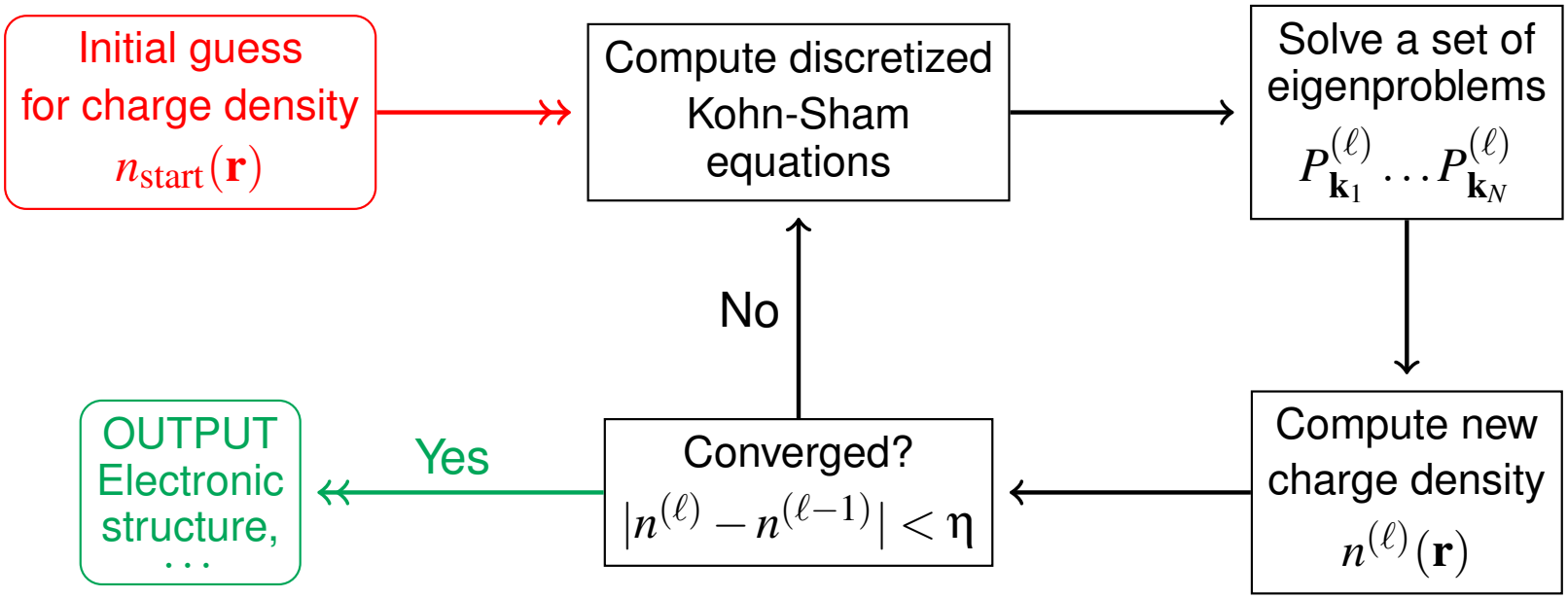
JÜLICH
FORSCHUNGSZENTRUM

An Optimized and Scalable Iterative Solver for Sequences of Dense Eigenvalue Problems

Copper Mountain, Colorado, USA. April 7th | M. Berljafa and **E. Di Napoli**

Motivation

Full-potential Linearized Augmented Plane Waves (FLAPW) self-consistent field cycle



1. every $P^{(\ell)}_{\mathbf{k}} : A^{(\ell)}_{\mathbf{k}} x = B^{(\ell)}_{\mathbf{k}} \lambda x$ is a generalized eigenvalue problem;
2. A and B are **DENSE** and hermitian (B is positive definite);
3. required: lower $2 \div 10$ % of eigenpairs;
4. momentum vector index: $\mathbf{k} = 1 : 10 \div 100$;
5. iteration cycle index: $\ell = 1 : 20 \div 50$.

