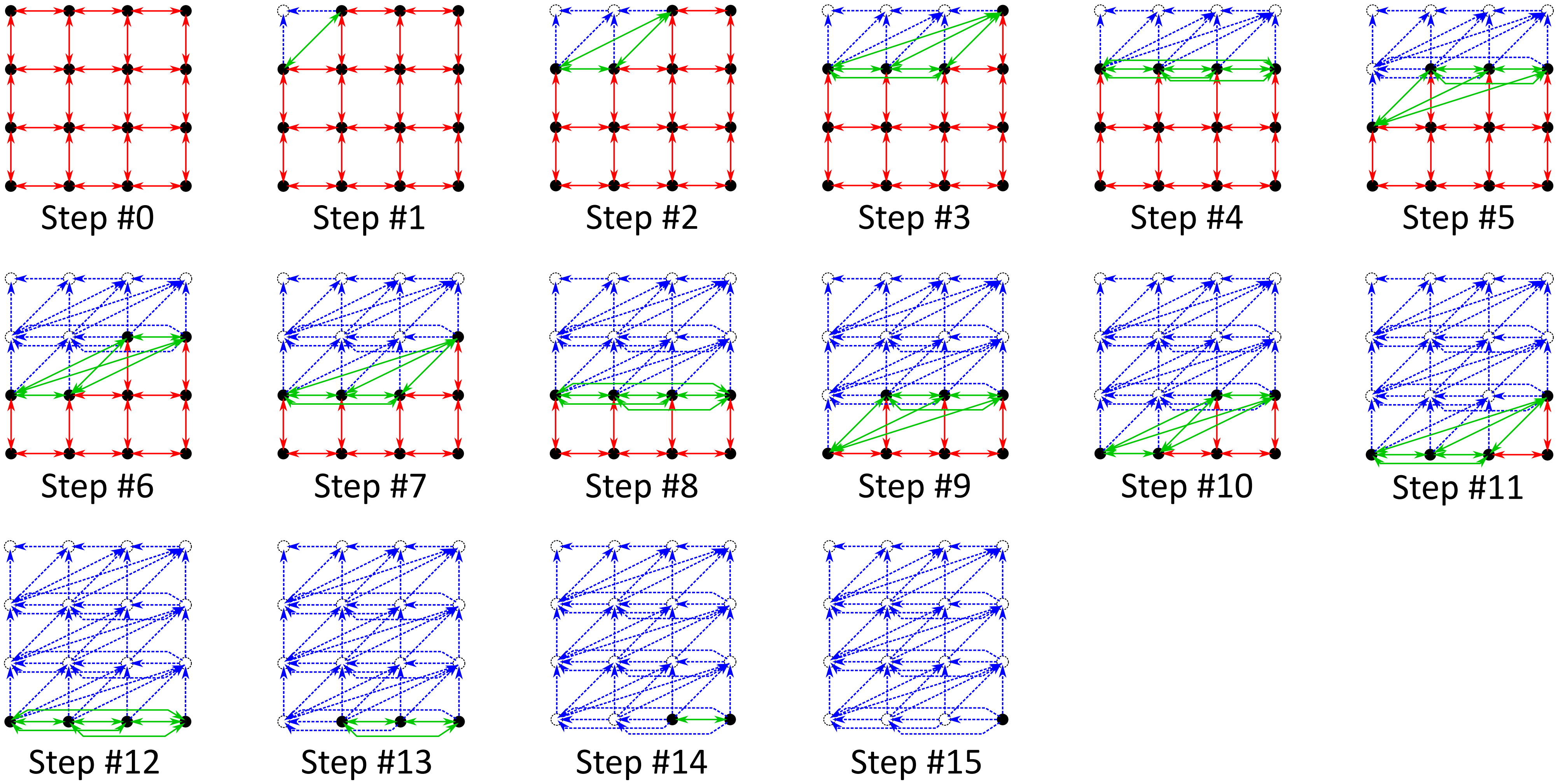


Variable elimination

- Initially in a linear system, unknown variables depend on other unknown variables.
- "Eliminating" an unknown variable X is to make all other non-eliminated unknown variables not dependent on X.
- This is done by substituting the expression for X in place of all appearances of X in the equations associated with the non-eliminated unknown variables.
- From the image on the right, the computational complexity is $O((a+b)(b+c))$ flops.

