# Towards Exascale Fully Relativistic Pseudopotential Density Functional Theory Calculations

## with Mixed-Precision Computation and Compressed Communication Using Residual-Based Subspace Iteration

**Nikhil Kodali**[*], Gourab Panigrahi[*], Nishant Gupta[*], Sundaresan Gourishanker[*], Kartick Ramakrishnan[*], Rudra Panch[*], Sambit Das[†], Vishwas Rao[‡], Phani Motamarri[*]

[*]Department of Computational and Data Sciences, Indian Institute of Science, Bangalore, India
[†]Department of Mechanical Engineering, University of Michigan, Ann Arbor, USA
[‡]Argonne National Laboratory, Lemont, IL, USA

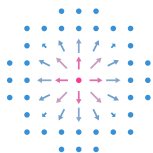
**SC26** · Chicago, IL · Technical Papers: Precision Engineering for Matrix Computation · 18 November 2026

# Why fully relativistic, noncollinear DFT?

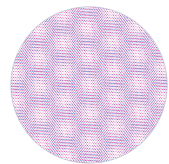
**Noncollinear (NC)**: magnetization direction varies in space
**Spin–orbit coupling (SOC)**: strong in heavy elements



Spin spirals



Skyrmions



Moiré lattices (nonlinear Hall effect)

Applications: spintronic memory and logic, topological qubits

Long-period textures (above) ⇒ **large supercells**; SOC ⇒ **fully relativistic** treatment

# Two challenges for large-scale NC-SOC-DFT

**Challenge 1:** computational cost of the fully relativistic eigenproblem

**Two-component spinors**
$M \times M \rightarrow 2M \times 2M$
($M$: grid points)

+

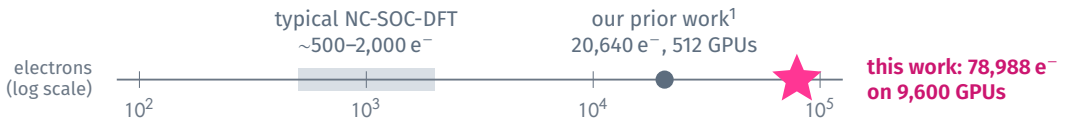
**Twice the eigenpairs**
$\geq N_{el}$, not $N_{el}/2$
($N_{el}$: electrons)

+

**Complex arithmetic**
cost ≈3–4× real

together →

**≈ 12–24× cost**
vs. collinear DFT (size-dependent); cubic-scaling cost dominates at smaller sizes




**Challenge 2:** GPUs fastest at **reduced precision**, but DFT needs **FP64 accuracy**

[1] N. Kodali and P. Motamarri, *Phys. Rev. B*, vol. 111, p. 195129, 2025

# Contributions

**FE** **Builds on the open-source DFT-FE code**[1]
adaptive higher-order finite elements (FE)

**i** **R-ChFSI eigensolver**[2]
residual-based Chebyshev filtered subspace iteration; tolerates inexact matrix products

**ii** **Mixed precision + compression**
TF32 or FP32 GEMMs; compressed ghost-exchange messages

**iii** **Band partitioning**
GPU groups own disjoint sets of bands (eigenvectors) ⇒ less communication

**iv** **Matrix-free GPU Poisson solver**
fused kernel for the Poisson operator

**4–6.4×**
smaller messages than FP64

**3.4×**
faster filtering than FP64 (best case)

**48 → 98%**
filtering efficiency, 384 nodes: without → with (iii)

**5.86×**
Poisson kernel throughput vs. deal.II

**up to 2×**
faster solution of the full nonlinear eigenproblem than FP64 (Aurora)

**78,988 e⁻**
on 9,600 GPUs, <7 min per self-consistent iteration

[1] S. Das, P. Motamarri, V. Subramanian, D. M. Rogers, and V. Gavini, "DFT-FE 1.0: A massively parallel hybrid CPU-GPU density functional theory code using finite-element discretization," *Comput. Phys. Commun.*, vol. 280, p. 108473, 2022
[2] N. Kodali *et al.*, *Comput. Phys. Commun.*, vol. 327, p. 110239, 2026

# Kohn–Sham DFT with two-component spinors

Spinors, charge density $\rho$, magnetization $\mathbf{m}$[1]

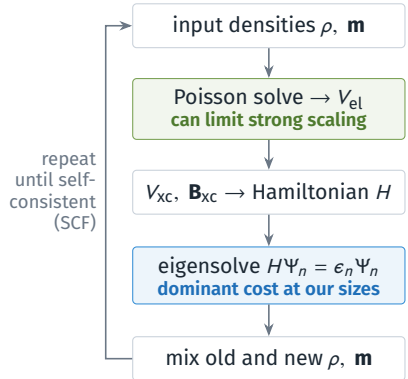
$$\Psi_n = \begin{pmatrix} \psi_n^{\uparrow} \\ \psi_n^{\downarrow} \end{pmatrix}, \quad \rho = \sum_n f_n \Psi_n^{\dagger} \Psi_n, \quad \mathbf{m} = \sum_n f_n \Psi_n^{\dagger} \vec{\sigma} \Psi_n$$

Nonlinear eigenproblem ($H$ depends on $\rho, \mathbf{m}$)

$$H\Psi_n = \epsilon_n \Psi_n, \quad H = \left(-\tfrac{1}{2}\nabla^2 + V_{\text{el}} + V_{\text{xc}} + V_{\text{loc}} - V_{\text{self}}\right)\mathbf{I} + \mathbf{B}_{\text{xc}} \cdot \vec{\sigma} + H_{\text{nloc}}$$

