Lecture Tue.3

Intro to the EPW code and recent developments

Hyungjun Lee

Oden Institute for Computational Engineering and Sciences
The University of Texas at Austin

NSF
U.S. DEPARTMENT OF ENERGY
TACC

Lecture Summary

- Brief introduction to EPW
- Computational flow
- Recent developments
- Future plans

Brief introduction to EPW [https://epw-code.org]

- Open-source Fortran code to calculate materials properties related to the electron-phonon interaction
  - Based on density-functional perturbation theory and maximally localized Wannier functions.
- One of core modules of Quantum ESPRESSO since 2016

Functionalities of EPW

- Interpolate electron-phonon matrix element on fine momentum grids
- Compute electron and phonon linewidths, scattering rates & lifetimes
- Electron and phonon spectral functions
- Electron-phonon coupling strength
- Superconducting properties
- Transport properties

Features of EPW

- Support LDA/GGA functionals
- MPI parallelization:
  - over **k** points and bands on coarse grids
  - over **k** points on fine grids
- Support spin-orbit coupling
- Support time-reversal symmetry
- Interpolate polar divergence correctly
- Integrated into QE and rely on Wannier90
- Have a test-farm for stability and portability of the code

Computational flow

epw.f90

```
  CALL epw_readin()
..
  IF (wannierize) CALL wann_run()
  CALL elphon_shuffle_wrap()
  >> CALL elphel2_shuffle
  > CALL ephwann_shuffle
..
```

➤ Read input keywords,
read wfns and misc. data from pw.x

Computational flow

epw.f90

```
  CALL epw_readin()
..
  IF (wannierize) CALL wann_run()
  CALL elphon_shuffle_wrap()
  >> CALL elphel2_shuffle
  > CALL ephwann_shuffle
..
```

➤ Generate inputs to W90 and Wannierization

Computational flow

epw.f90

```
  CALL epw_readin()
..
  IF (wannierize) CALL wann_run()
  CALL elphon_shuffle_wrap()
  >> CALL elphel2_shuffle
  > CALL ephwann_shuffle
..
```

➤ Calculated *e*-ph vertex on coarse grids

Computational flow

epw.f90

```
  CALL epw_readin()
..
  IF (wannierize) CALL wann_run()
  CALL elphon_shuffle_wrap()
  >> CALL elphel2_shuffle
  > CALL ephwann_shuffle
..
```

➤ Bloch to Wannier and Wannier to Bloch, + …

Computational flow:

Wannierization

- Call W90 as a library (run serially)
- Generate internally inputs (amn, mmn, etc.) to W90
- Generate directly real-space Wannier orbitals [.cube]
- Main input keywords of W90 ➤ inputs of EPW [Tue.4 Hands-on]
- Others can also be passed to W90 [Tue.4 Hands-on]

Computational flow:

e-ph evaluation

$$\boxed{g(\mathbf{k}_{\mathrm{c}},\mathbf{q}_{\mathrm{c}})} \blacktriangleright \boxed{g(\mathbf{R}_e,\mathbf{R}_p)} \blacktriangleright \boxed{g(\mathbf{k}_{\mathrm{f}},\mathbf{q}_{\mathrm{f}})} \blacktriangleright \boxed{\sum\nolimits''_{n\mathbf{k}_{\mathrm{f}}}[g(\mathbf{k}_{\mathrm{f}},\mathbf{q}_{\mathrm{f}})], \Pi''_{\nu\mathbf{q}_{\mathrm{f}}}[g(\mathbf{k}_{\mathrm{f}},\mathbf{q}_{\mathrm{f}})], ...}$$

Computational flow:

e-ph evaluation

$$\boxed{\boxed{g(\mathbf{k}_\mathrm{c},\mathbf{q}_\mathrm{c})}} \rightarrow \boxed{g(\mathbf{R}_e,\mathbf{R}_p)} \rightarrow \boxed{g(\mathbf{k}_\mathrm{f},\mathbf{q}_\mathrm{f})} \rightarrow \boxed{\sum\nolimits''_{n\mathbf{k}_\mathrm{f}}[g(\mathbf{k}_\mathrm{f},\mathbf{q}_\mathrm{f})],\ \Pi''_{\nu\mathbf{q}_\mathrm{f}}[g(\mathbf{k}_\mathrm{f},\mathbf{q}_\mathrm{f})],\ldots}$$

Wave functions from pw.x (NSCF)

$$g_{mn\nu}(\mathbf{k}_\mathrm{c},\mathbf{q}_\mathrm{c}) = \langle u_{m\mathbf{k}_\mathrm{c}+\mathbf{q}_\mathrm{c}} | \Delta_{\mathbf{q}_\mathrm{c}\nu} v^{\mathrm{KS}} | u_{n\mathbf{k}_\mathrm{c}} \rangle_{\mathrm{uc}}$$

Perturbing potentials
(In patterns basis)
from ph.x

$$\Delta_{\mathbf{q}_\mathrm{c}\nu} v^{\mathrm{KS}} = \sum_{\kappa\alpha} \sqrt{\frac{\hbar}{2M_\kappa \omega_{\mathbf{q}_\mathrm{c}\nu}}}\, e_{\kappa\alpha,\nu}(\mathbf{q}_\mathrm{c})\, \partial_{\kappa\alpha,\mathbf{q}_\mathrm{c}} v^{\mathrm{KS}}$$

Caculated from
dynamical matrices from ph.x

, where $\partial_{\kappa\alpha,\mathbf{q}_\mathrm{c}} v^{\mathrm{KS}} = e^{-i\mathbf{q}_\mathrm{c}\cdot\mathbf{r}} \sum_p e^{i\mathbf{q}_\mathrm{c}\cdot\mathbf{R}_p} \frac{\partial V^{\mathbf{KS}}(\mathbf{r})}{\partial \tau_{\kappa\alpha p}}$

[Phys. Rev. B **76**, 165108 (2007), Rev. Mod. Phys. **89**, 1 (2017)]

Technicalities:

Difference from ph.x

ph.x explicitly calculates the states at **k** and **k**+**q** .