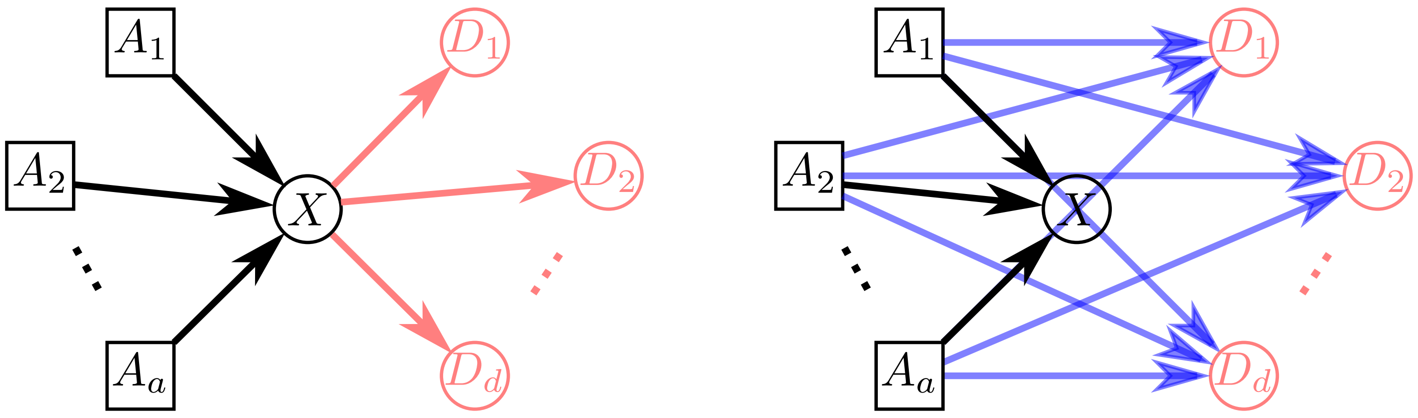


Example Variable Elimination

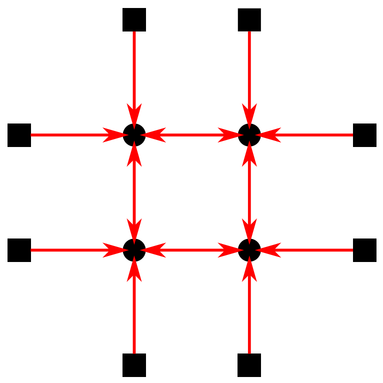


Back substitution

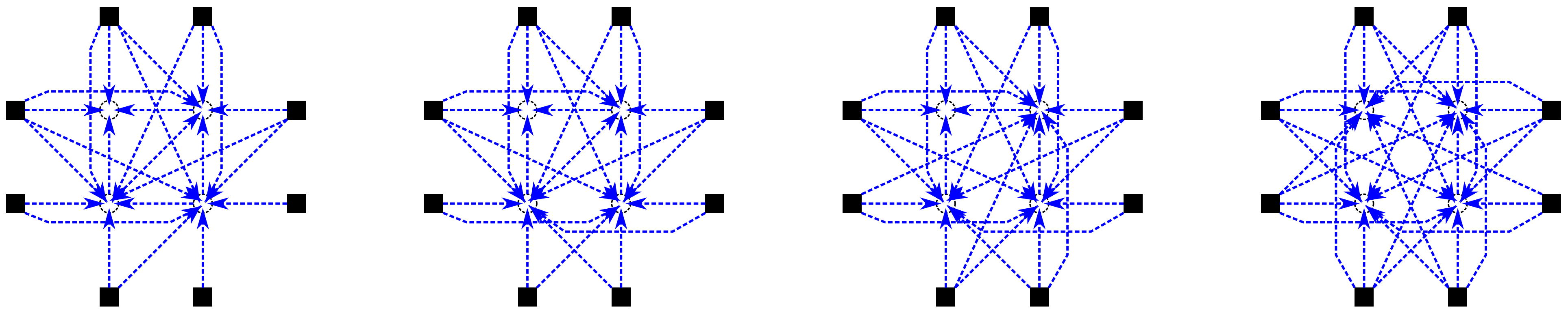
- Variable elimination did not eliminate the dependence of eliminated variables on other variables. Now is time to make those replacements, starting from the most recently eliminated variables.
- The computational complexity of back substitution is almost always cheaper than variable elimination.
- From the image below, the computational complexity is $O(a \cdot d)$ flops.