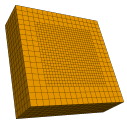
DFT electrostatics $\quad -\nabla^2 V_{\text{el}} = 4\pi\,(\rho + b)$

$N_e \geq N_{\text{el}}$ lowest states; $f_n$: occupations; $\vec{\sigma}$: Pauli matrices
$V_{\text{xc}}, \mathbf{B}_{\text{xc}}$: exchange–correlation potential and magnetic field
$V_{\text{loc}}, H_{\text{nloc}}$: norm-conserving (ONCV) pseudopotential[2]; $H_{\text{nloc}}$ carries SOC
$b$: smeared nuclear charges; $V_{\text{self}}$: their self-interaction

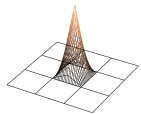



[1] U. von Barth and L. Hedin, *J. Phys. C: Solid State Phys.*, vol. 5, pp. 1629–1642, 1972; J. Kübler *et al.*, *J. Phys. F: Met. Phys.*, vol. 18, pp. 469–483, 1988
[2] D. R. Hamann, *Phys. Rev. B*, vol. 88, p. 085117, 2013; A. Dal Corso and A. M. Conte, *Phys. Rev. B*, vol. 71, p. 115106, 2005

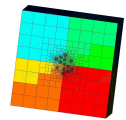
# Finite elements → sparse generalized eigenproblem



Adaptive mesh, refined near atoms



Compact-support basis functions



One subdomain per GPU

Spinors in a degree-$p = 7$ Lagrange basis on Gauss–Lobatto–Legendre nodes[1]

$$\Psi_n(\mathbf{r}) = \sum_{I=1}^{M} \begin{bmatrix} \psi_n^{I,\uparrow} \\ \psi_n^{I,\downarrow} \end{bmatrix} N_I(\mathbf{r})$$

- $\mathbf{HX} = \mathbf{MX}\Lambda$, $2M \times 2M$ ($M$: FE nodes); **H**: Hamiltonian, **M**: mass matrix
- **M** kept in full: a diagonal **M** would add error to the magnetization density[2]
- Compact support ⇒ **HX** via cell-wise **batched dense GEMMs**
- Electrostatics: $\mathbf{KV}_{\text{el}} = \mathbf{f}$ (Laplacian **K**), conjugate gradient (CG)

[1] P. Motamarri *et al.*, *J. Comput. Phys.*, vol. 253, pp. 308–343, 2013
[2] N. Kodali and P. Motamarri, *Phys. Rev. B*, vol. 111, p. 195129, 2025

*Largest run: $2M \approx 1.5 \times 10^8$ degrees of freedom (DoFs) and $N_e \geq 78{,}988$ eigenpairs in every SCF iteration.*

# Chebyshev filtered subspace iteration (ChFSI)

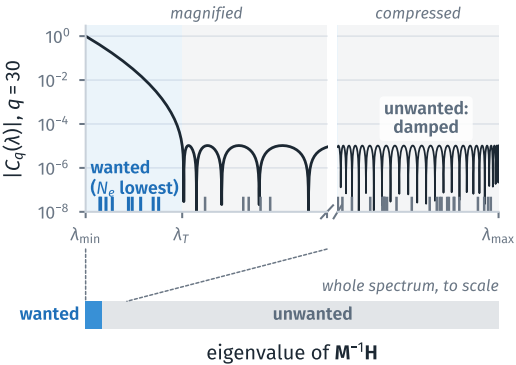



Illustration ($q = 30$; ticks: eigenvalues $\lambda$ of $\mathbf{M}^{-1}\mathbf{H}$). $|C_q|$ is small above the cut-off $\lambda_T$ and grows rapidly below it. Largest run: $N_e < 0.1\%$ of $2M$

**Chebyshev filtering (CF)**[1]

$\mathbf{Y} = C_q(\mathbf{M}^{-1}\mathbf{H})\mathbf{X}$ ($C_q$: degree-$q$ Chebyshev polynomial) via a three-term recurrence; each step: one $\mathbf{HX}$ product + one $\mathbf{M}^{-1}$ application

**Rayleigh–Ritz (RR)**

project onto $\mathbf{Y}$, solve the dense $N_e \times N_e$ eigenproblem (eigenvectors $\mathbf{Q}$), rotate $\mathbf{X} \leftarrow \mathbf{YQ}$; cost cubic in system size

Repeat until all residuals $\|\mathbf{Hx}_j - \epsilon_j \mathbf{Mx}_j\| < \tau$ (tolerance)

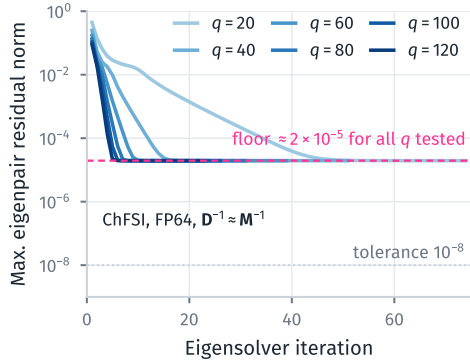
[1] Y. Zhou *et al.*, *J. Comput. Phys.*, vol. 219, pp. 172–184, 2006

