GRID EMT

SIMULATION INITIATIVE

Structures in EMT Linear System and Adaptive Schur Solver

Presented by David Gleich (Purdue University)
Joint work with Zhiyao Xu (Purdue),
Joseph Francis (ORNL), Phani Marthi (ORNL)

October 27, 2026



U.S. DEPARTMENT of ENERGY | Office of Electricity

Motivation

Solving linear systems fast in EMT simulation involves utilizing the structural property of the matrix, including:

- Block diagonal **(BD)** – fully decoupled structure
- Bordered Block Diagonal **(BBD)** – minimal coupling

This talk is about how graph and network techniques can be used to inform us the structure inside a linear system.

And how we can use that to accelerate a GPU solve

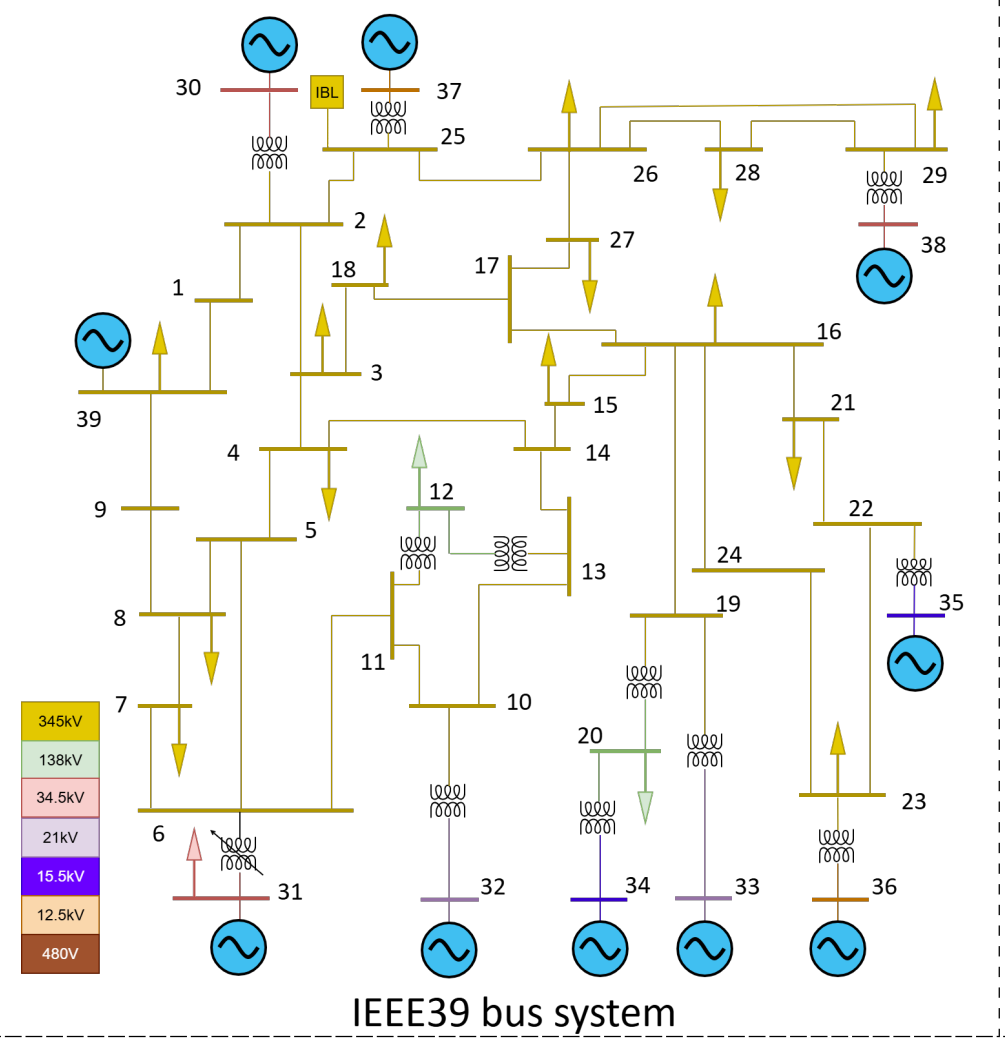
Linear system viewed as graph

(1)
(2)
(3)
(4)
(5)

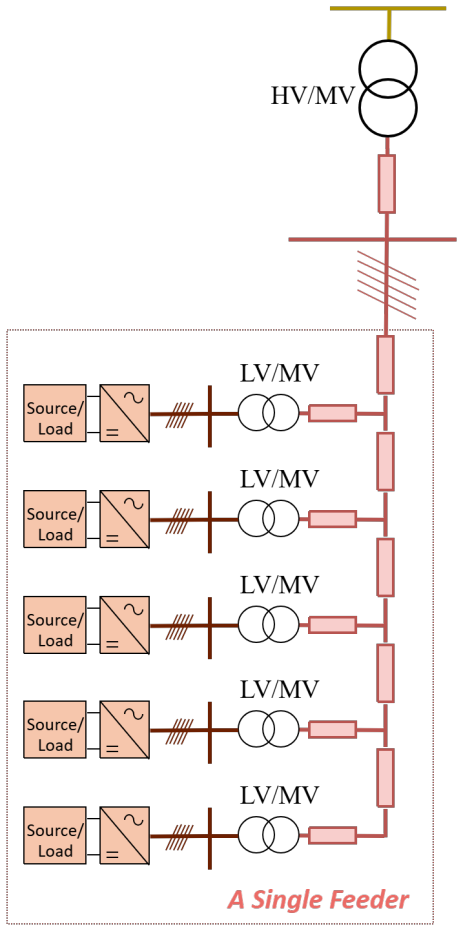
We can view linear system as a network where:

- nodes are equations
- links are shared variables

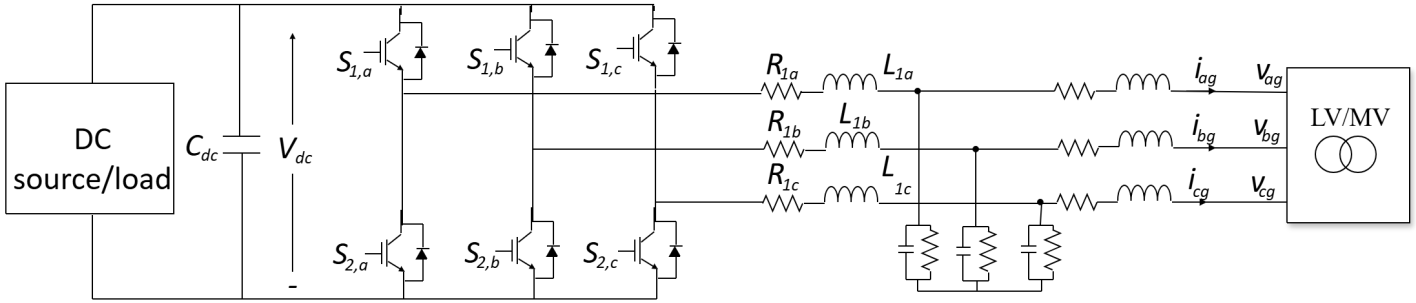
How most people look at these systems



IEEE39 bus system



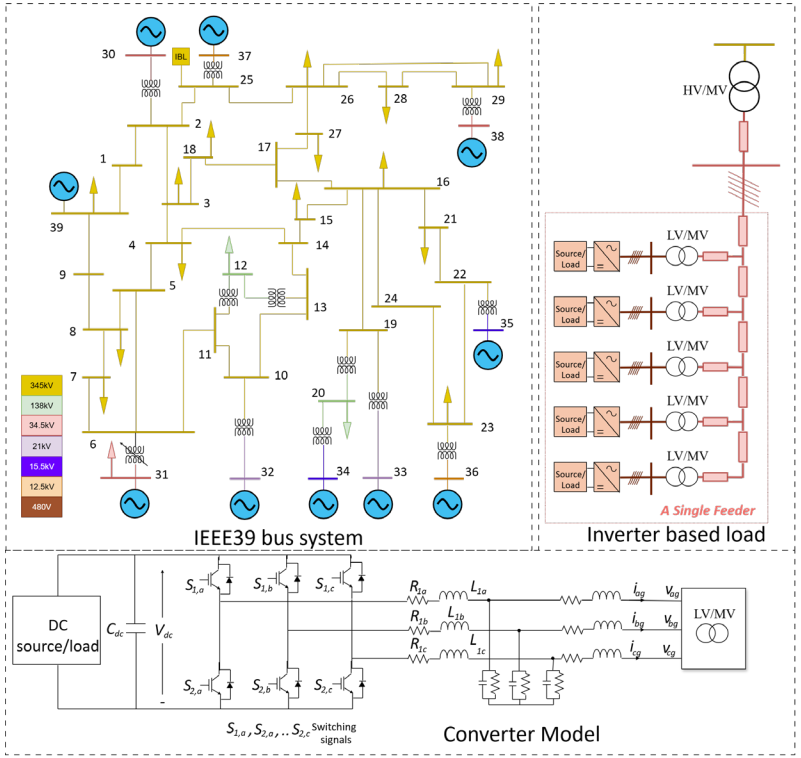
Inverter based load