Variable Elimination



Back Substitution



Complexity analysis

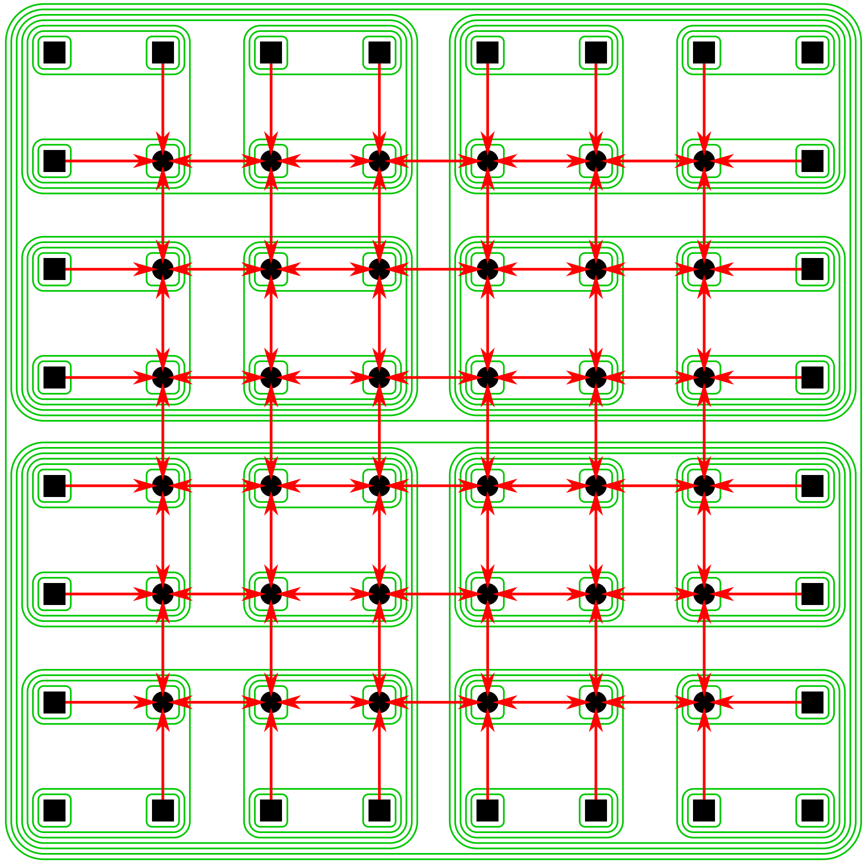
- Variable elimination proceeds row-by-row, followed by layer-by-layer.
- The computational complexity of variable elimination is $O(N^7)$.
  - Most of the $O(N^3)$ nodes require $O(N^4)$ flops to eliminate.
- The computational complexity of back substitution is $O(N^7)$.
  - Most of the $O(N^3)$ nodes depend on $O(N^2)$ other nodes, which each in turn have an expression that has a size of $O(N^2)$. The cost of substitution for each node is $O(N^4)$, resulting in a total cost of $O(N^7)$.

Complexity analysis

- Scenario description:
  - 3D cubic lattice where each side has dimensions of N.
  - Connections only exist between neighboring nodes.
  - Diagonal connections do not exist.
  - The conditioned variables are the surface of the cube.
  - The unknown variables are the interior of the cube.
- Variable elimination of each interior node will proceed in a row-by-row, layer-by-layer process.
- Approximately $4 \cdot N^7$ multiplication flops are required for variable elimination, assuming that $N > 40$.