# ChFSI can stagnate with inexact products

- Two sources of inexact products:
  - diagonal $\mathbf{D}^{-1} \approx \mathbf{M}^{-1}$, since exact $\mathbf{M}^{-1}$ needs a linear solve per filter step
  - rounding in reduced precision
- Error in the filtered block[1] is **independent of the residual**: $\|\hat{\mathbf{\Delta}}_q\| \leq \gamma_m \eta_q$ (rounding) $+ \zeta \tilde{\eta}_q$ ($\mathbf{D}^{-1} \neq \mathbf{M}^{-1}$)
- $\theta_i$ (angle to wanted eigenspace, iteration $i$) decreases if

  $$|C_q(\lambda_{N_e})| - |C_q(\lambda_{N_e+1})| > \|\hat{\mathbf{\Delta}}_q\| \, (\sec\theta_i + \csc\theta_i)$$

  $\csc\theta_i \to \infty$ as $\theta_i \to 0$, so this eventually fails $\Rightarrow$ residual **floor** (right)




[1] N. Kodali *et al.*, *Comput. Phys. Commun.*, vol. 327, p. 110239, 2026

*$\hat{\mathbf{\Delta}}_q = \mathbf{M}^{1/2}\mathbf{\Delta}_q$ (computed − exact); $\gamma_m$: floating-point error constant; $\zeta = \|\mathbf{D}^{-1} - \mathbf{M}^{-1}\|$; $\eta_q, \tilde{\eta}_q$: $q$-dependent constants. Plot: benchmark of ref. 1 (carbon supercell with a vacancy, 85.8M DoFs, 13,500 eigenpairs, Aurora).*

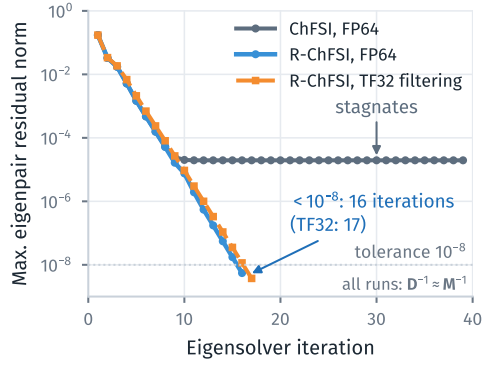
# R-ChFSI: residual-based ChFSI tolerates inexactness

Chebyshev recurrence in weighted residuals[1]
$\mathbf{Z}_k = \mathbf{D}(\mathbf{Y}_k - \mathbf{X}\Lambda_k)$; residual $\mathbf{R} = \mathbf{HX} - \mathbf{MX}\Lambda$:

$$\mathbf{Z}_{k+1} = a_k \mathbf{H}\mathbf{D}^{-1}\mathbf{Z}_k + b_k \mathbf{Z}_k + c_k \mathbf{Z}_{k-1} + a_k \mathbf{R}\Lambda_k,$$
$$\mathbf{Y}_q = \mathbf{D}^{-1}\mathbf{Z}_q + \mathbf{X}\Lambda_q$$

- Filter-error bound $\|\hat{\mathbf{\Delta}}_q\| \le (\gamma_m \eta_q + \zeta \tilde{\eta}_q)\,\|\hat{\mathbf{R}}\| \to 0$ **as the eigenpairs converge**

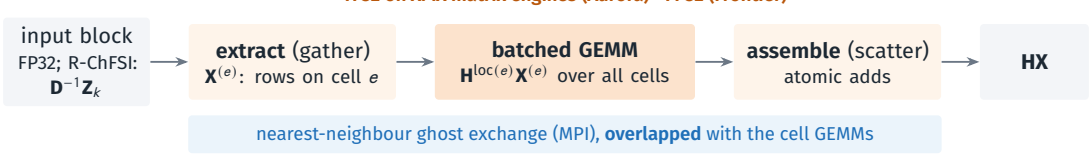
**Convergence guarantee.** If
$|C_q(\lambda_{N_e})| - |C_q(\lambda_{N_e+1})| > 2\|\hat{\mathbf{H}}\|(\gamma_m \eta_q + \zeta \tilde{\eta}_q)(1 + \tan\theta_i)$
holds at some iteration $i_0$, it holds at all later ones: $\theta_i$ keeps decreasing.




[1] N. Kodali *et al.*, *Comput. Phys. Commun.*, vol. 327, p. 110239, 2026

$\mathbf{Y}_k$: *block after k filter steps;* $\Lambda_k = C_k(\Lambda)$; $a_k, b_k, c_k$: *recurrence coefficients (backup B1);* *practical form using* $\mathbf{DM}^{-1} \approx \mathbf{I}$; $\hat{\mathbf{R}} = \mathbf{M}^{-1/2}\mathbf{R}$, $\hat{\mathbf{H}} = \mathbf{M}^{-1/2}\mathbf{H}\mathbf{M}^{-1/2}$. *Same benchmark as the previous slide,* $q = 60$.