Outline

Sequences of correlated eigenproblems

The algorithm: Chebyshev Filtered Sub-space Iteration method (**ChFSI**)

ChFSI parallelization and numerical tests

Outline

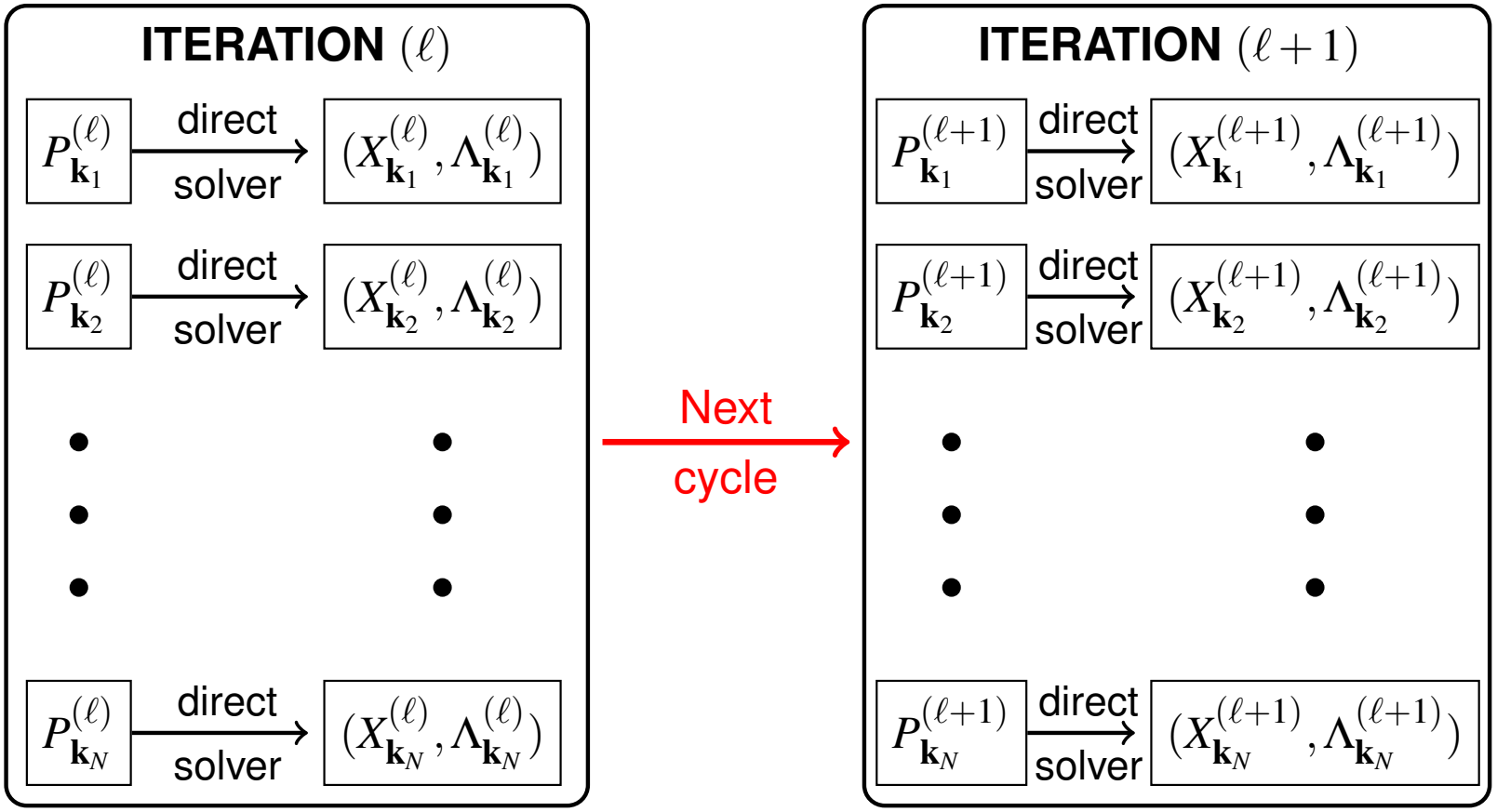
Sequences of correlated eigenproblems

The algorithm: Chebyshev Filtered Sub-space Iteration method (**ChFSI**)