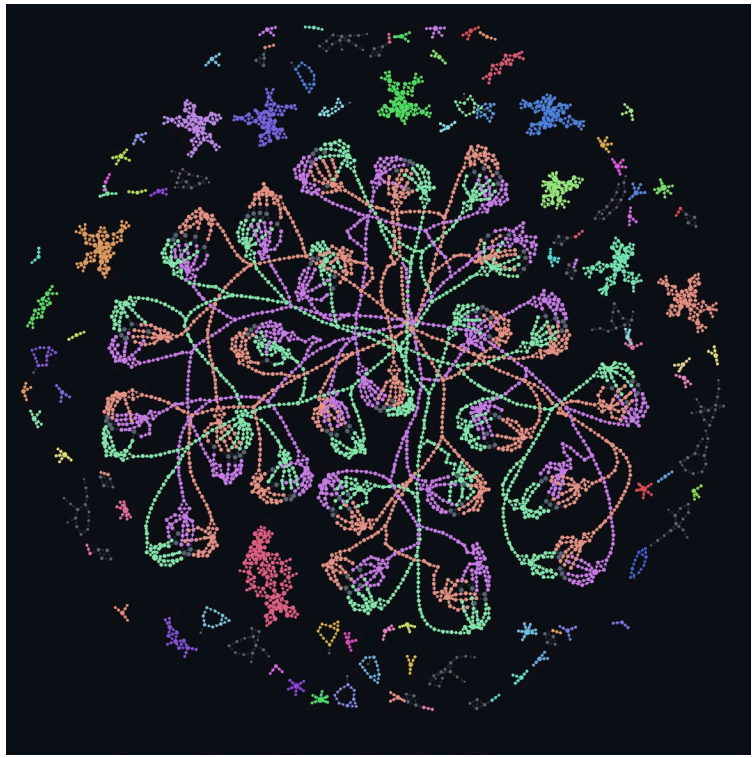


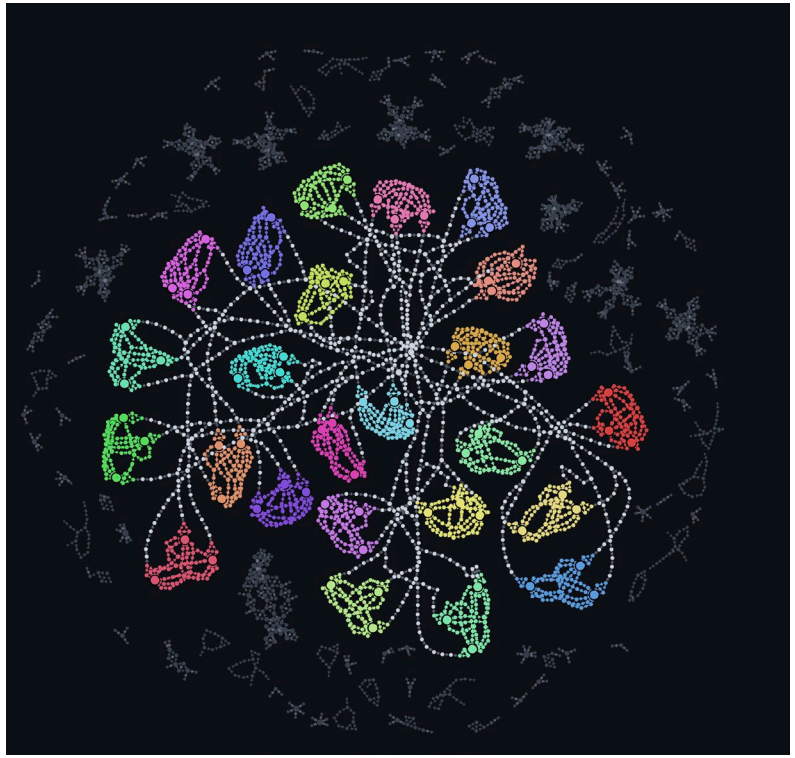
Converter Model

U.S. DEPARTMENT of ENERGY | Office of Electricity

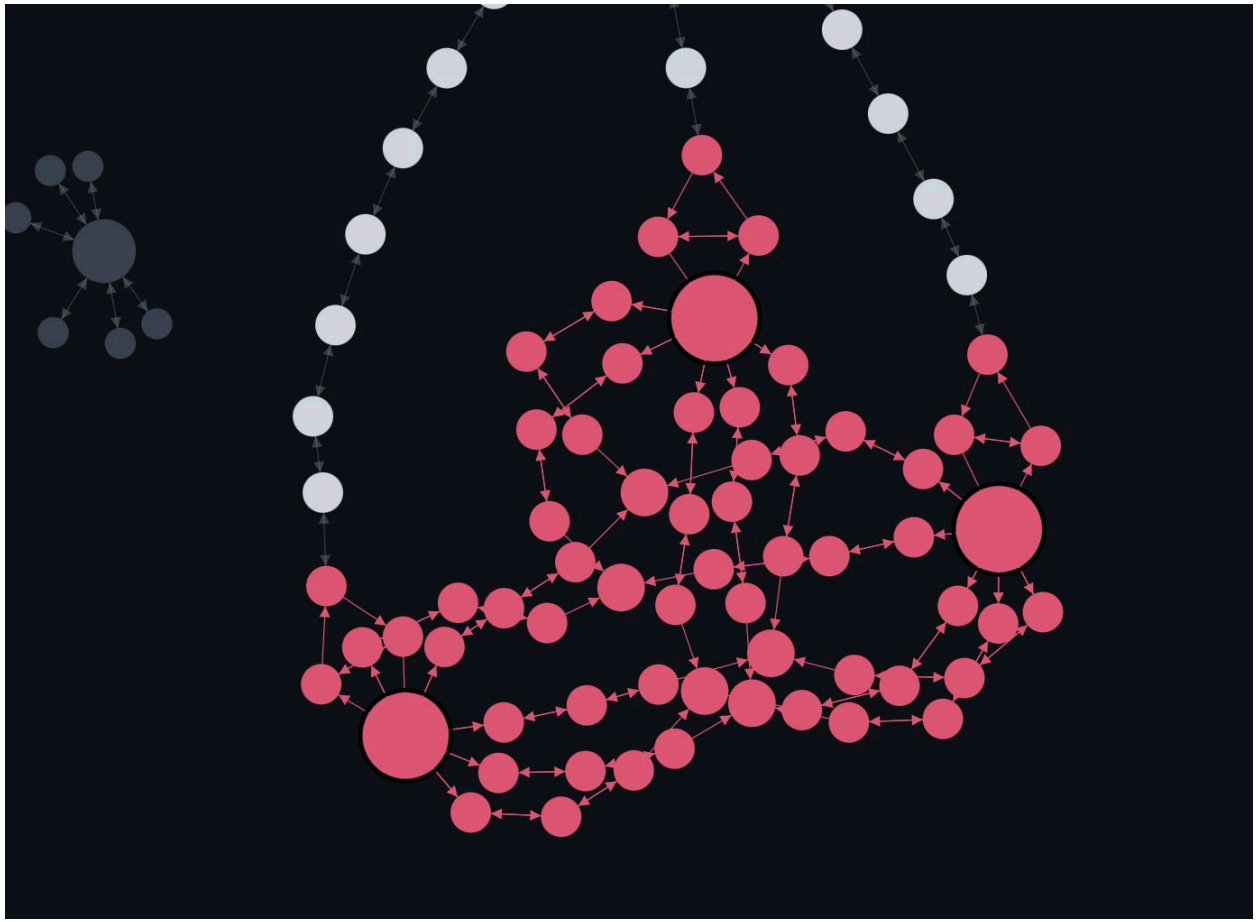
How we look at this system! IEEE39 with IBL