# Reduced-precision HX: FE-cell-level batched GEMMs




Local cell Hamiltonian[1], $2n_p^3 \times 2n_p^3$

$$\mathbf{H}^{\mathrm{loc}(e)} = \begin{bmatrix} \mathbf{H}_s + \mathbf{H}_{B_z} & \mathbf{H}_{B_x} - i\,\mathbf{H}_{B_y} \\ \mathbf{H}_{B_x} + i\,\mathbf{H}_{B_y} & \mathbf{H}_s - \mathbf{H}_{B_z} \end{bmatrix}$$

$n_p = p + 1$; $\mathbf{H}_s$: kinetic + potentials; $\mathbf{H}_{B_d}$: component $d$ of $\mathbf{B}_{\mathrm{xc}}$. Built in FP64, stored in FP32. Nonlocal (SOC) part: extra batched GEMMs

**Precision in the R-ChFSI filter and rotation**

| | |
|---|---|
| **FP64** | residual **R**, $\Lambda_k$ recurrence, rotation by the diagonal part of **Q** |
| **TF32 or FP32** | cell GEMMs, extraction/assembly; rotation by the off-diagonal part of **Q** (FP32) |
| **BFP** | ghost-exchange messages in filtering (next slide) |

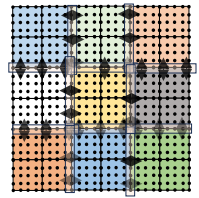
Faster GEMMs ⇒ larger **communication** share

[1] N. Kodali and P. Motamarri, *Phys. Rev. B*, vol. 111, p. 195129, 2025; G. Panigrahi *et al.*, *J. Parallel Distrib. Comput.*, vol. 192, p. 104925, 2024

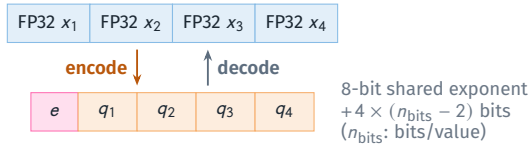
# Block floating-point compression of ghost exchanges

- Every **HX**: 2 **ghost exchanges** (update, accumulation); message size $\propto$ bits/value
- **BFP**[1]: 4 values share one 8-bit exponent $\Rightarrow$ BFP16 has $\approx 2\times$ the significand bits of BF16[2] at 16 bits/value
- R-ChFSI **tolerates the rounding error**

| Format | bits/value | coefficient bits | smaller than FP64 | smaller than FP32 |
|---|---|---|---|---|
| BFP16 | 16 | 14 | 4× | 2× |
| BFP12 | 12 | 10 | 5.3× | 2.7× |
| BFP10 | 10 | 8 | 6.4× | 3.2× |



Dots: FE nodes; colours: MPI ranks; boxed strips: boundary values exchanged with neighbours




[1] M. Drumond *et al.*, *Adv. Neural Inf. Process. Syst.*, vol. 31, pp. 451–461, 2018; B. D. Rouhani *et al.*, *arXiv:2310.10537*, 2023
[2] N. Kodali *et al.*, *Comput. Phys. Commun.*, vol. 327, p. 110239, 2026; K. Ramakrishnan and P. Motamarri, *arXiv:2604.26037*, 2026

*Smaller than FP64 (FP32) = 64 (32) / bits per value. Figure from the SC26 paper.*

# Testbeds and benchmark systems

## Aurora (ALCF)[1]

10,624 nodes in total



6 Intel Max 1550 GPUs × 2 tiles = **12 GPUs/node**
8× Slingshot-11, 1.6 Tb/s · GEMMs in **TF32** (XMX, oneMKL)

## Frontier (OLCF)

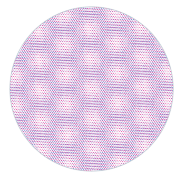
9,856 nodes in total



4 AMD MI250X × 2 GCDs = **8 GPUs/node**
4× Slingshot-11, 800 Gb/s · GEMMs in **FP32** (hipBLAS)

each filled square = one GPU in this talk: an Aurora tile or a Frontier GCD (graphics compute die)

| Bilayer | Twist | Atoms | Valence e⁻ | DoFs/atom |
|---|---|---|---|---|
| $Fe_3GeTe_2$ | 7.3° | 732 | 11,468 | ~45k |
| $MoSe_2$ | 2.1° | 4,326 | 37,492 | ~16k |
| $MoSe_2$ | 1.5° | 9,114 | **78,988** | ~16k |



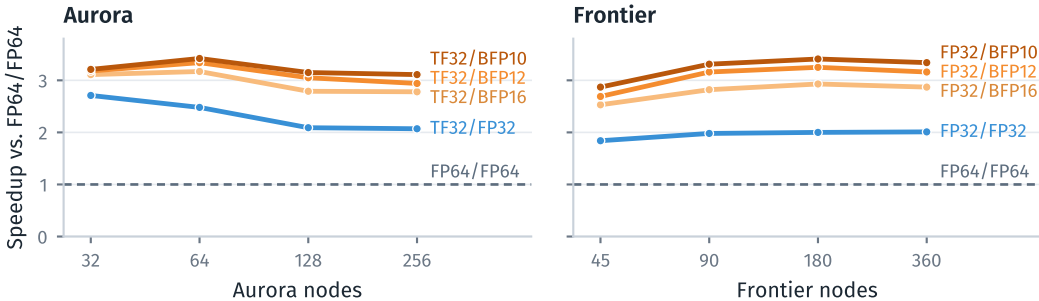
moiré (schematic)

- $Fe_3GeTe_2$: van der Waals ferromagnet; $MoSe_2$: semiconductor
- PBE, SG15 ONCV pseudopotentials[2]
- FE degree $p = 7$; meshes for an energy discretization error of $10^{-4}$ Ha/atom (Ha: hartree)