EPW folds the states at **k**+**q** by using the periodic gauge condition

$$u_{n,\mathbf{k}+\mathbf{G}}(\mathbf{r}) = e^{-i\mathbf{G}\cdot\mathbf{r}}u_{n\mathbf{k}}(\mathbf{r})$$

Computational flow:

e-ph evaluation

$$\boxed{g(\mathbf{k}_{\mathrm{c}},\mathbf{q}_{\mathrm{c}})} \rightarrow \boxed{\boxed{g(\mathbf{R}_e,\mathbf{R}_p)}} \rightarrow \boxed{g(\mathbf{k}_{\mathrm{f}},\mathbf{q}_{\mathrm{f}})} \rightarrow \boxed{\textstyle\sum''_{n\mathbf{k}_{\mathrm{f}}}[g(\mathbf{k}_{\mathrm{f}},\mathbf{q}_{\mathrm{f}})], \Pi''_{\nu\mathbf{q}_{\mathrm{f}}}[g(\mathbf{k}_{\mathrm{f}},\mathbf{q}_{\mathrm{f}})], \ldots}$$

$$g_{mn\kappa\alpha}(\mathbf{R}_p,\mathbf{R}_{p'}) = \langle \mathrm{w}_{m0}(\mathbf{R})|\frac{\partial V^{\mathrm{KS}}}{\partial \tau_{\kappa\alpha}}(\mathbf{r}-\mathbf{R}_{p'})|\mathrm{w}_{m0}(\mathbf{R}-\mathbf{R}_p)\rangle_{\mathrm{sc}}$$

[Phys. Rev. B **76**, 165108 (2007), Rev. Mod. Phys. **89**, 1 (2017)]

Use of minimal-distance replica selection for Wannier interpolation

[J. Phys. Cond. Matt. 32, 165902 (2020), arXiv:2105.04192 (2021)]

Computational flow:

e-ph evaluation

$$\boxed{g(\mathbf{k}_\mathrm{c},\mathbf{q}_\mathrm{c})} \rightarrow \boxed{g(\mathbf{R}_e,\mathbf{R}_p)} \rightarrow \boxed{\boxed{g(\mathbf{k}_\mathrm{f},\mathbf{q}_\mathrm{f})}} \rightarrow \boxed{\sum\nolimits''_{n\mathbf{k}_\mathrm{f}}[g(\mathbf{k}_\mathrm{f},\mathbf{q}_\mathrm{f})], \Pi''_{\nu\mathbf{q}_\mathrm{f}}[g(\mathbf{k}_\mathrm{f},\mathbf{q}_\mathrm{f})], \ldots}$$

$$g_{mn\nu}(\mathbf{k}_\mathrm{f},\mathbf{q}_\mathrm{f}) = \sqrt{\frac{\hbar}{2M_\kappa\omega_{\mathbf{q}_\mathrm{f}\nu}}}\sum_{pp'} e^{i(\mathbf{k}_\mathrm{f}\cdot\mathbf{R}_p+\mathbf{q}_\mathrm{f}\cdot\mathbf{R}_{p'})}$$
$$\times\,[U_{\mathbf{k}_\mathrm{f}+\mathbf{q}_\mathrm{f}}\, g(\mathbf{R}_p,\mathbf{R}_{p'})\cdot e_{\kappa\alpha,\nu}(\mathbf{q}_\mathrm{f})\, U^\dagger_{\mathbf{k}_\mathrm{f}}]_{mn}$$

[Phys. Rev. B **76**, 165108 (2007), Rev. Mod. Phys. **89**, 1 (2017)]

Use of crystal symmetry (w.r.t **k**) and
Fermi window

Computational flow:

e-ph evaluation

$$\boxed{g(\mathbf{k}_\mathrm{c},\mathbf{q}_\mathrm{c})} \rightarrow \boxed{g(\mathbf{R}_e,\mathbf{R}_p)} \rightarrow \boxed{g(\mathbf{k}_\mathrm{f},\mathbf{q}_\mathrm{f})} \rightarrow \boxed{\textstyle\sum''_{n\mathbf{k}_\mathrm{f}}[g(\mathbf{k}_\mathrm{f},\mathbf{q}_\mathrm{f})],\ \Pi''_{\nu\mathbf{q}_\mathrm{f}}[g(\mathbf{k}_\mathrm{f},\mathbf{q}_\mathrm{f})],\ \ldots}$$

$$\begin{aligned}\Sigma''_{n\mathbf{k}_\mathrm{f}}(\omega,T) =& \pi\sum_{m\nu}\int_\mathrm{BZ}\frac{d\mathbf{q}_\mathrm{f}}{\Omega_\mathrm{BZ}}|g_{mn\nu}(\mathbf{k}_\mathrm{f},\mathbf{q}_\mathrm{f})|^2 \\ &\{[n_{\mathbf{q}_\mathrm{f}\nu}(T)+f_{m\mathbf{k}_\mathrm{f}+\mathbf{q}_\mathrm{f}}(T)]\delta(\omega-(\varepsilon_{m\mathbf{k}_\mathrm{f}+\mathbf{q}_\mathrm{f}}-\varepsilon_\mathrm{F})+\omega_{\mathbf{q}_\mathrm{f}\nu}) \\ &+[n_{\mathbf{q}_\mathrm{f}\nu}(T)+1-f_{m\mathbf{k}_\mathrm{f}+\mathbf{q}_\mathrm{f}}(T)]\delta(\omega-(\varepsilon_{m\mathbf{k}_\mathrm{f}+\mathbf{q}_\mathrm{f}}-\varepsilon_\mathrm{F})-\omega_{\mathbf{q}_\mathrm{f}\nu})\}\end{aligned}$$

[Comput. Phys. Commun. **209**, 116 (2016), Rev. Mod. Phys. **89**, 1 (2017)]

Technicalities:

e-ph evaluation on coarse grids

- Use of the periodic gauge condition for folding $u_{n,\mathbf{k}+\mathbf{G}}(\mathbf{r}) = e^{-i\mathbf{G}\cdot\mathbf{r}} u_{n\mathbf{k}}(\mathbf{r})$

- Use of crystal symmetry

$$\begin{aligned} g_{mn\nu}(\mathbf{k}, S\mathbf{q}) &= \langle \psi_{m,\mathbf{k}+S\mathbf{q}}(\mathbf{r}) | \partial_{\nu,S\mathbf{q}} V(\mathbf{r}) | \psi_{n,\mathbf{k}}(\mathbf{r}) \rangle \\ &= \langle \psi_{m,\mathbf{k}+S\mathbf{q}}(\mathbf{r}) | \partial_{\nu\mathbf{q}} V(\{S|\mathbf{v}\}^{-1}\mathbf{r}) | \psi_{n,\mathbf{k}}(\mathbf{r}) \rangle \\ &= \langle \psi_{m,\mathbf{k}+S\mathbf{q}}(\{S|\mathbf{v}\}\mathbf{r}) | \partial_{\nu\mathbf{q}} V(\mathbf{r}) | \psi_{n,\mathbf{k}}(\{S|\mathbf{v}\}\mathbf{r}) \rangle \end{aligned}$$

- Use of time-reversal symmetry

(only when the small group of **q** doesn’t include **q** <> -**q** operation)

$$g_{mn\nu}(\mathbf{k}, -S\mathbf{q}) = \langle \psi_{m,\mathbf{k}-S\mathbf{q}}(\{S|\mathbf{v}\}\mathbf{r}) | \left(\partial_{\nu\mathbf{q}} V(\mathbf{r}) \right)^* | \psi_{n,\mathbf{k}}(\{S|\mathbf{v}\}\mathbf{r}) \rangle$$