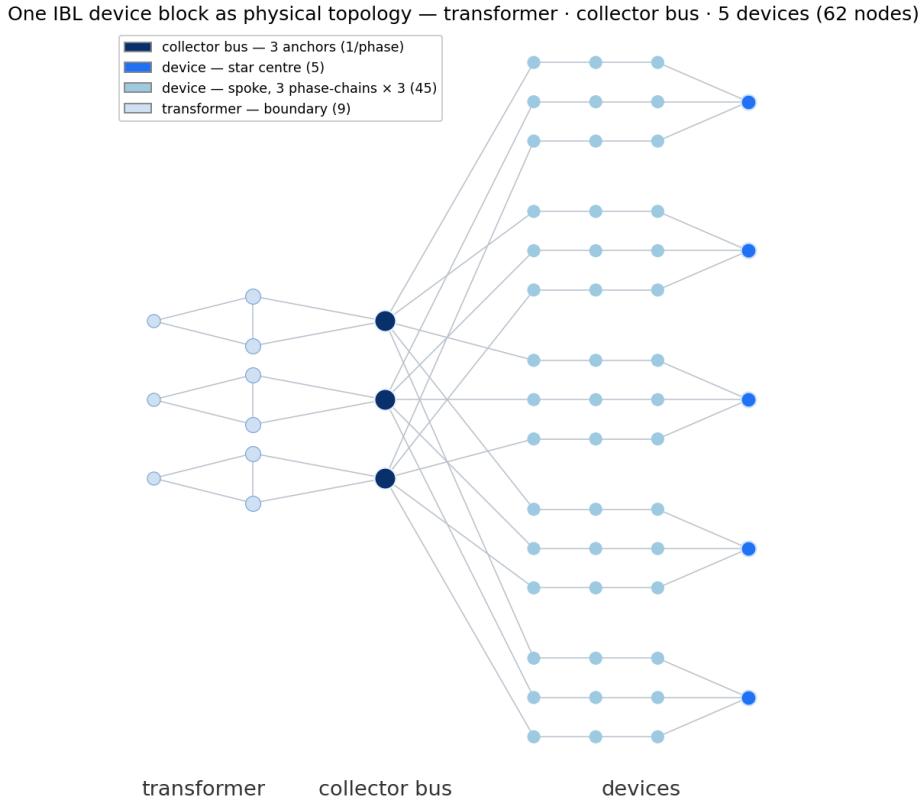




Example: repeating 62x62 pieces

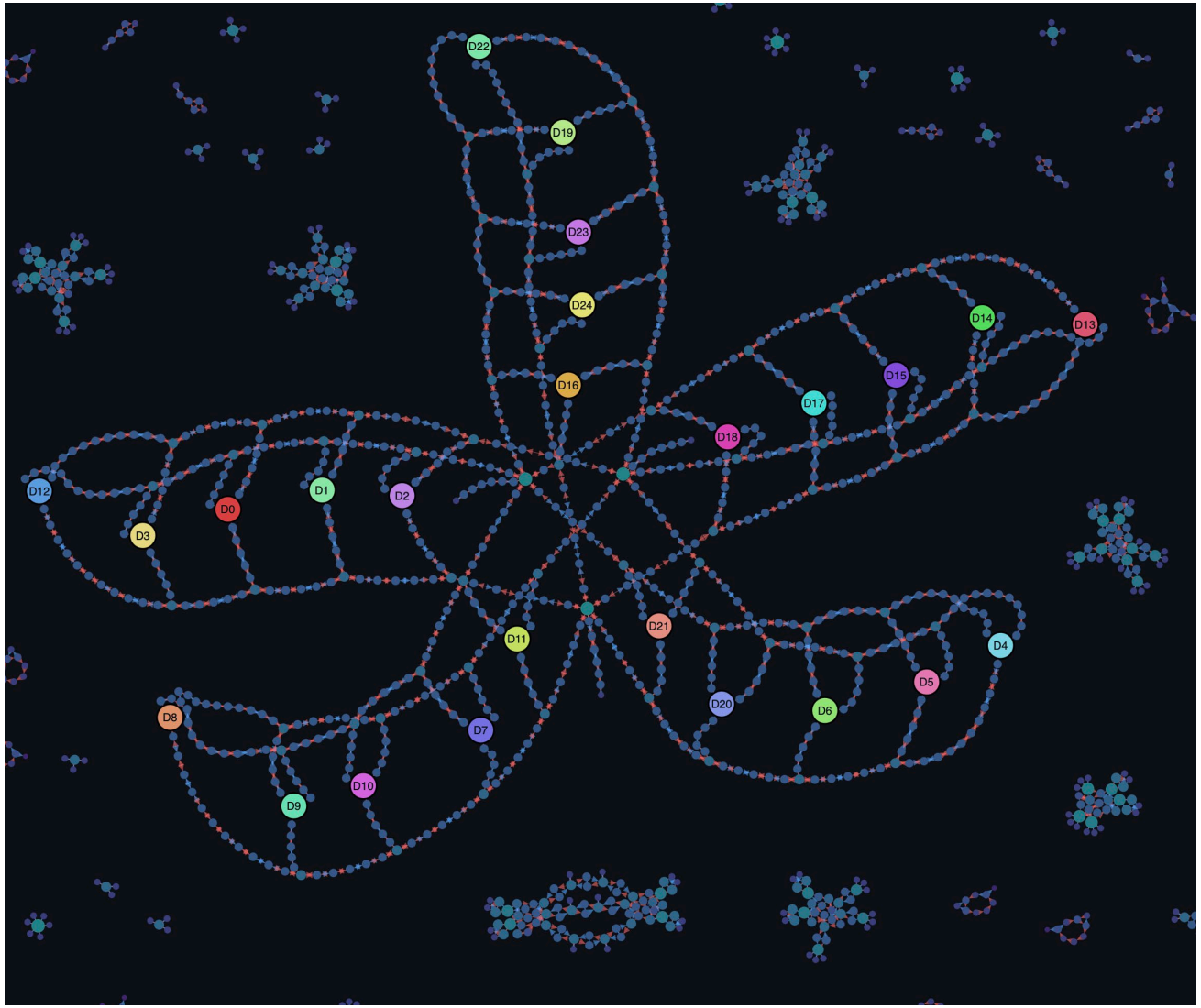


These are the inverters at the end of the distribution chain.



The grid topology and structure is visible in these visualizations

Here is the 5 leg feeder hosting the 125 inverters



Knowing the structure