[1] W. E. Allcock *et al.*, *arXiv:2509.08207*, 2025
[2] J. P. Perdew *et al.*, *Phys. Rev. Lett.*, vol. 77, pp. 3865–3868, 1996; M. Schlipf and F. Gygi, *Comput. Phys. Commun.*, vol. 196, pp. 36–44, 2015

RESULTS

# Mixed precision + BFP: 2.5–3.4× faster filtering




**FP32 messages**
Aurora: 2.7× → 2.1× from 32 to 256 nodes (communication grows)

**BFP messages**
Aurora: 2.8–3.4×
Frontier: 2.5–3.4×

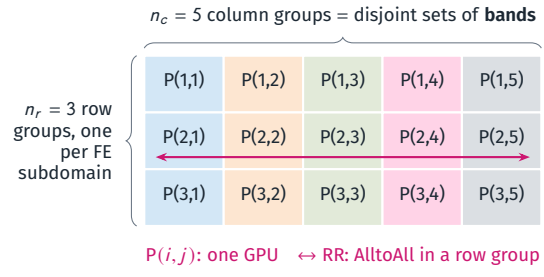
**Efficiency at 8× nodes**
BFP vs. FP32 messages
Aurora: 60–65% vs. 51%
Frontier: 59–61% vs. 57%

*Labels: compute/message precision. 732-atom $Fe_3GeTe_2$; speedups vs. FP64/FP64 at the same node count; strong-scaling efficiency at 256 vs. 32 (Aurora) and 360 vs. 45 (Frontier) nodes. BFP12: default for the band-partitioning and large-scale runs.*

# Band partitioning reduces filtering communication

Strong scaling ⇒ fewer DoFs per GPU:

- **Problem:** filtering becomes communication-bound
- **Opportunity:** GPU memory is freed ⇒ **spend it to cut communication**




**Filtering**

each column group: own bands + own copy of the sparse **H**, **M**, split over its $n_r$ GPUs ⇒ communication only inside the group, **none between groups**

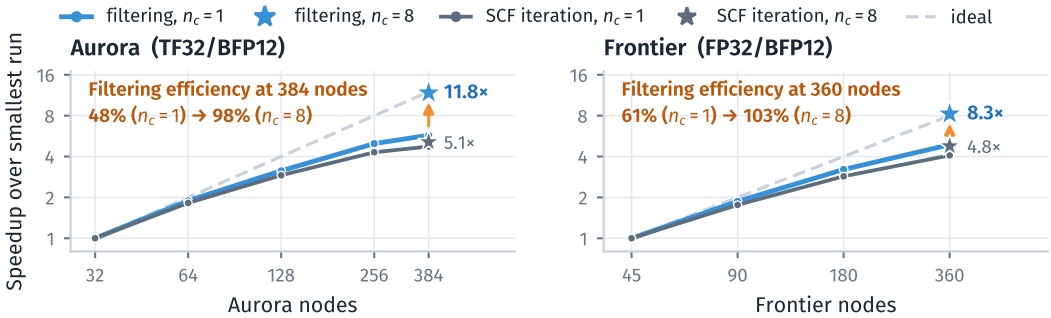
**Rayleigh–Ritz (RR) needs all bands**

AlltoAll within each row group to the $n_c = 1$ layout, standard RR, convert back

**Cost**

extra memory for copies of **H**, **M** and layout buffers; AlltoAll latency grows with $n_c$ (backup B4)

# Band partitioning: 98% / 103% filtering efficiency




**Filtering with $n_c = 8$**

Aurora: 11.8× at 12× the nodes (384 vs. 32): 98%
Frontier: 8.3× at 8× the nodes (360 vs. 45): 103%

**One SCF iteration, efficiency $n_c = 1 \rightarrow 8$**

39 → 43% (Aurora), 51 → 60% (Frontier) ⇒ the non-filtering steps appear to limit scaling

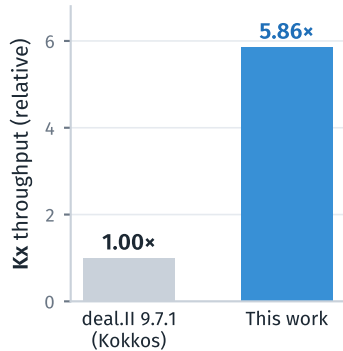
*732-atom $Fe_3GeTe_2$; baselines TF32/BFP12 (Aurora) and FP32/BFP12 (Frontier) with $n_c = 1$; 2nd SCF iteration onward.*

# Matrix-free Poisson kernel: 5.86× deal.II's throughput

**Kx**, the main cost of each CG iteration, runs as one fused GPU kernel; sum factorization: $O(p^{2d}) \rightarrow O(dp^{d+1})$ work per cell ($d = 3$)

| | |
|---|---|
| **1** gather nodal values + $z$-contraction | registers |
| **2** $y$-, $x$-contractions (even/odd symmetry) | shared memory |
| **3** gradients + Jacobian $\mathbf{J}^{-\top}\mathbf{J}^{-1}$ | registers |
| **4** transposed contractions + atomic assembly | reg. → global |

- Our earlier kernel[1] had a similar speedup; this one: higher occupancy, supports larger $p$




baseline: deal.II matrix-free[2]; both FP64

[1] G. Panigrahi *et al.*, *J. Parallel Distrib. Comput.*, vol. 192, p. 104925, 2024