[Phys. Rev. B **76**, 165108 (2007), Comput. Phys. Commun. **209**, 116 (2016)]

General design principles

- Improve the computational workflow
- Reduce the computational size
- Optimize the computation

Improvement on workflow

- Divide EPW into two separate codes
- Incorporate NSCF step into EPW
- Start from irreducible coarse grids.
- Other simplifications

Efficient calculation of *e*-ph vertex

- e-ph vertex is a main ingredient to calculate electron-phonon related quantities

$$g_{mn\nu}(\mathbf{k},\mathbf{q}) = \langle \psi_{m,\mathbf{k}+\mathbf{q}}(\mathbf{r}) | \partial_{\nu,\mathbf{q}} V(\mathbf{r}) | \psi_{n,\mathbf{k}}(\mathbf{r}) \rangle$$

- A large portion of time spent on its evaluation in EPW
- Calculated in plane-waves basis on coarse grids and in reduced basis on dense grids after Wannier interpolation
- Gauges of wave functions are arbitrary, but consistent ones should be used throughout all calculations
- On coarse grids, calculated for all pairs of states at ***k*** and ***k*+*q*** $N_b^2 \times N_k \times N_q \times N_\nu$(evaluations)

(number of bands N_b, $\mathbf{k}$ vector N_k, $\mathbf{q}$ vector N_q, and modes N_ν)

Efficient calculation of e-ph vertex:

Coarse grids: reduction of computational size

- Exploit the crystal symmetry to reduce the number of evaluations for e-ph vertex

$$\begin{aligned} g_{mn\nu}(\mathbf{k}, S\mathbf{q}) &= \langle \psi_{m,\mathbf{k}+S\mathbf{q}}(\mathbf{r}) | \partial_{\nu,S\mathbf{q}} V(\mathbf{r}) | \psi_{n,\mathbf{k}}(\mathbf{r}) \rangle \\ &= \langle \psi_{m,\mathbf{k}+S\mathbf{q}}(\mathbf{r}) | \partial_{\nu\mathbf{q}} V(\{S|\mathbf{v}\}^{-1}\mathbf{r}) | \psi_{n,\mathbf{k}}(\mathbf{r}) \rangle \\ &= \langle \psi_{m,\mathbf{k}+S\mathbf{q}}(\{S|\mathbf{v}\}\mathbf{r}) | \partial_{\nu\mathbf{q}} V(\mathbf{r}) | \psi_{n,\mathbf{k}}(\{S|\mathbf{v}\}\mathbf{r}) \rangle \end{aligned}$$

[Phys. Rev. B **76**, 165108 (2007)]

- If the following relations hold throughout all wave functions,

$$\begin{aligned} \psi_{m,S^{-1}\mathbf{k}+\mathbf{q}}(\mathbf{r}) &= \psi_{m,\mathbf{k}+S\mathbf{q}}(\{S|\mathbf{v}\}\mathbf{r}) \\ \psi_{n,S^{-1}\mathbf{k}}(\mathbf{r}) &= \psi_{n,\mathbf{k}}(\{S|\mathbf{v}\}\mathbf{r}) \end{aligned}$$

- Can we use the symmetry relation between e-ph vertices?

$$g_{mn\nu}(\mathbf{k}, S\mathbf{q}) = g_{mn\nu}(S^{-1}\mathbf{k}, \mathbf{q})$$

Efficient calculation of e-ph vertex:

Coarse grids: reduction of computational size

- We can’t use the equation below to reduce the number of evaluations:

$$g_{mn\nu}(\mathbf{k}, S\mathbf{q}) = g_{mn\nu}(S^{-1}\mathbf{k}, \mathbf{q})$$

  - Ex) For the inversion operation in the little group of the wavevector ***k***, and its odd-parity state *n*,

$$\psi_{n\mathbf{k}}(\{S|\mathbf{v}\}\mathbf{r}) = \{S|\mathbf{v}\}^{-1}\psi_{n\mathbf{k}}(\mathbf{r}) = -\psi_{n\mathbf{k}}(\mathbf{r}) \neq \psi_{n,S^{-1}\mathbf{k}}(\mathbf{r})$$

$[S^{-1} = S \text{ for inversion operation}]$

- We need to multiply the additional factors (the same for states at ***k***+***q***):

$$\boxed{g_{mn\nu}(\mathbf{k}, S\mathbf{q}) = \sum_{m'n'} \langle \psi_{m',S^{-1}\mathbf{k}+\mathbf{q}}(\mathbf{r}) | \psi_{m,\mathbf{k}+S\mathbf{q}}(\{S|\mathbf{v}\}\mathbf{r}) \rangle^{*} g_{m'n'\nu}(S^{-1}\mathbf{k}, \mathbf{q}) \langle \psi_{n',S^{-1}\mathbf{k}}(\mathbf{r}) | \psi_{n\mathbf{k}}(\{S|\mathbf{v}\}\mathbf{r}) \rangle}$$

Efficient calculation of *e*-ph vertex:

Coarse grids: reduction of computational size

- We can reduce the number of evaluations:
  - For each $\boldsymbol{q}_{\mathrm{irr}}$, explicitly evaluate the e-ph vertex
  - For the star of $\boldsymbol{q}_{\mathrm{irr}}$, skip the explicit evaluation by using the symmetry relation of e-ph vertex
- Speedup by a factor of $N_{\boldsymbol{q}}/N_{\boldsymbol{q}\mathrm{irr}}$ (-α)
  - For GaN with 6x6x6 ***k*** and ***q*** points ($N_{\boldsymbol{q}}/N_{\boldsymbol{q}\mathrm{irr}}$=7.7) , speedup by 4

Efficient calculation of *e*-ph vertex:

Fine grids: optimize the computation

- Symmetry broken during Wannierization

$$g_{mn\nu}(\mathbf{k},\mathbf{q}) = \sum_{pp'} e^{i(\mathbf{k}\cdot\mathbf{R}_p + \mathbf{q}\cdot\mathbf{R}_{p'})} \sum_{m'n'\kappa\alpha} U_{mm'\mathbf{k}+\mathbf{q}} g_{m'n'\kappa\alpha}(\mathbf{R}_p, \mathbf{R}_{p'}) U^{\dagger}_{n'n\mathbf{k}} u_{\kappa\alpha,\mathbf{q}\nu}\,.$$

, where Wigner-Seitz vector $\mathbf{R}_{p(p')}$, rotation matrix U, eigenmode u, band index $m(m'), n(n')$, mode index ν, and Cartesian directions α