Linear solver's knowledge of system structure is:

Given to us by the producer of the matrix

or

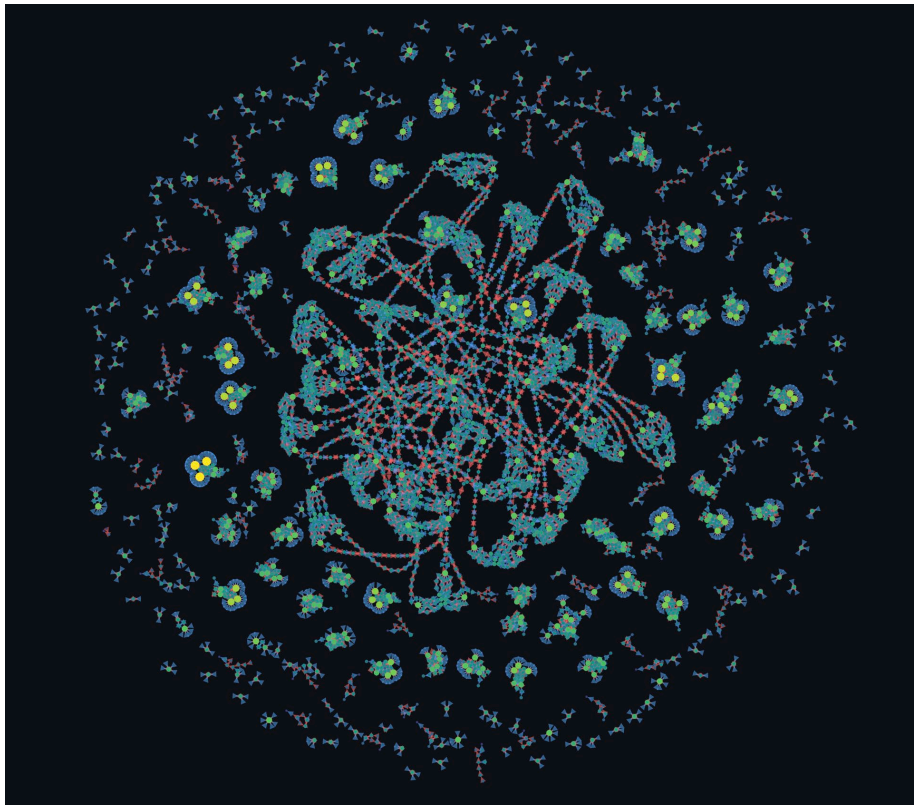
Discovered by us, which takes computation per new matrix

(it may be able to be updated)

Analyze Structure

BFS to get connected components. (reachability)

Reordering to get BD matrix structure.