[2] D. Arndt *et al.*, *J. Numer. Math.*, vol. 33, pp. 403–415, 2025; M. Kronbichler and K. Kormann, *Comput. Fluids*, vol. 63, pp. 135–147, 2012

*732-atom $Fe_3GeTe_2$, 32 Aurora nodes (384 GPUs), $p = 7$; same FP64 preconditioned CG solver (degree-5 Chebyshev–Jacobi, tolerance $10^{-10}$); deal.II 9.7.1, Kokkos 4.7.01.*

# Full SCF solve, all iterations: 1.8–2.0× faster than FP64

732-atom $Fe_3GeTe_2$, 256 Aurora nodes; each SCF iteration = eigensolve + electrostatics + mixing

| Compute/Comm. | Filter passes | SCF iterations | Wall time (s) | Speedup | $\Delta E$ (Ha/atom) |
|---|---|---|---|---|---|
| FP64/FP64 | 41 | 38 | 2873 | 1.0× | — |
| TF32/BFP16 | 44 | 39 | 1540 | 1.9× | $1.9 \times 10^{-8}$ |
| TF32/BFP12 | 44 | 38 | 1420 | 2.0× | $3.4 \times 10^{-9}$ |
| TF32/BFP10 | 49 | 39 | 1565 | 1.8× | $7.3 \times 10^{-9}$ |

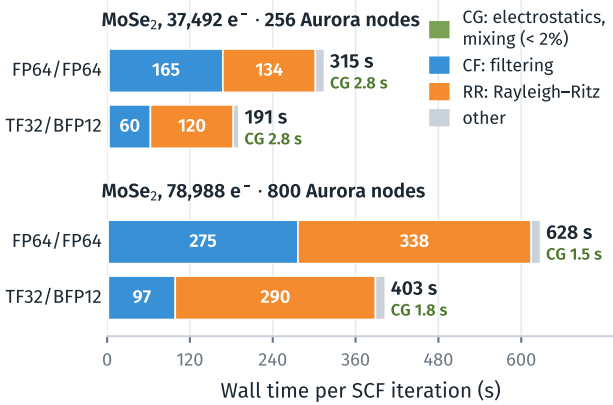
**Convergence**
38–39 SCF iterations (FP64: 38); filter passes 41 → 44–49

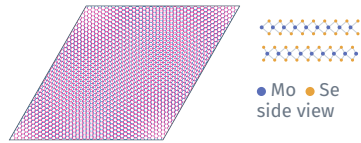
**Energy**
$\Delta E \leq 1.9 \times 10^{-8}$ Ha/atom $\ll 10^{-4}$ Ha/atom discretization error

**BFP width**
best width may vary with the problem (here BFP12)

*Filter pass: one filtering + RR step, summed over the SCF solve. SCF tolerance $5 \times 10^{-4}$ on $\rho$ and $\mathbf{m}$; wall time excludes initialization; $\Delta E$ relative to FP64/FP64.*

# Largest run: 78,988 electrons, 9,600 GPUs




9,114 atoms, 1.5° twist (top view)

## 403 s

per SCF iteration, 800 Aurora nodes

| TF32/BFP12 vs. FP64/FP64 | 37k e⁻ | 79k e⁻ |
|---|---|---|
| filtering | **2.75×** | **2.85×** |
| SCF iteration | **1.65×** | **1.56×** |

At 79k e⁻: RR 290 of 403 s, limited by communication in ELPA[1] and distributed GEMMs

[1] A. Marek *et al.*, *J. Phys.: Condens. Matter*, vol. 26, p. 213201, 2014

*Twisted bilayer $MoSe_2$ with SOC; representative 2nd SCF iteration (one filtering pass); Chebyshev degree $q = 60$; CG tolerance $10^{-10}$.*

# Takeaways

**1.8–2.0×**
faster full SCF solve vs. FP64 (732 atoms, Aurora);
$\Delta E < 2 \times 10^{-8}$ Ha/atom

**3.4×**
faster filtering vs. FP64 (best case)

**98% / 103%**
filtering efficiency with band partitioning, at 12× / 8× nodes (Aurora / Frontier)

**78,988 $e^-$**
on 9,600 GPUs, < 7 min per SCF iteration

## Implemented in DFT-FE, applicable beyond it

- All methods are in the open-source **DFT-FE** code
- R-ChFSI: largely **basis-independent** ⇒ transferable to real-space and plane-wave codes
- Band partitioning: adaptable to plane-wave codes (e.g. Quantum ESPRESSO)
- Gains depend on scale: for ~200–1,000 electrons, FP64 ChFSI and simpler parallelization may suffice

# Thank you

## Questions?

Nikhil Kodali nikhilkodali@iisc.ac.in
Phani Motamarri (corresponding author) phanim@iisc.ac.in

**DFT-FE is open source**

github.com/dftfeDevelopers/dftfe

**Acknowledgments.** Aurora (ALCF) and Frontier (OLCF), U.S. DOE Office of Science user facilities, Contracts DE-AC02-06CH11357 and DE-AC05-00OR22725. N.K., G.P., K.R.: Prime Minister's Research Fellowship (PMRF), Ministry of Education, India. P.M.: Google India Research Award 2023; ANRF ARG MATRICS grant ANRF/ARGM/2025/002824/TS; National Quantum Mission, DST, Govt. of India. V.R.: U.S. DOE Office of Science, ASCR, Contract DE-AC02-06CH11357.