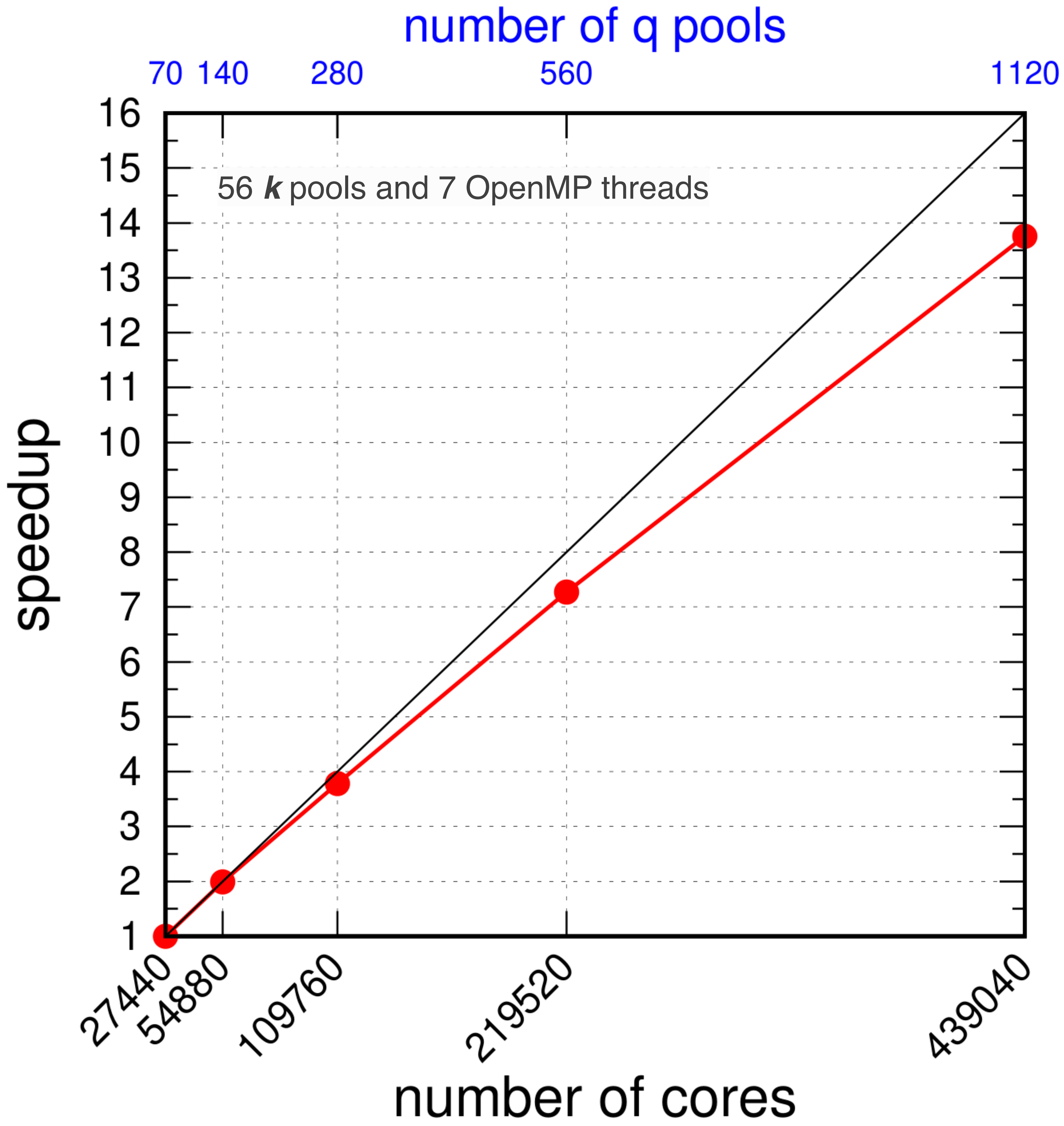
[Phys. Rev. B **76**, 165108 (2007), Rev. Mod. Phys. **89**, 1 (2017)]

- Instead, hybrid parallelization scheme enables highly-scalable evaluations
  - ***q*** point parallelization using MPI
  - ***k*** point parallelization using MPI and OpenMP

Efficient calculation of e-ph vertex:

Fine grids: optimize the computation

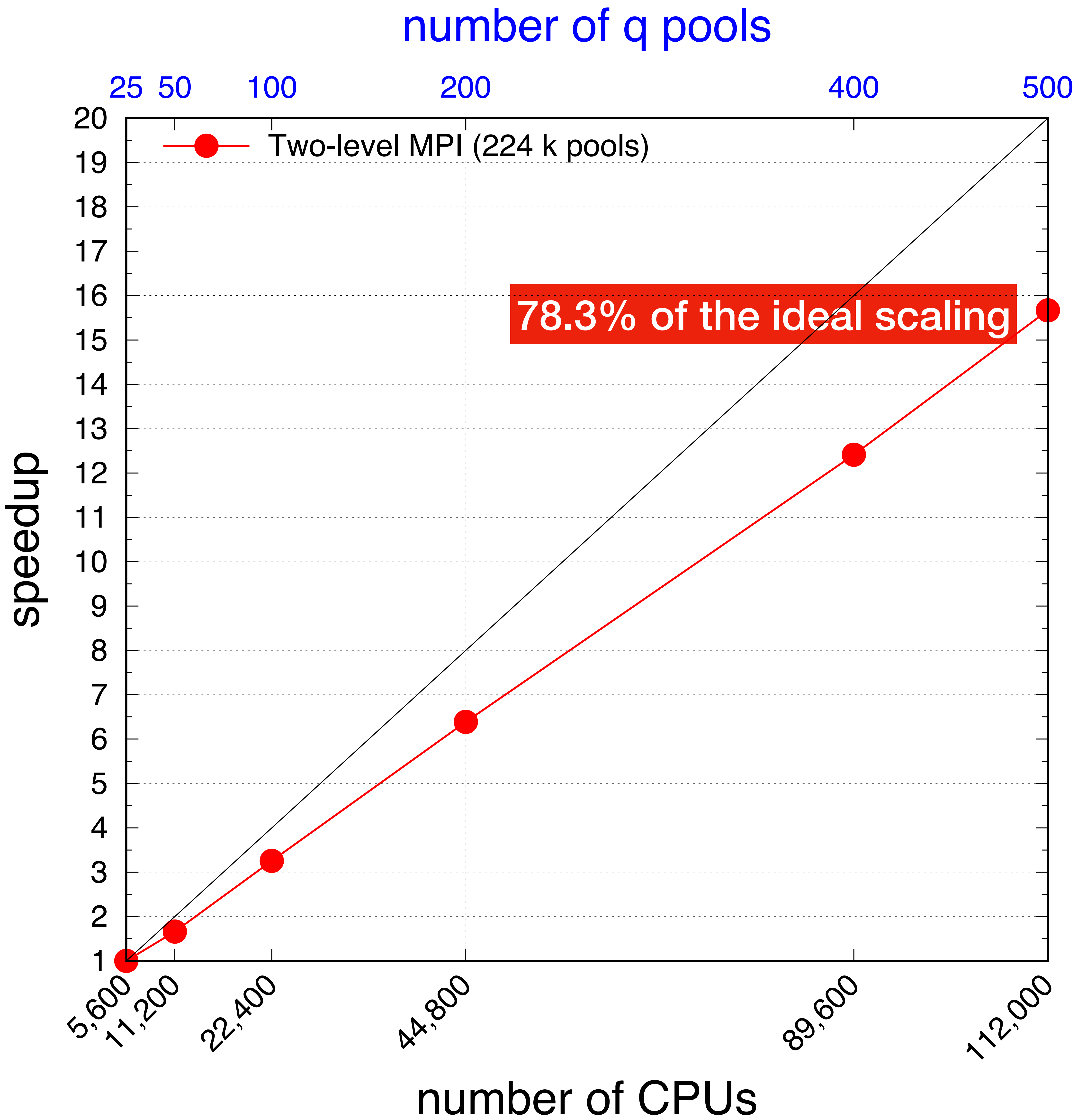
- Hybrid two-level MPI and OpenMP
- 86% speedup of the ideal one on half-million cores
  [done on Frontera during Texascale Days in Dec. 2020]
- Time-to-solution of 21 min. for the interpolation of e-ph vertex on 10^6 full ***k*** and ***q*** points with 3 bands for MgB_2 ($8.1\text{x}10^{13}$ elements)