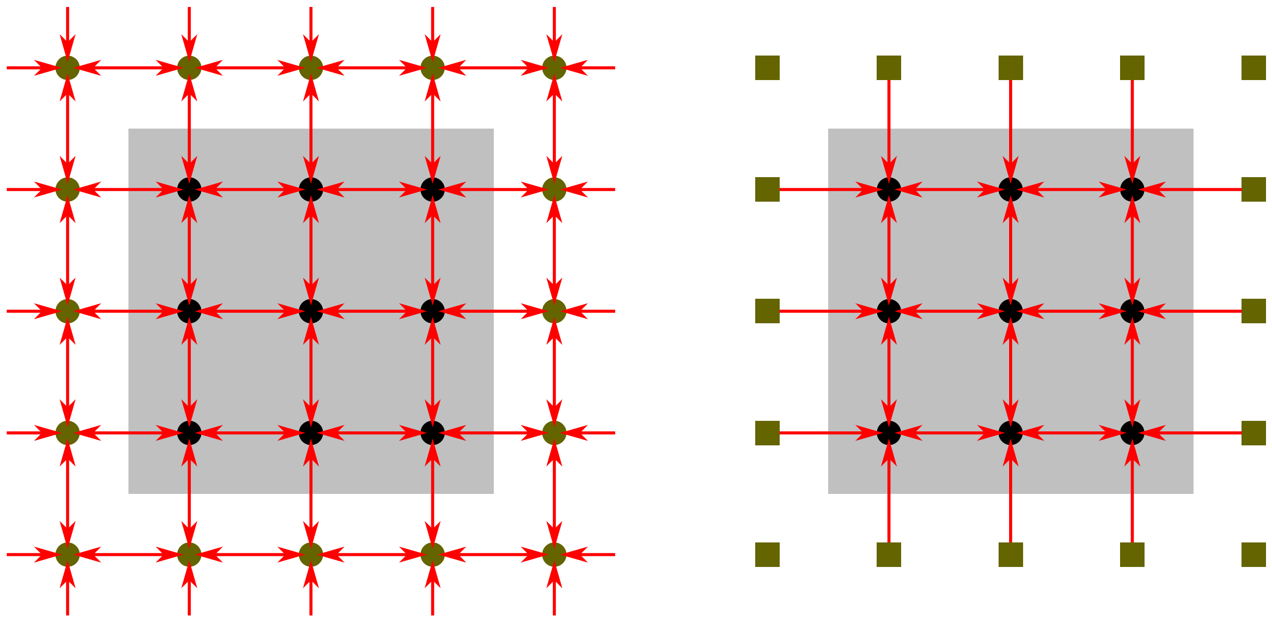
Nested dissection

- Nested dissection was first proposed in [1].
- Terminology from [2] will be used.
- Recursive partitioning of the set of nodes. Each set of nodes is called a **node cluster**.
- The node clusters that are not subject to further partitioning are **leaf node clusters**.