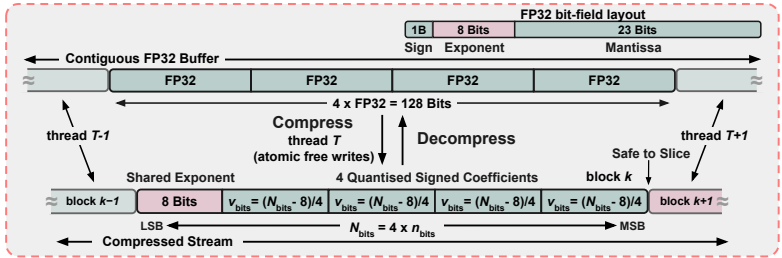
# R-ChFSI eigensolver

**Require:** $\mathbf{H} \in \mathbb{C}^{2M\times 2M}$, $\mathbf{M} \in \mathbb{R}^{2M\times 2M}$, Ritz pairs $(\mathbf{X}^{(0)}, \mathbf{\Lambda}^{(0)})$, degree $q$, bounds $\lambda_{\min}, \lambda_{\max}, \lambda_T$, diagonal $\mathbf{D}^{-1} \approx \mathbf{M}^{-1}$, tolerance $\tau$

$i \leftarrow 0$
**while** $\max_j \|\mathbf{H}\mathbf{x}_j^{(i)} - \epsilon_j^{(i)}\mathbf{M}\mathbf{x}_j^{(i)}\| \geq \tau$ **do**
  $\mathbf{R}^{(i)} = \mathbf{H}\mathbf{X}^{(i)} - \mathbf{M}\mathbf{X}^{(i)}\mathbf{\Lambda}^{(i)}$ — FP64
  $\mathbf{Z}_0 = 0$, $\mathbf{Z}_1 = \sigma_1\mathbf{R}^{(i)}/e$; run the R-ChFSI recurrence up to $\mathbf{Z}_q$ — TF32 or FP32 products, BFP messages
  $\mathbf{Y}_q = \mathbf{D}^{-1}\mathbf{Z}_q + \mathbf{X}^{(i)}\mathbf{\Lambda}_q$
  Rayleigh–Ritz: solve $\mathbf{Y}_q^\dagger\mathbf{H}\mathbf{Y}_q\,\mathbf{Q} = \mathbf{Y}_q^\dagger\mathbf{M}\mathbf{Y}_q\,\mathbf{Q}\,\mathbf{\Lambda}^{(i+1)}$ (ELPA)
  Subspace rotation $\mathbf{X}^{(i+1)} \leftarrow \mathbf{Y}_q\mathbf{Q}$ — diagonal part of $\mathbf{Q}$ in FP64, off-diagonal part in FP32
  Update $\lambda_{\min}, \lambda_T$ from $\mathbf{\Lambda}^{(i+1)}$; $i \leftarrow i+1$
**end while**

Recurrence: $\mathbf{Z}_{k+1} = a_k\mathbf{H}\mathbf{D}^{-1}\mathbf{Z}_k + b_k\mathbf{Z}_k + c_k\mathbf{Z}_{k-1} + a_k\mathbf{R}\mathbf{\Lambda}_k$ with $a_k = 2\sigma_{k+1}/e$, $b_k = -2\sigma_{k+1}c/e$, $c_k = -\sigma_k\sigma_{k+1}$; $c = (\lambda_{\max}+\lambda_T)/2$, $e = (\lambda_{\max}-\lambda_T)/2$, $\sigma_1 = e/(\lambda_{\min}-c)$, $\sigma_{k+1} = (2/\sigma_1 - \sigma_k)^{-1}$; $\mathbf{\Lambda}_k = C_k(\mathbf{\Lambda}^{(i)})$ follows the same three-term recurrence in FP64. The Rayleigh–Ritz projection and rotation are blockwise, pipelining communication with GEMMs.

# BFP format and encoding

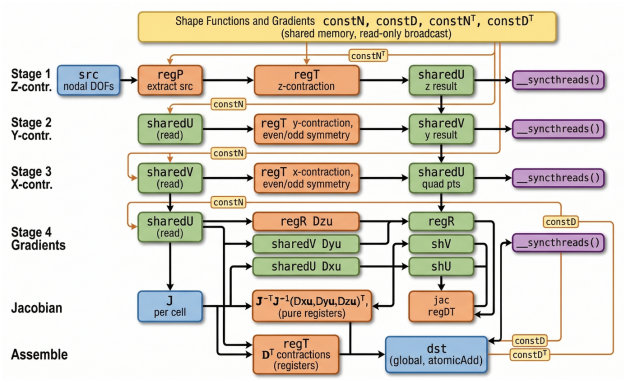



- Shared exponent: $2^{e_{\max}-1} \le \max_i |x_i| < 2^{e_{\max}}$, stored biased ($e_{\max} + 127$); an all-zero block is flagged by a zero exponent field
- Quantization: $q_i = \mathrm{round}\left(x_i \, 2^{v_{\text{bits}}-1-e_{\max}}\right)$, $v_{\text{bits}} = n_{\text{bits}} - 2$; decoding $\tilde{x}_i = q_i \, 2^{e_{\max}-v_{\text{bits}}+1}$ reconstructs the quantized value exactly
- Wavefunction block size $N_b$ a multiple of $4 \Rightarrow$ messages of exactly $n_\Gamma N_b n_{\text{bits}}/8$ bytes ($n_\Gamma$: ghost nodes per message; $\times 2$ for complex); BFP16/12/10 stored as 1 `uint64` / 3 `uint16` / 5 `uint8`