Analyze structure

Inside large blocks, set high-degree nodes as hub / connection-line candidates (most coupling), set the rest as interior blocks (least coupling).

Results in a BBD / arrowhead matrix:

Bordered Block Diagonal / Block Arrowhead Matrix

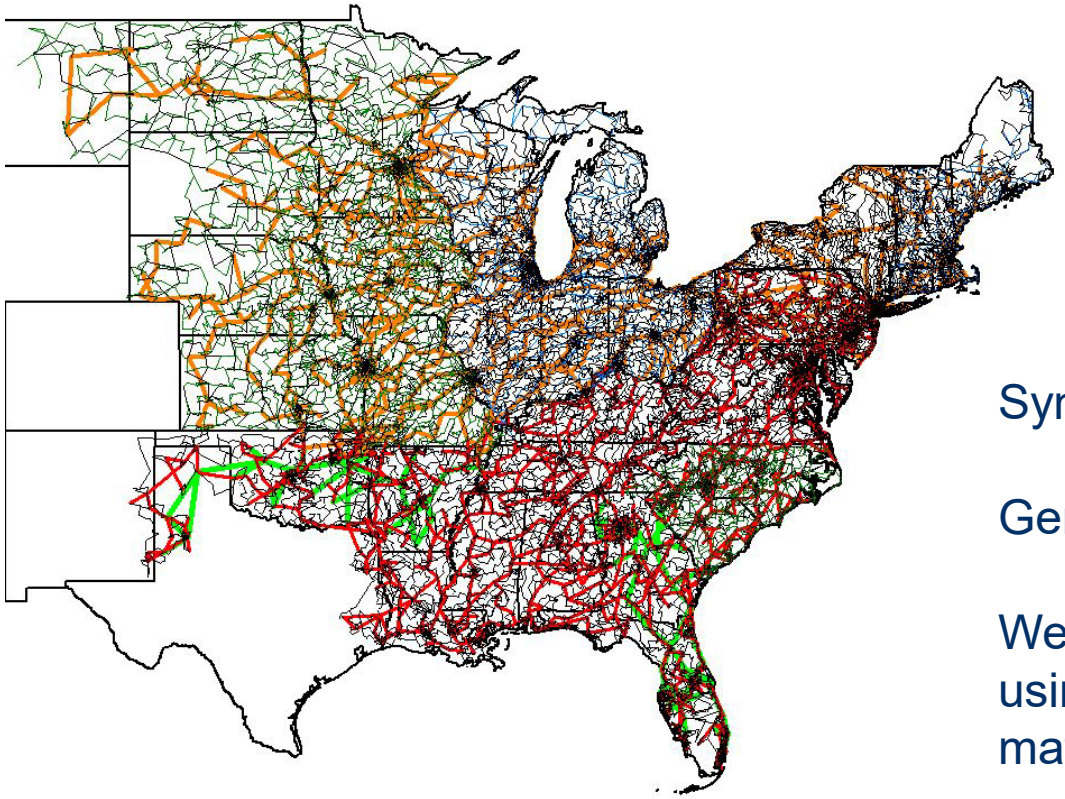
Analyze structure

The overall system could be a BD of BBD matrices, that looks like

Upon entry value change, only affected blocks are updated.

A_i may have similar BBD structure or even $A_i = A_j$

ACTIVSg70k: a nation scale synthetic grid



Synthetic 70k bus grid of eastern U.S.

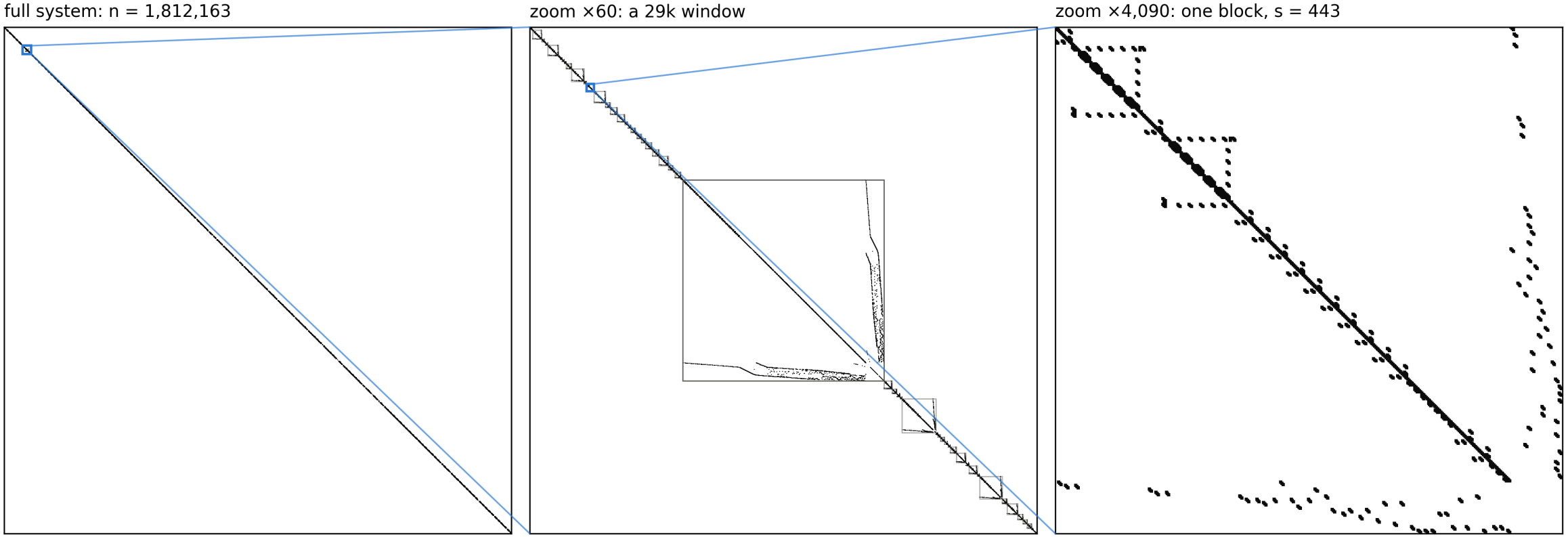
Generation and load profiles are similar to real public data.