Extension of Hybrid Parallelization Sheme:

Superconducting module

- Done on Frontera up to 112,000 cores
- MgB_2
  - 36x36x36 *k* and *q* fine grids (2,413 irreducible **k** points and 46,656 full **q** points)
  - No Fermi window (five bands considered)
  - Maximum number (40) of iterations set for benchmark purpose
- Use of low I/O mode



Other additions

- Symmetric *e*-ph vertex in USPP and PAW
- OpenMP parallelization for *G* operations
- Parallel I/O using parallel HDF5
    - Reduce the time for I/O
        - Average time of 15.3 sec for file I/O of 3.3 TB using 5,040 cores [done on Summit with GPFS file system]
    - Reduce the number of files to minimum
    - Increased portability and flexibility
- Optimized linear tetrahedron method [Phys. Rev. B **89**, 094515 (2014)]

Symmetric *e*-ph vertex in USPP and PAW

- Only a few correction terms needed to add to ultrasoft pseudopotential (USPP) case [Phys. Rev. B **81**, 075123 (2010)]

$$\boxed{\sum_{\sigma}\sum_{I,mn}\Delta D^{1,\sigma,\mu}_{I,mn} b^{\sigma,\lambda}_{I,mn}}$$

$$\text{where } \Delta D^{1,\sigma,\mu}_{I,mn} = \left(\frac{dD^{1,\sigma}_{I,mn}}{d\mu} - \frac{d\tilde{D}^{1,\sigma}_{I,mn}}{d\mu}\right),\ \frac{dD^{1,\sigma}_{I,mn}}{d\mu} = \sum_{\sigma_1}\int_{\Omega_I} d^3r\,\Phi^{I,\mathrm{AE}}_m(\mathbf{r})\Phi^{I,\mathrm{AE}}_n(\mathbf{r})\frac{dV^{I,\sigma}_{\mathrm{eff}}}{d\rho^{1,I}_{\sigma_1}}\frac{d\rho^{1,I}_{\sigma_1}}{d\mu}$$

$$,\ \text{and } b^{\sigma,\mu}_{I,mn} = \sum_i \langle\tilde{\psi}_{i,\sigma}|\frac{\partial(|\beta^I_m\rangle\langle\beta^I_n|)}{\partial\mu}|\tilde{\psi}_{i,\sigma}\rangle - \sum_i \langle\tilde{\psi}_{i,\sigma}|\beta^I_m\rangle\langle\beta^I_n|\delta^\mu\tilde{\psi}_{i,\sigma}\rangle$$

- Calculation of terms common in both PAW and USPP parallelized over ***G***

Symmetric *e*-ph vertex in USPP and PAW

- Use of symmetric expression of e-ph vertex
  [Phys. Rev. B **100**, 174304 (2019), Phys. Rev. B **101**, 184302 (2020)]

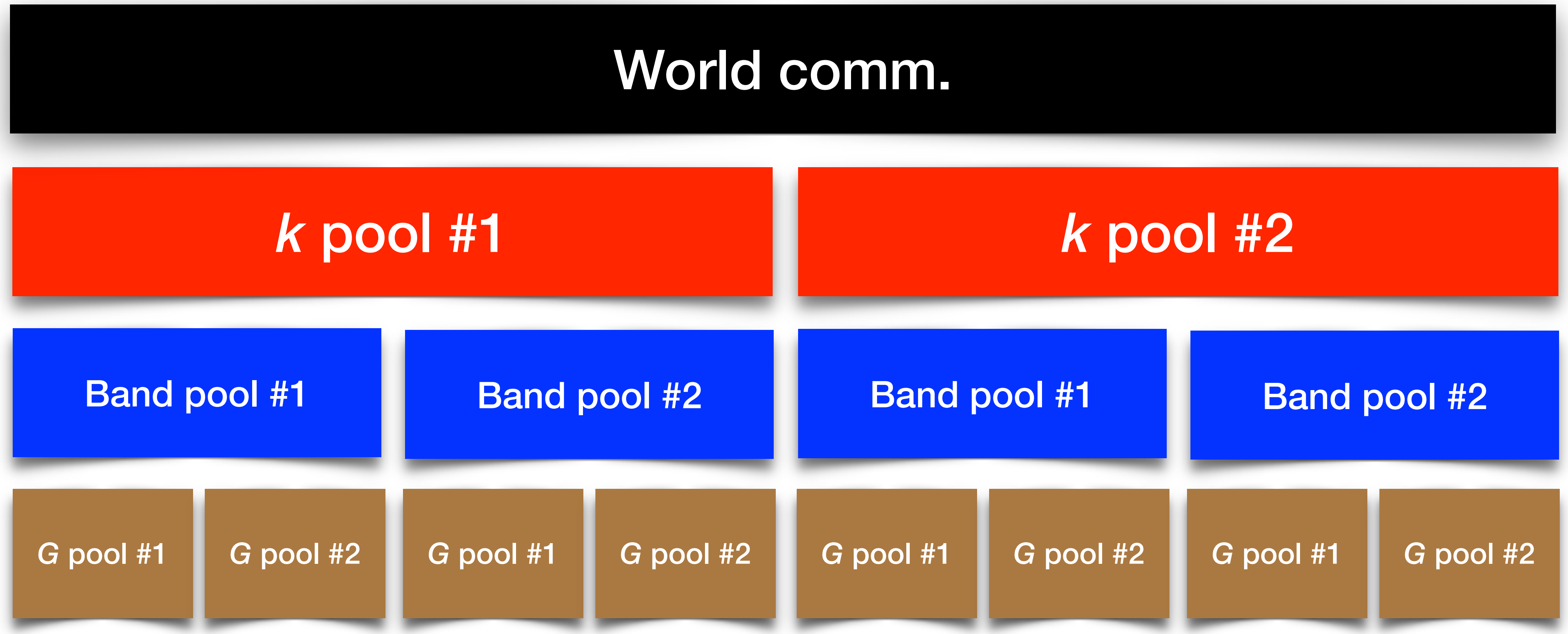
$$g_{mn\nu}(\mathbf{k},\mathbf{q}) = \underbrace{\langle \tilde{\Psi}_{m,\mathbf{k}+\mathbf{q}} | \partial_{\nu\mathbf{q}} \tilde{H} - \epsilon_{n\mathbf{k}} \partial_{\nu\mathbf{q}} \tilde{S} | \tilde{\Psi}_{n\mathbf{k}} \rangle}_{g^{\mathrm{QE}}_{mn\nu}(\mathbf{k},\mathbf{q})} + (\epsilon_{n\mathbf{k}} - \epsilon_{m,\mathbf{k}+\mathbf{q}}) \langle \tilde{\Psi}_{m,\mathbf{k}+\mathbf{q}} | \hat{T}^{\dagger} \partial_{\nu\mathbf{q}} \hat{T} | \tilde{\Psi}_{n\mathbf{k}} \rangle$$

