Hardware-Aware Matrix-Free Algorithms for Accelerating Large-Scale Finite-Element Simulations on Distributed CPU and GPU Architectures

Gourab Panigrahi | Advisor: Prof. Phani Motamarri | Department of Computational and Data Sciences, Indian Institute of Science, Bangalore
pgourab@iisc.ac.in | SC26 Doctoral Showcase

DFT-FE code | Website | LinkedIn

1. Two kernels dominate at scale

Finite-element (FE) simulations spend most of their time in two kernels.

PDE on Ω → FE discretization → **A**, sparse, higher order

Eigenproblems
subspace iteration (ChFSI)
needs **AX** on 10^3–10^4 **vectors**

Linear systems
preconditioned Krylov methods
need **Au** over 10^2–10^4 **iterations**

Data movement, not FLOPs
memory hierarchy and interconnect set the time

Cost grows with order p
for both classical ways of applying **A**

Applying $\mathbf{A}^{(e)}$	Stored	FLOPs/cell
Assembled sparse **A**	whole matrix	memory bound
Cell-matrix (per element)	$n_p^3 \times n_p^3 \times n_c$	$O(n_p^6)$
Matrix-free (this work)	one $n_q \times n_p$ pair	$O(n_p^4)$

Matrix-free storage is **independent of cell and vector counts**.

2. Requirements beyond single-vector codes

deal.II, MFEM, libCEED and Nek5000 target a *scalar, real* operator on a *single* vector. Eigensolvers also need:

Multivectors
10^3–10^4 vectors at once

Real and complex
one code path

Local and nonlocal
both operator parts

Constraints
hanging-node, periodic, Dirichlet, no atomics

Objective. Matrix-free operator actions and linear solves for higher-order FE, from one node to exascale.

3. Common methodology

Sum factorization: the cell matrix is never formed.

$$\mathbf{A}^{(e)}\mathbf{X}^{(e,t)} = \mathbf{N}^{\mathrm{T}}\Big([\tilde{\mathbf{D}}^{(s)}]^{\mathrm{T}}[\mathbf{G}^{(s,d)}][\tilde{\mathbf{D}}^{(s)}] + \mathbf{G}^{\kappa}\Big)\mathbf{N}\mathbf{X}^{(e,t)}$$

Factoring out **N**: **24 → 12** 1D contractions per cell.

Multivector X, $c\,n_v$ vectors per DoF

Level-1, $i_b^{(1)} = 1$ | DoF 1 | DoF 1 | ⋯ | DoF m | DoF m |
Level-1, $i_b^{(1)} = 2$ | DoF 1 | DoF 1 | ⋯ | DoF m | DoF m |
. . .
Level-1, $i_b^{(1)} = n_b^{(1)}$ | DoF 1 | DoF 1 | ⋯ | DoF m | DoF m |

Complex, $c = 2$: real | imag Real, $c = 1$: real | real

Multilevel batched layout, vectors fastest: one layout for real and complex.

Four design choices recur in every chapter:

Batching
one SIMD lane or GPU thread per b vectors of one DoF

Even–odd decomposition
symmetric 1D matrices, **half the contraction FLOPs**

Fusion
one launch per cell batch, **constant memory**, registers, overlapped ghost exchange

Mixed precision
FP32, TF32 and BF16 where the outer iteration allows

Part I, AX → Helmholtz **(4)** → Kohn–Sham CPU **(5)** → Kohn–Sham GPU **(6)**
Part II, Au = f → tensor-product mesh: FD **(7)** → adaptive mesh: CJ p-MG **(8)**

Platforms used

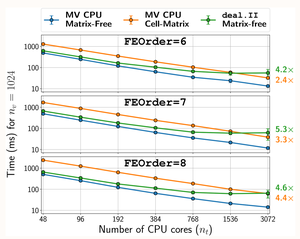
Summit — OLCF, NVIDIA V100
Polaris — ALCF, NVIDIA A100
Aurora — ALCF, Intel Max GPU
Frontier — OLCF, MI250X, EPYC
Fugaku — RIKEN, Fujitsu A64FX
Pravega — IISc, Intel Xeon

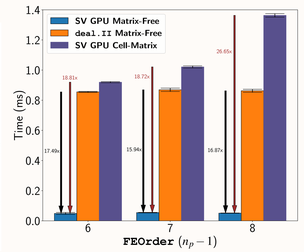
Part I | Operator action inside eigensolvers