We assemble a 1.8 million by 1.8 million linear system from it using Claude to generate EMT-like structures similar to matrices from REINTEGRATE.

ACTIVSg70k synthetic grid, Texas A&M Electric Grid Test Case Repository (electricgrids.engr.tamu.edu)

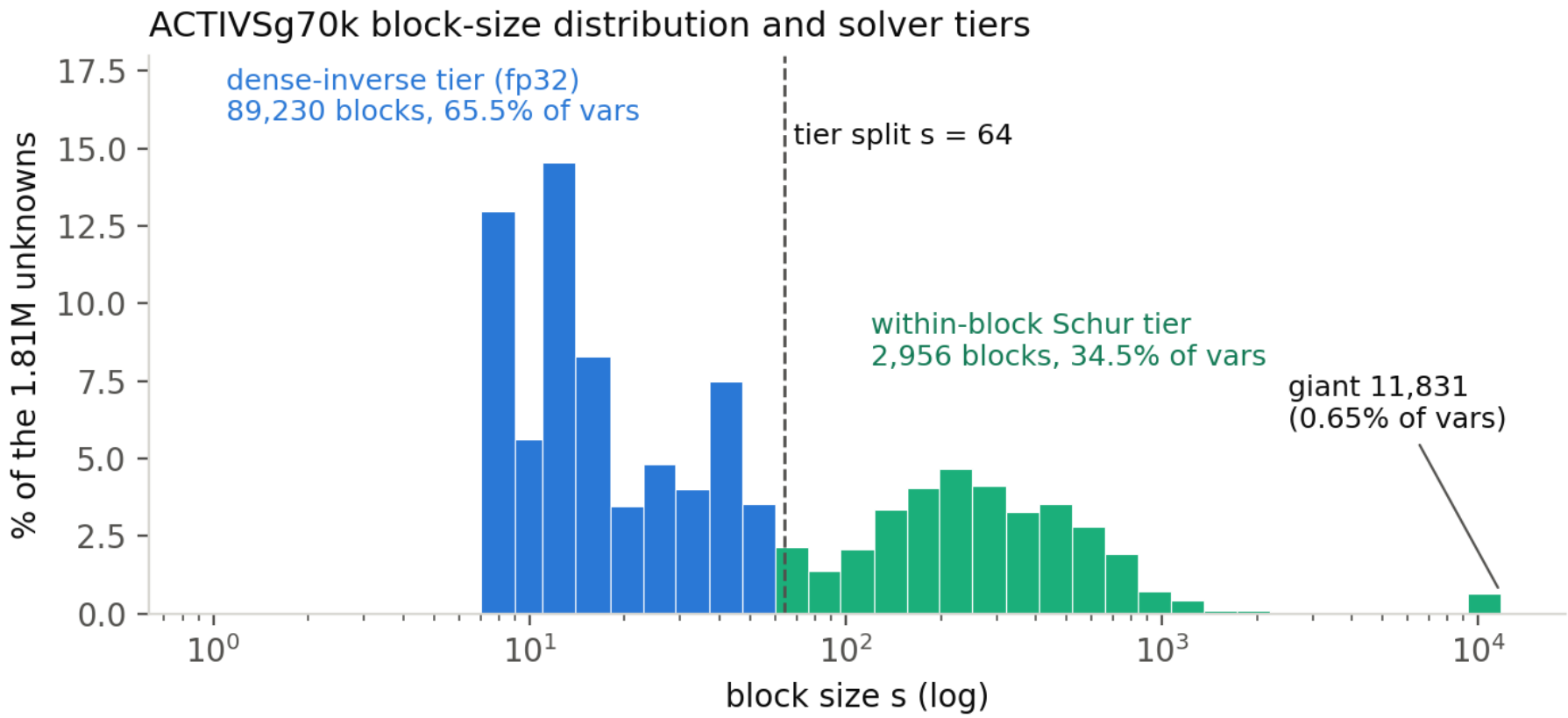
Linear system from ACTIVSg70k

Very sparse, averages less than 3 entries per row



This matrix is systematically generated with AI assistance from Texas A&M ACTIVSg70k synthetic powergrid. It may reflect the structure, sparsity, and other properties of smaller EMT linear systems generated by various DAE methods.