$$\Rightarrow \boxed{|g_{mn\nu}(\mathbf{k},\mathbf{q})| = |g^{*}_{nm\nu}(\mathbf{k}+\mathbf{q},-\mathbf{q})|}$$

- For use with USPP as well, ignore the negligible term with derivatives of on-site partial waves

New Parallel Structure

$$\boxed{g(\mathbf{k}_{\mathrm{c}},\mathbf{q}_{\mathrm{c}})} \rightarrow \boxed{g(\mathbf{R}_e,\mathbf{R}_p)} \quad \rightarrow \quad \boxed{g(\mathbf{k}_{\mathrm{f}},\mathbf{q}_{\mathrm{f}})} \rightarrow \boxed{\begin{array}{l}\sum''_{n\mathbf{k}_{\mathrm{f}}}[g(\mathbf{k}_{\mathrm{f}},\mathbf{q}_{\mathrm{f}})],\\ \Pi''_{\nu\mathbf{q}_{\mathrm{f}}}[g(\mathbf{k}_{\mathrm{f}},\mathbf{q}_{\mathrm{f}})],\ldots\end{array}}$$

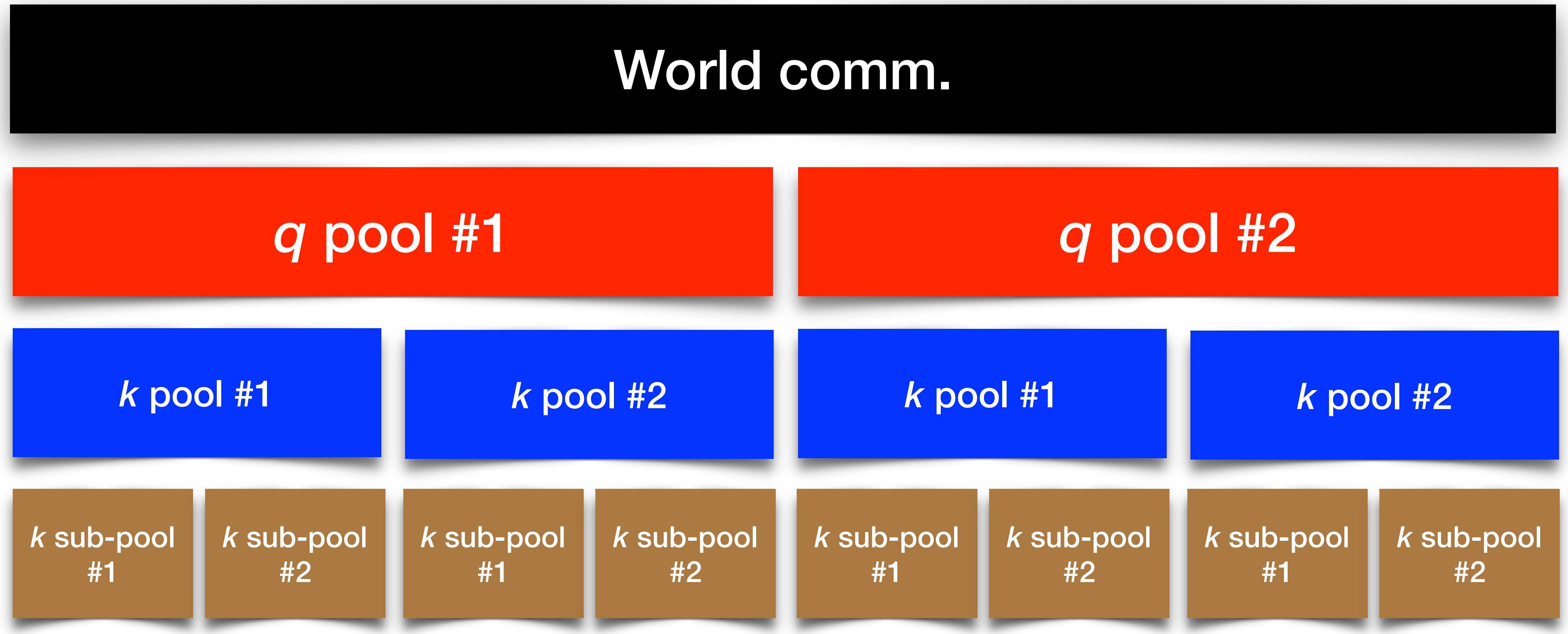


World comm.

k pool #1 | *k* pool #2

Band pool #1 | Band pool #2 | Band pool #1 | Band pool #2

G pool #1 | *G* pool #2 | *G* pool #1 | *G* pool #2 | *G* pool #1 | *G* pool #2 | *G* pool #1 | *G* pool #2



World comm.