4. Batched FE Helmholtz matrix–multivector products

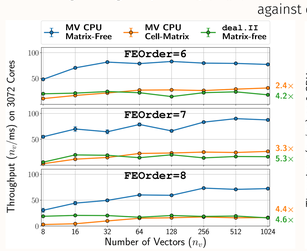
CPU and GPU, JPDC 192, 104925 (2024)

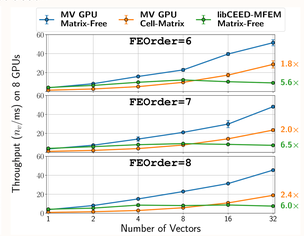
- CPU: tuned batch sizes, even–odd decomposition, SIMD
- GPU: kernel fusion, constant-memory shape functions, register blocking





CPU strong scaling at n_v = **1024**, 48–3072 Xeon cores (4.7× at 48). GPU single vector on one V100, against deal.II





Throughput vs n_v: 3072 Xeon cores, 8 A100s (Selene), with deal.II and libCEED–MFEM

2.8× / 2.6×
one V100 node (Summit) / one Xeon node, 1024 vectors

17× / 6.5×
vs single-vector deal.II / libCEED–MFEM

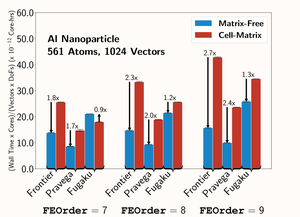
4.0× / 2.2×
end-to-end ChFSI, one CPU / GPU node

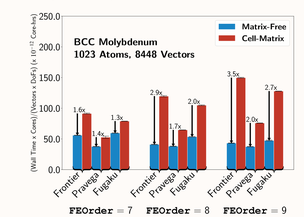
Poisson solves of the **SC'23 ACM Gordon Bell Prize** study, run to **64,000 MI250X GCDs** on OLCF Frontier.

5. Kohn–Sham DFT Hamiltonian action on CPUs

CPC 329, 110397 (2026)

- Local *and* nonlocal actions as structured tensor contractions
- One real/complex layout, cached on **AVX2, AVX-512, SVE**
- Mixed precision with a residual-based ChFSI (R-ChFSI)





One code path on AMD EPYC (Frontier), Intel Xeon (Pravega) and Fujitsu A64FX (Fugaku).

2.9× / 3.6×
real / complex pseudopotential, vs cell-matrix

5.8×
all-electron operator action

4.4×
full eigensolve with R-ChFSI

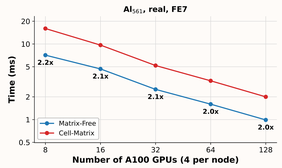
Gains grow with order and nonlocal share, except A64FX at order 7 (0.9×).

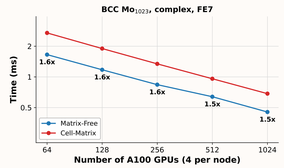
6. Kohn–Sham Hamiltonian action on GPUs

IEEE TPDS (under review)

The full Hamiltonian action in **one fused kernel**.

- Register–shared-memory sum factorization, even–odd
- Constraints resolved **inline**, no atomics
- **Morton (Z-order)** partition, **BF16** ghost exchange overlapped with compute
- **TF32** tensor cores on the nonlocal projectors, `cp.async` on A100



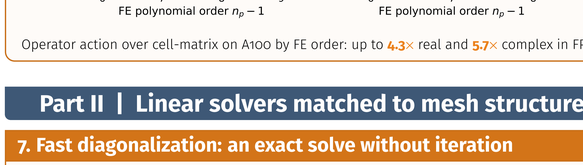


R-ChFSI filtering step, strong scaling on Polaris A100s, FP32, FE order 7: **2.0–2.2×** for Al_{561} on 8 to 128 GPUs, **1.5–1.6×** for Mo_{1023} out to 1024.

2.2–3.7×
real operator action, V100

3.3–5.4×
complex operator action, V100

24–32×
constraint path vs atomics



Operator action over cell-matrix on A100 by FE order: up to **4.3×** real and **5.7×** complex in FP32.

