Structure-Aware Linear Solvers for Matrix-Free High-Order Finite-Element Simulations on GPUs

Gourab Panigrahi and Phani Motamarri
Department of Computational and Data Sciences, Indian Institute of Science, Bangalore 560012, India
pgourab@iisc.ac.in, phanim@iisc.ac.in

***Abstract*—Higher-order finite-element (FE) simulations solve the same discretized elliptic system repeatedly, inside implicit time integration, pressure projection, or the self-consistent-field loop of an electronic-structure code. Production codes use preconditioned conjugate gradient (PCG), whose cost is set by an iteration count that climbs with polynomial order and mesh grading, and every iteration carries one latency-bound global reduction that degrades with rank count. We report two GPU solvers for the FE Poisson and shifted-Helmholtz operator, each selected at setup from the algebraic structure that the mesh and the operator retain.**

I. Exact Fast Diagonalization on Tensor-Product Meshes

When the finite-element nodes form a tensor product, uniform or graded, and the coefficient is constant, $\mathbf{A} = \mathbf{K}_{3D} + \alpha \mathbf{M}_{3D}$ is a sum of Kronecker products of the *assembled global* one-dimensional stiffness and mass matrices. One small generalized eigenproblem per direction, lifted through the mixed-product rule, diagonalizes every term at once, giving the inverse in closed form. One apply is six GEMMs and one diagonal scaling at $\mathcal{O}(N^{4/3})$ operations. It is exact up to round-off, needs no iterations, preconditioner or reductions, and does not depend on the condition number. Assembly and inversion do not commute, so the assembled matrices are diagonalized, not the element ones. Fully periodic problems are singular, so we set the reciprocal of the single zero eigenvalue to zero, which applies the pseudoinverse and returns the zero-integral solution. Pinning a degree of freedom or adding a dense mean-value row would destroy the Kronecker structure, whereas the zeroed reciprocal keeps it.

Distribution is harder: a contraction along one direction needs whole lines of the field on one rank, and no slab layout makes all three local. Our solver redistributes the field between contractions rather than the mesh, so one apply is seven stages and four collectives, with two MPI transposes between k-slab and i-slab layouts. The index maps are exchanged once at setup, and the per-rank working set is $\mathcal{O}(N/P)$. On the GPU, CUDA-graph replay collapses the kernel launches into one, and a padded on-node all-to-all with an inter-node variable-payload path cuts redistribution from 37 to 1.31 ms.

Measured inside DFT-FE against converged Chebyshev–Jacobi PCG at its own default tolerances, one apply is 138× faster on a 561-atom aluminium cluster with uniform Dirichlet boundaries, 463× on a 12.5M-degree-of-freedom lithium-garnet solid-electrolyte interface with mixed boundary conditions, and 4570× on a geometrically graded slab, so the advantage grows with the conditioning that degrades CG. Against the Chebyshev–Jacobi p-multigrid solver of the next section, one apply is still 14–140× faster on uniform Dirichlet meshes on Frontier. From 4 to 32 nodes, per-rank memory halves with every rank doubling, 22.9 to 2.8 MB, while the PCG iteration count stays flat.

II. Chebyshev–Jacobi p-Multigrid Where the Structure Is Gone

Atom-centred refinement and the variable coefficient of continuum solvation each destroy the Kronecker form, so the method must become iterative. We keep tensor-product kernels and no global reductions, and give up only exactness. The smoother is a Chebyshev polynomial in the Jacobi-scaled operator, built from operator applies alone, so it needs no inner products or collectives. Coarsening in polynomial order rather than in the mesh lowers the largest eigenvalue level by level, so the modes the smoother cannot reach become the next level's own top decade. The ladder $7 \to 4 \to 2 \to 1$, or $7 \to 4 \to 1$ where measured faster, runs on the same cells, partition and ghost pattern, with a coarsest level solved on every rank. The V-cycle stays on the device inside a fused conjugate gradient with one all-reduce of three scalars per outer iteration.

On an atom-refined octree with hanging nodes the V-cycle converges in 5, 6 and 7 iterations at orders 2, 4 and 7, against 106, 326 and 668 for Jacobi PCG. On a 304-atom protein in implicit solvent at 11.4M degrees of freedom it takes 63× fewer iterations and 3.2× less wall time than the best point-preconditioned solver. On Frontier the same solve strong-scales to 1024 MI250X GCDs at 5.5–5.9 ms per CG iteration.

III. Scope and Limits

Fewer iterations do not translate one for one into wall time: on the solvation operator the coarsest space is too large to invert densely, and one V-cycle costs about 20 Jacobi iterations, so the 3.2× wall-time gain trails the 63× iteration drop. Iteration and collective counts are the robust signal. The distributed fast-diagonalization apply turns communication-bound below roughly 10^5 degrees of freedom per rank. The benchmarks come from density functional theory, but neither method assumes more than the operator and the mesh, so both carry over to any higher-order FE application.

IV. Status

Both parts are unpublished work in preparation, and no result here has appeared elsewhere.

Acknowledgment

Parts of this summary and of the poster text were drafted with the assistance of Claude, a large language model by Anthropic [1]. The authors verified all content against their data and are accountable for it.

References

[1] Anthropic, *Claude* (large language model), https://www.anthropic.com/claude, 2026.