q pool #1 | *q* pool #2

k pool #1 | *k* pool #2 | *k* pool #1 | *k* pool #2

k sub-pool #1 | *k* sub-pool #2 | *k* sub-pool #1 | *k* sub-pool #2 | *k* sub-pool #1 | *k* sub-pool #2 | *k* sub-pool #1 | *k* sub-pool #2

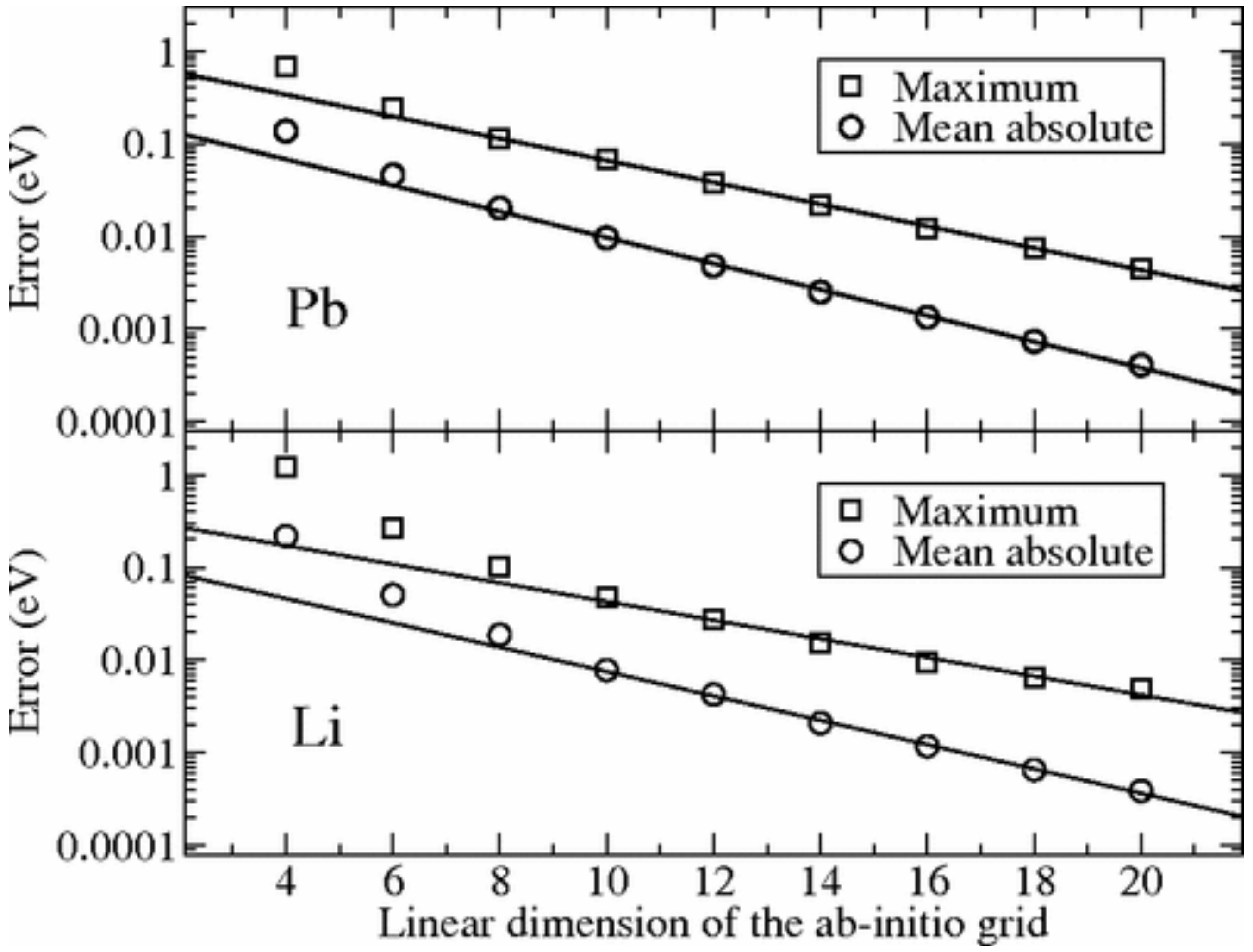
Short-term plans

- Easy construction of (symmetric) localized basis
- Non-collinear magnetism

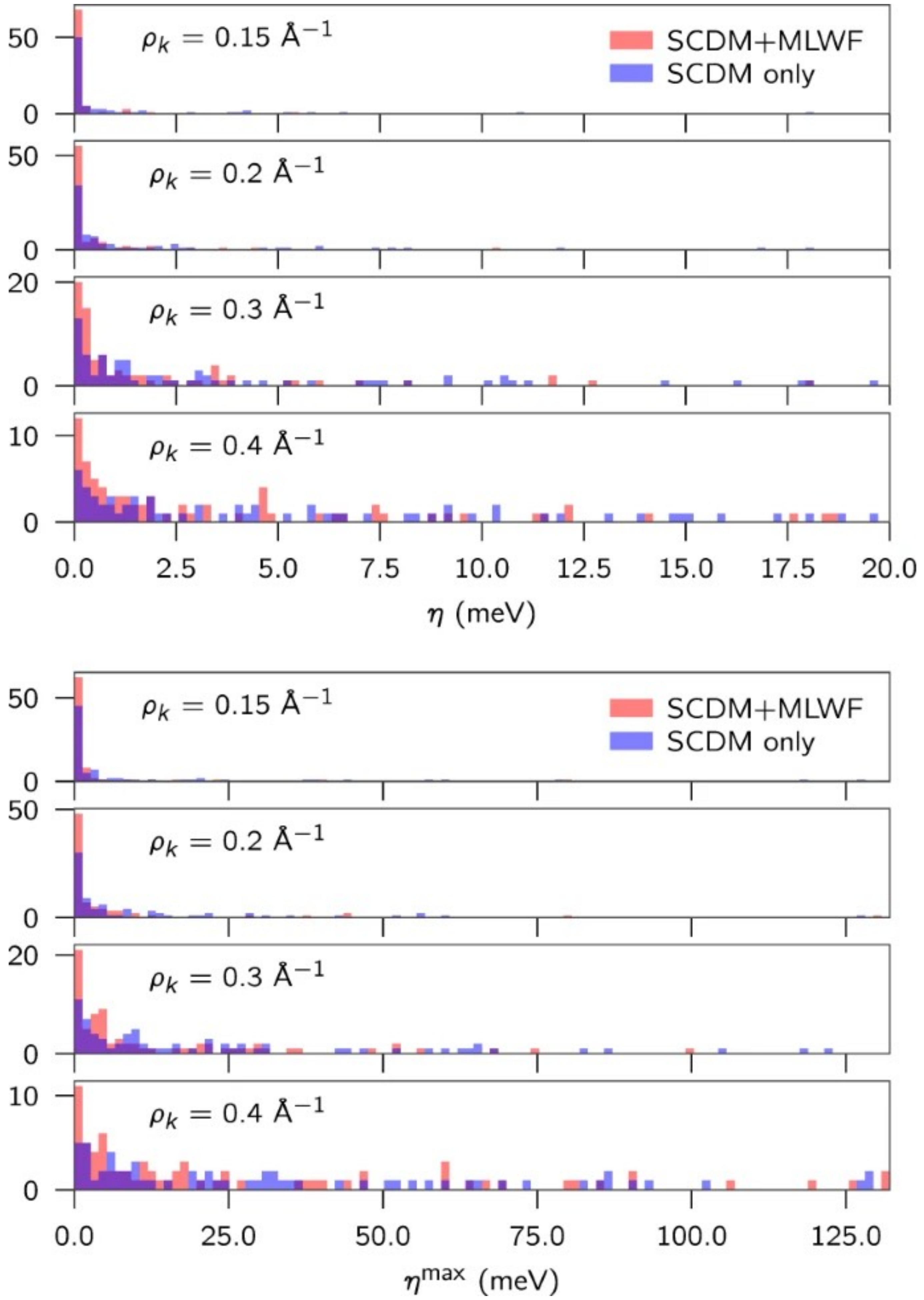
Short-term plans

Easy construction of (symmetric) localized basis

- Importance of high grid density



[Phys. Rev. B 75, 195121 (2007)]



[npj Comput Mater 6, 66 (2020)]

Short-term plans

Easy construction of (symmetric) localized basis

- Projectability [Phys. Rev. B **88**, 165127 (2013)]
- Optimal choice of SCDM-k parameters [npj Comput Mater **6**, 66 (2020)]
- Energy window optimization [Phys. Rev. Materials **2**, 103805 (2018)]

Mid-term plans