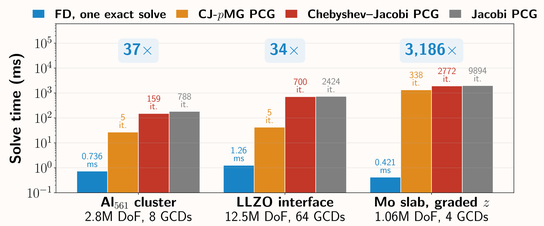
Part II | Linear solvers matched to mesh structure

7. Fast diagonalization: an exact solve without iteration

in preparation

On a tensor-product mesh the FE Laplacian is a sum of Kronecker products, and its *exact* inverse is one sequence of contractions.

- Distributed GPU solver, $O(N^{4/3})$ FLOPs, $O(N/P)$ memory per rank
- Dirichlet, periodic and semi-periodic boundaries
- Fully periodic: zero reciprocal for the zero eigenvalue gives the pseudoinverse and keeps the Kronecker form



One FD solve vs PCG to 10^{-10} on Frontier MI250X, badges vs CJ-pMG.

34–3186×
vs CJ-pMG (block 8), Frontier

138–4570×
vs Chebyshev–Jacobi PCG, Polaris

22.9 → 2.8 MB
per rank, LLZO, 4 to 32 nodes

FD's cost is independent of conditioning. Residual 1.4×10^{-14} (Al_{561}), 9.4×10^{-14} (LLZO, neutral RHS), 9.5×10^{-10} (graded slab).

8. Exascale iterative solves on general meshes

SC'26 (accepted) and in preparation

Adaptive refinement loses the tensor-product mesh, so the solve stays iterative, with the whole conjugate gradient (CG) loop on the device.

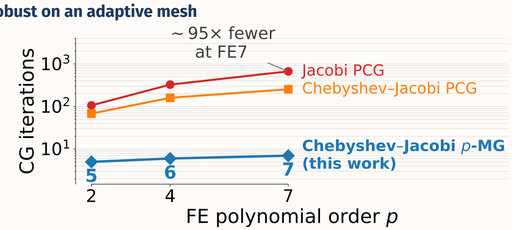
- Laplace kernel fused into CG, tree-reduction inner products
- **Chebyshev–Jacobi** smoothing: local products, no global reductions
- Device-native p-**multigrid**, 7 → 4 → 2 → 1

Poisson solve on 384 Aurora GPUs (SC'26)

deal.II 9.7.1 — 23.5 s
this work — 4.01 s — **5.86×**

Fe_3GeTe_2, 732 atoms, 16.4M DoF, p = 7, FP64, same solver, preconditioner and tolerance.

p-robust on an adaptive mesh



CG iterations, Al_{147} atom-refined octree with hanging nodes, A100s, same for 1 to 8 ranks: **95× fewer** than Jacobi at p = 7.

Variable coefficient: continuum solvation

Jacobi PCG — 1054 ms, 2086 it.
CJ-pMG — 327 ms, 33 it. — **3.2×**

Trp-cage in implicit solvent, 304 atoms, 11.4M DoF, 8 GPUs, where no fast direct solver exists. Strong-scales to **1024 MI250X GCDs** at 5.5–5.9 ms per iteration.

9. Critical assessment

- **Wall time gains less than iterations:** on solvation one V-cycle costs about 20 Jacobi iterations, so a **63×** iteration drop gives **3.2×** in time
- **Tensor cores help only part of the operator:** TF32 carries the nonlocal GEMMs, not the shallow-k local contractions
- **FD needs a tensor-product mesh**, hence block 8, and is communication-bound below about 10^5 DoF per rank
- **Speedups track the nonlocal fraction**, largest for all-electron
- **Time to solution, never FLOP rate:** matrix-free is DRAM-bound where cell-matrix is compute-bound
- **Baselines are DFT-FE's default solvers** or deal.II at the same solver and tolerance. SC'26 artifact: `doi:10.5281/zenodo.19817143`

10. Outlook

Block floating point
in the R-ChFSI filter, graph-captured solvers at scale

Fast diagonalization as a preconditioner
periodic cells, variable-coefficient surrogates for solvation

HASTE
these kernels as a standalone library for FE and spectral-element codes

All algorithms are implemented in DFT-FE, an open-source real-space FE code for DFT, and all speedups are measured against its default solvers and operators at the tolerances of its SCF runs.

Funding and compute: PMRF (Ministry of Education, Government of India), Google Research, NSM, OLCF, ALCF, RIKEN and SERC, IISc.