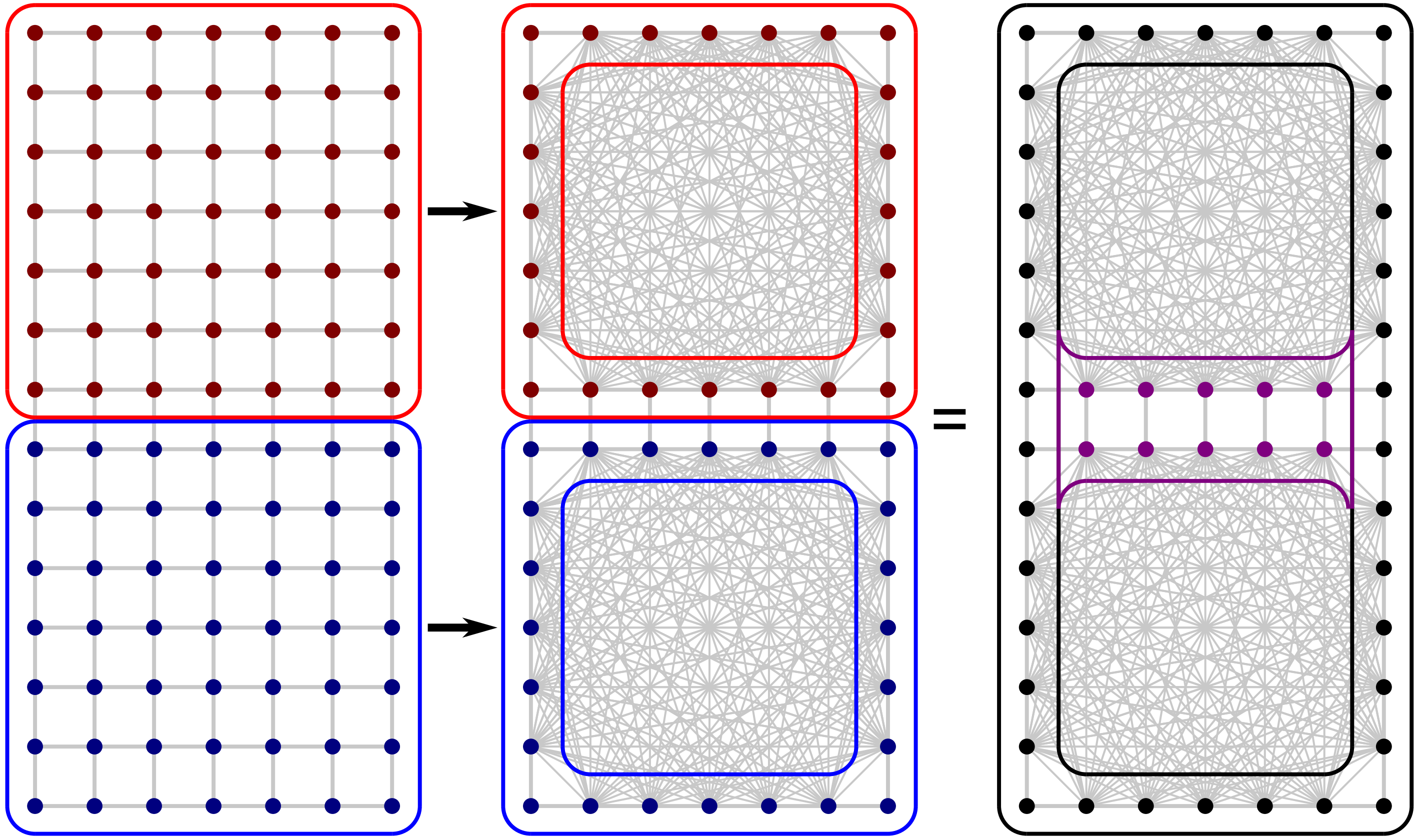
Boundary and Interior

- Nodes that have connections to nodes outside of the current cluster, or are conditioned, are considered **boundary nodes**.
- Non-boundary nodes are interior nodes.



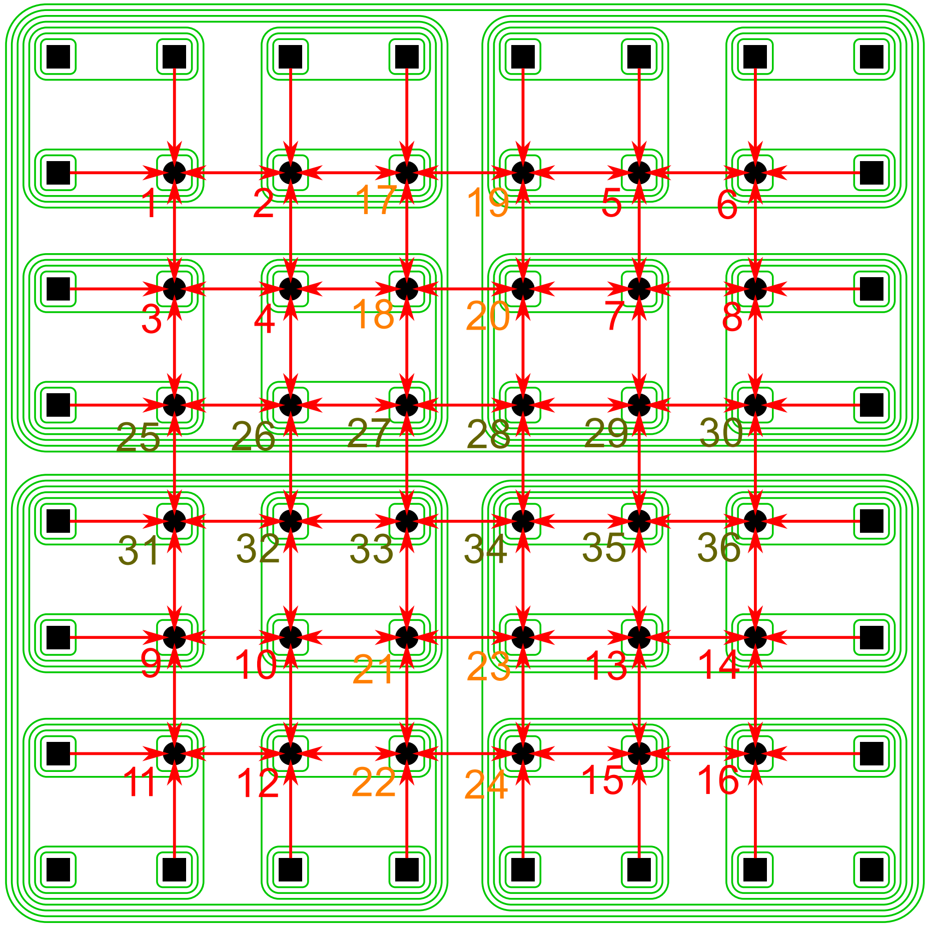
The private interior

- For non leaf node clusters, the **private interior** consists of the interior nodes that are not also part of the interior of either child node cluster.
- For leaf node clusters, the private interior is the interior.
- Elimination of the private interior follows the elimination of the interior of each child node cluster.