Linear system from ACTIVSg70k



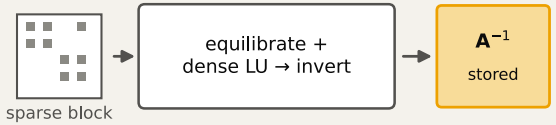
Solver pipeline

SETUP — once per topology, on CPU (OpenMP), all arithmetic in fp64

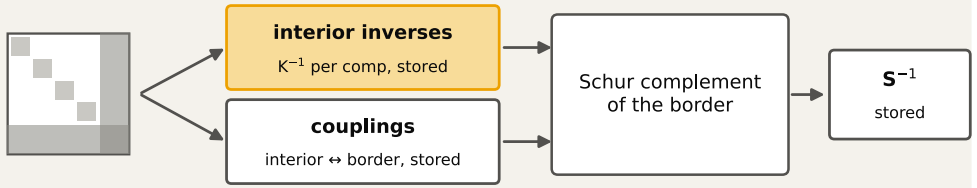
STRUCTURE — today: BFS finds decoupled blocks; in large ones, degree ≥ 5 → border, another BFS splits the interior

— future: the assembler reports structure and unit scales

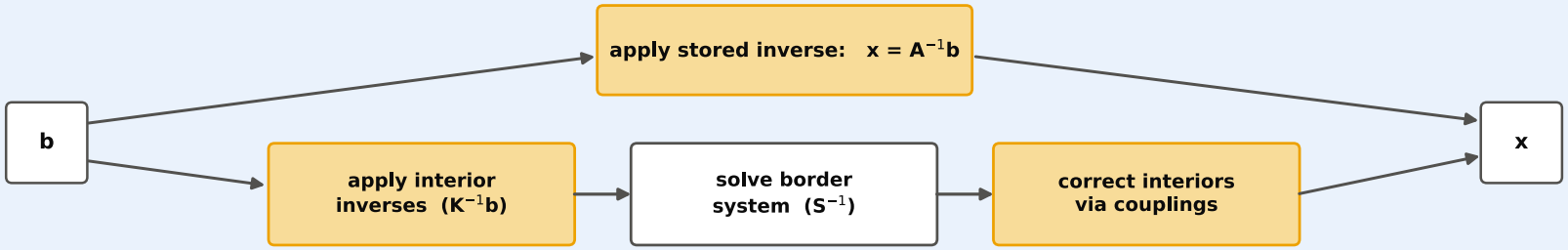
TIER A blocks ≤ 64 — dense inverse



TIER B blocks > 64 — within-block Schur



PER-RHS — every time step, on GPU as one CUDA graph, streaming the stored data once per solve



data stored on the GPU in fp32

Solver pipeline

A CUDA graph encodes:

- Parallel block-matrix-inverse application
- Border-coupling-inverse application
- Interior correction based on coupling

within a single launch.

Sparse blocks exploit sparsity, dense inverse application promotes parallelism.

Key properties of solver

- 0.55s fresh setup (with CUDA warm in context) from CSR, mostly CPU;
- 255μs per solve on GPU (RHS on GPU).
- KLU solve was ~1000μs per RHS on CPU.
- CUDA GPU accelerates matrix routines (highly parallel blocks, explicit inverse).
- Despite dense storage and dense routine, sparsity is exploited by the BD and residing BBD structure.
- GPU Bandwidth, not throughput, is the bottleneck.
- FP32-stored inverse doubles bandwidth performance, against FP64.

Changing the matrix without redoing all of the analysis

Changes in a block applies a low rank update to the Schur complement as:

$$S_{new} = S + U \cdot V$$

The stored inverse of the updated Schur complement can be update in place as:

$$S_{\text{new}}^{-1} = S^{-1} - S^{-1}U(I + VS^{-1}U)^{-1}VS^{-1}$$

In-place updating reduces the time to update five interior variables in the largest block from 33ms to 6ms.

Current programming interface

- We have designed a KLU-like interface to make it easy to put into REINTEGRATE. We still need to add the “update” features.

```
GblsSolver* Gbls_Create (config);                    // Create solver instance with configuration
int         Gbls_Setup  (slv, n, rowptr, colidx, vals);  // Setup, or update existing setup
int         Gbls_Solve  (slv, b, x);                 // Solve for given RHS
void        Gbls_Free   (slv);
```

Learning about EMT matrices: Hardware constrained performance and system limits

- This formulation of problem roughly scales linearly with the VRAM bandwidth.
- If producer and consumer of RHS is on CPU, the PCIe roundtrip **triples** the per RHS solve time.
- A unified memory computing unit with HBM can: eliminate PCIe roundtrip, and linearly accelerate per solve performance. E.g. AMD instinct, NVIDIA GB200

Next Steps

- Design a general structure communication interface.
- Save duplicated inverses among identical blocks (fewer in-flight bytes).
- Precomputed generator block configuration for faster update.
- Test the performance on unified memory computing unit.

Thank You

Acknowledgement: The authors thank U.S. Department of Energy (DOE) Office of Electricity (OE) Advanced Grid Modeling (AGM) and the EMT initiative for their invaluable guidance and kind support in this work!

Legal Disclaimer: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.



U.S. DEPARTMENT of ENERGY

Office of Electricity