# Fused PCG iteration and the Kx kernel

1: **Kx kernel: $\mathbf{w} \leftarrow \mathbf{K}\mathbf{p}$**
2: **dot:** $\eta \leftarrow \mathbf{p} \cdot \mathbf{w}$
3: `Allreduce`$(\eta)$; $\alpha \leftarrow \delta/\eta$
4: **fused update:** $\mathbf{x} \mathrel{+}= \alpha\mathbf{p}$, $\mathbf{r} \mathrel{+}= \alpha\mathbf{w}$
5: **Chebyshev precond.:** $\mathbf{z} \leftarrow \mathbf{P}^{-1}\mathbf{r}$
6: **fused dot:** $(\delta_{\text{new}}, \gamma) \leftarrow (\mathbf{r} \cdot \mathbf{z}, \|\mathbf{r}\|^2)$
7: `Allreduce`$([\delta_{\text{new}}, \gamma])$
8: $\beta \leftarrow \delta_{\text{new}}/\delta$; $\delta \leftarrow \delta_{\text{new}}$
9: $\mathbf{p} \leftarrow -\mathbf{z} + \beta\mathbf{p}$
10: **if** $\sqrt{\gamma} < \varepsilon$ **then** converged

$\mathbf{P}^{-1} = \chi(\mathbf{K}_{\text{diag}}^{-1}\mathbf{K})\,\mathbf{K}_{\text{diag}}^{-1}$, spectral bounds from a short Lanczos run. Dot products: registers, warp shuffles, shared-memory tree, block sum, two atomics per block.




Hybrid register/shared-memory strategy: intermediates stay in registers unless needed across threads.

# Rayleigh–Ritz with band partitioning

**Require:** filtered $\mathbf{Y}_q$ in the 2D block layout ($n_r \times n_c$)
1: $\mathbf{Z_M} \leftarrow \mathbf{M}\mathbf{Y}_q$, $\mathbf{Z_H} \leftarrow \mathbf{H}\mathbf{Y}_q$
2: **for all** $\mathbf{A} \in \{\mathbf{Y}_q, \mathbf{Z_M}, \mathbf{Z_H}\}$ **do**
3: `AlltoAll`($\mathbf{A}$) in row group
4: `ConvertLayout`($\mathbf{A}$)
5: **end for**
6: $\mathbf{X}, \Lambda \leftarrow$ standard RR($\mathbf{Y}_q, \mathbf{Z_M}, \mathbf{Z_H}$)
7: `ConvertLayout`($\mathbf{X}$); `AlltoAll`($\mathbf{X}$) in row group
8: **return** $\mathbf{X}, \Lambda$

- Filtering uses $n_c > 1$ (no inter-group traffic); the RR substeps run with $n_c = 1$ and communicate as without band partitioning, plus the layout conversions
- 4 `AlltoAll`s per RR step, each $\approx \alpha(n_c - 1) + \beta s_{\mathbb{C}} M N_e / n_g$
- Extra memory per GPU: $8 s_{\mathbb{C}} M N_e / n_g + (n_c - 1)\mathcal{M}_{\mathbf{H},\mathbf{M}}/n_g$ (out-of-place layout buffers and replicas)
- The best $n_c$ balances filtering savings against layout conversions; $n_c = 8$ at the largest node counts

*$M$: FE nodes; $n_g$: GPUs in total; $s_{\mathbb{C}}$: bytes per complex scalar; $\mathcal{M}_{\mathbf{H},\mathbf{M}}$: memory of $\mathbf{H}$ and $\mathbf{M}$ without replication; AlltoAll cost per layout conversion ($\alpha$–$\beta$ model).*

# Chebyshev filtering strong-scaling data

| | **Aurora** (X = 32 nodes) | | | | **Frontier** (X = 45 nodes) | | | |
|---|---|---|---|---|---|---|---|---|
| **Compute/Comm.** | X | 2X | 4X | 8X | X | 2X | 4X | 8X |
| entries: speedup (strong-scaling efficiency %) | | | | | | | | |
| FP64/FP64 | 1.00 | 1.00 (90) | 1.00 (82) | 1.00 (67) | 1.00 | 1.00 (79) | 1.00 (66) | 1.00 (52) |
| TF32 or FP32/FP32 | 2.71 | 2.48 (82) | 2.09 (63) | 2.07 (51) | 1.84 | 1.98 (85) | 2.00 (72) | 2.01 (57) |
| . . . /BFP16 | 3.11 | 3.17 (92) | 2.79 (73) | 2.78 (60) | 2.53 | 2.82 (88) | 2.93 (77) | 2.87 (59) |
| . . . /BFP12 | 3.18 | 3.34 (95) | 3.05 (79) | 2.94 (62) | 2.69 | 3.16 (93) | 3.25 (80) | 3.16 (61) |
| . . . /BFP10 | 3.21 | 3.42 (96) | 3.15 (80) | 3.11 (65) | 2.87 | 3.31 (91) | 3.41 (79) | 3.34 (61) |

Speedup over FP64/FP64 at the same node count; strong-scaling efficiency (%) relative to X nodes of the same configuration in parentheses. Compute is TF32 on Aurora and FP32 on Frontier. 732-atom $Fe_3GeTe_2$, 2nd SCF iteration onward.