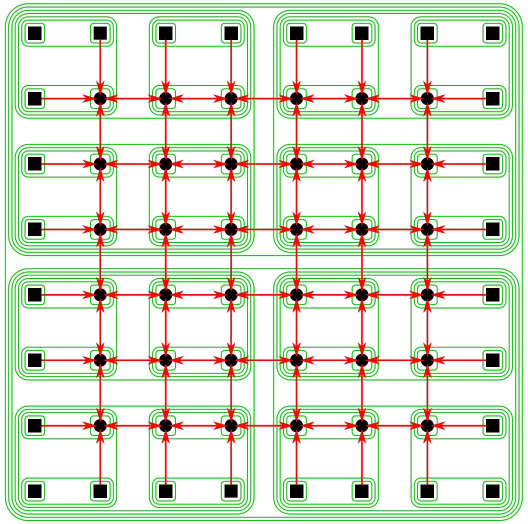


Variable elimination order

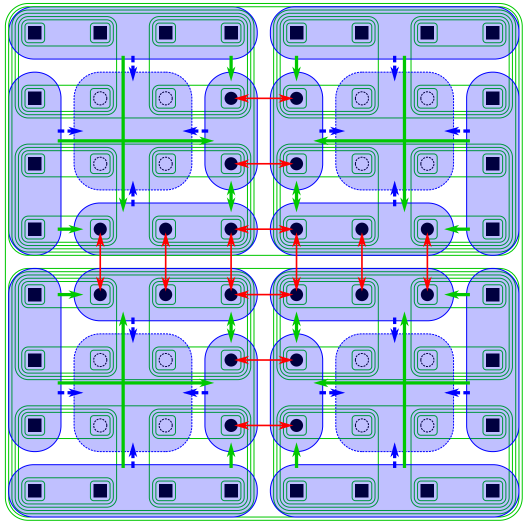
- Variables are eliminated starting with the interior of the leaf node clusters, followed by the private interiors of the parent node clusters until only the conditioned nodes in the root node cluster remain.



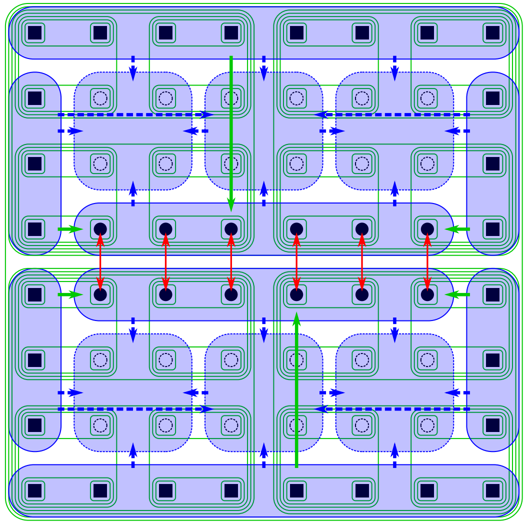
Example Nested Dissection Variable Elimination