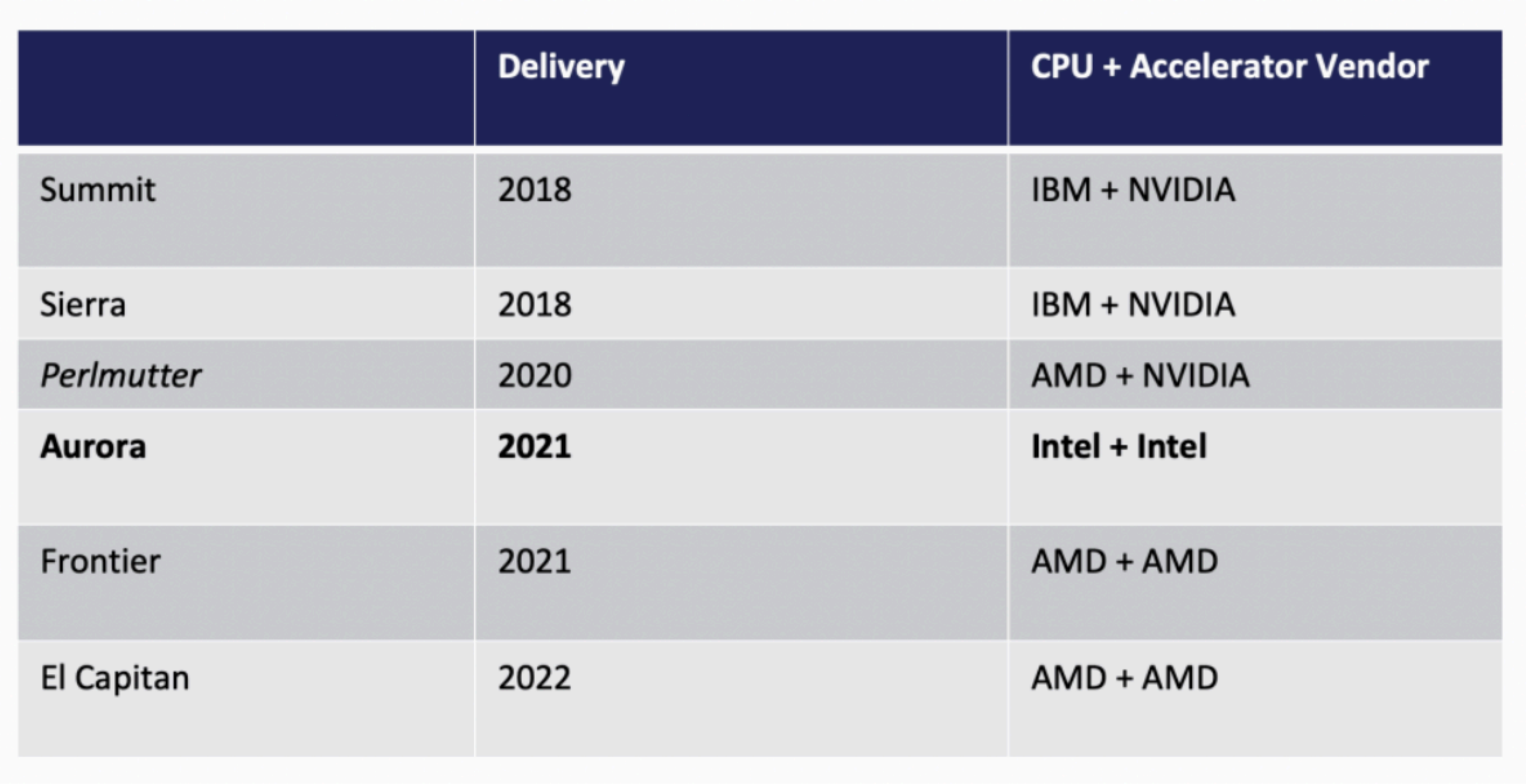
- Port EPW to the heterogeneous architecture (GPU)
- Interface EPW to other DFT codes

Mid-term plans

Porting EPW to GPU

- Importance of OpenMP
- Choice of programming models

An Overview of Some DOE Systems



	Delivery	CPU + Accelerator Vendor
Summit	2018	IBM + NVIDIA
Sierra	2018	IBM + NVIDIA
Perlmutter	2020	AMD + NVIDIA
Aurora	**2021**	**Intel + Intel**
Frontier	2021	AMD + AMD
El Capitan	2022	AMD + AMD

https://alcf.anl.gov/support-center/aurora/porting-aurora

References

- F. Giustino et al., Phys. Rev. B 76, 165108 (2007) [link]
- C. Verdi et al., Phys. Rev. Lett. 115, 176401 (2015) [link]
- S. Ponc´e et al., Comput. Phys. Commun. 209, 116 (2016) [link]
- F. Giustino, Rev. Mod. Phys. 89, 1 (2017) [link]

More Info

- https://epw-code.org
- https://forum.epw-code.org
- https://gitlab.com/QEF/q-e