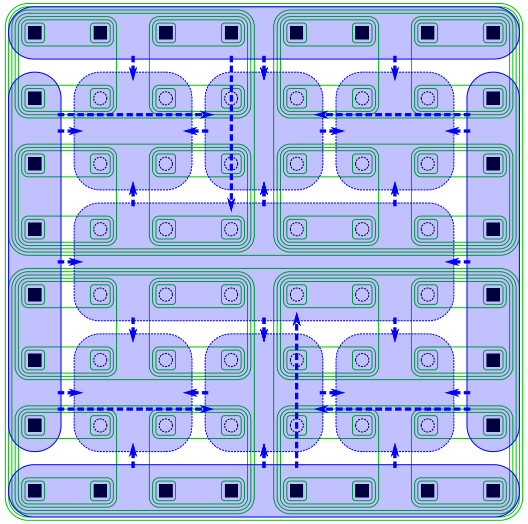
Original



Level #2



Level #1



Level #0

Complexity analysis

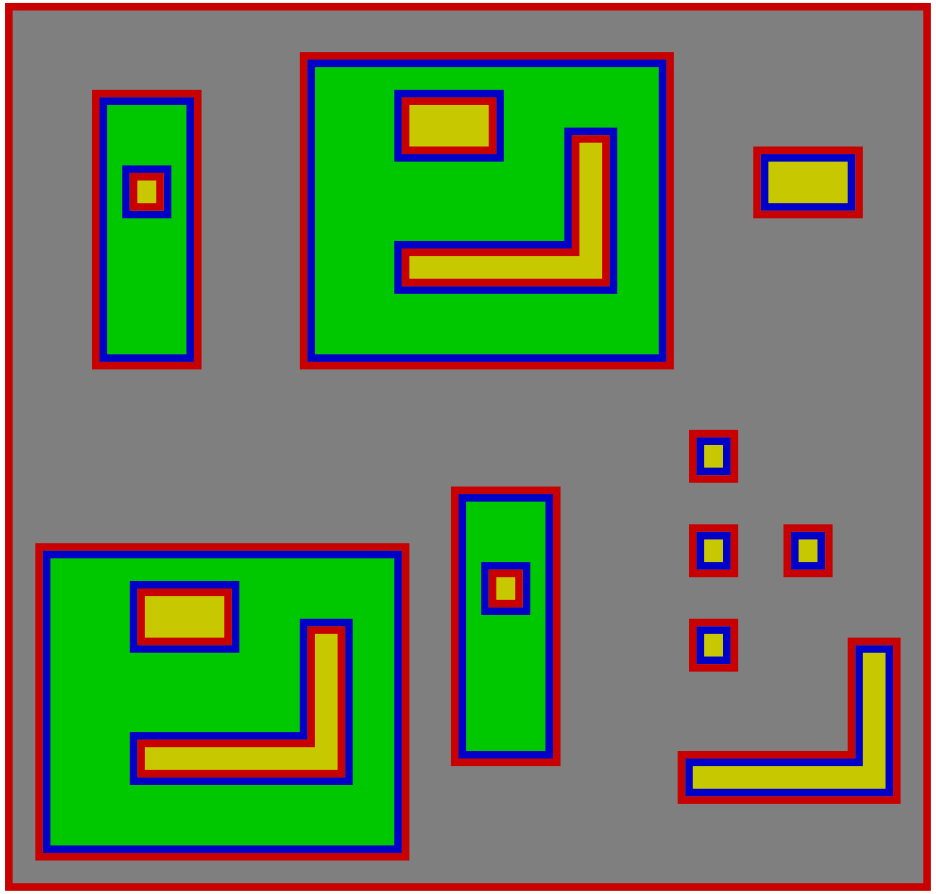
- The computational complexity of variable elimination is $O(N^6)$.
  - Eliminating the private interior of node cluster whose length is $O(M)$ costs $O(M^6)$.
- The computational complexity of back substitution is $O(N^6)$.
  - Evaluating the private interior of node cluster whose length is $O(M)$ costs $O(M^4 \cdot N^2)$.

Complexity analysis

- Scenario description:
  - 3D cubic lattice where each side has dimensions of N.
  - Connections only exist between neighboring nodes.
  - Diagonal connections do not exist.
  - The conditioned variables are the surface of the cube.
  - The unknown variables are the interior of the cube.
- Approximately $19 \cdot N^6$ multiplication flops are required for variable elimination, assuming that $N > 40$.

Reusability

- The calculations can be reused should the same node clusters appear multiple times.
- Node clusters can tightly wrap repeating components.



Summary

- The direct solution without any exploitation of the sparse structure costs $O(N^9)$ flops, resulting in $O(N^4)$ flops per entry of A.
- The naïve layer-by-layer exploitation of the sparse structure costs $O(N^7)$ flops, resulting in $O(N^2)$ flops per entry of A.
- Nested dissection costs $O(N^6)$ flops, resulting in $O(N)$ flops per entry of A. Moreover, the calculations can be stored and reused.
- From the specific cubic lattice,
    - Layer-by-layer variable elimination costs approximately $4 \cdot N^7$ multiplication flops ($N > 40$).
    - Nested dissection variable elimination costs approximately $19 \cdot N^6$ multiplication flops ($N > 40$).

References

- [1] George A. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis* 1973; 10(2):345–363.
- [2] Darve E, et al. A hybrid method for the parallel computation of Green’s functions. *Journal of Computational Physics* 2009; 228:5020–5039.

Any questions?