ChFSI parallelization and numerical tests

Sequences of Eigenproblems

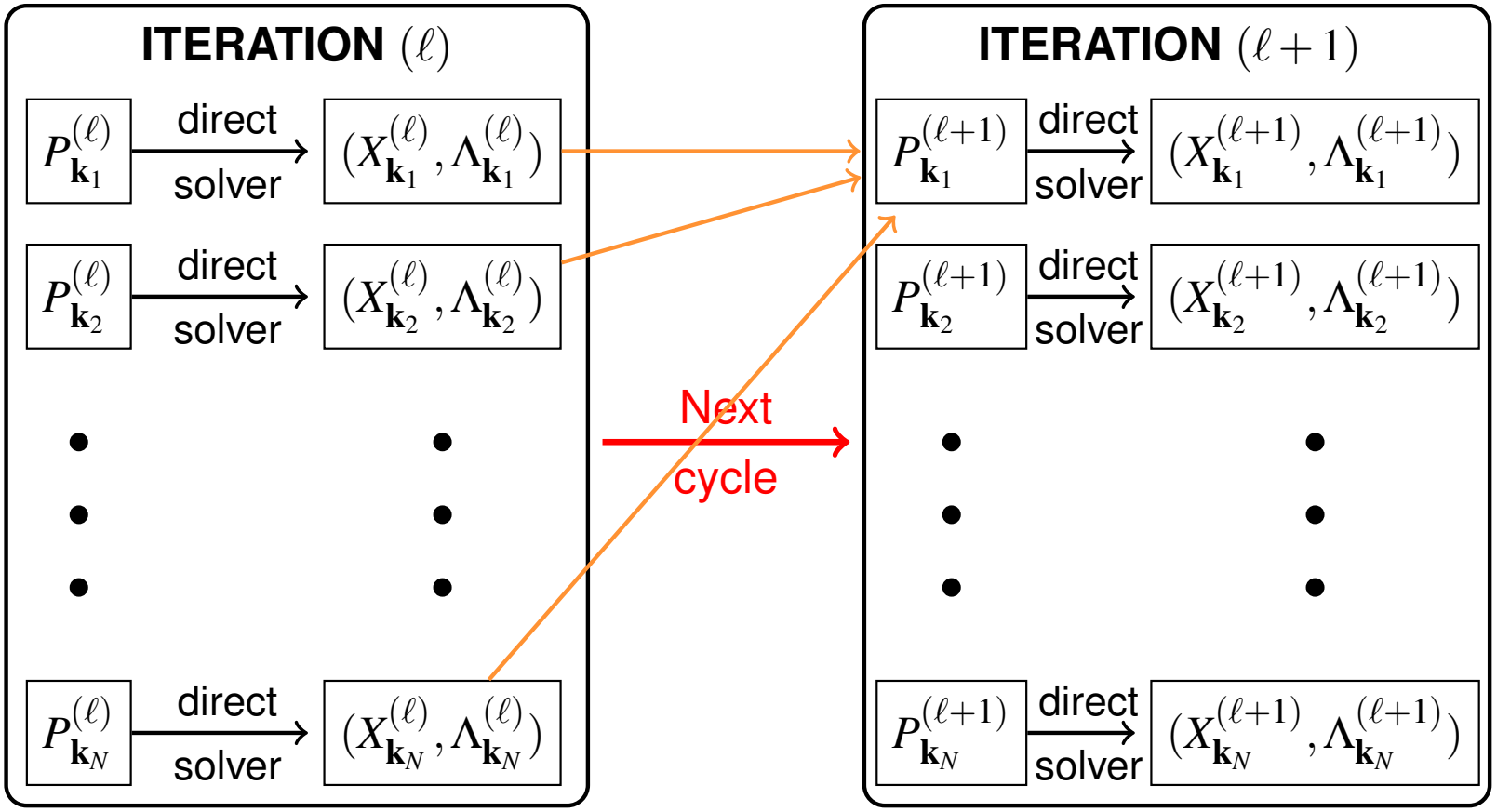
Adjacent iteration cycles



$$X \equiv \{x_1, \ldots, x_n\} \qquad \Lambda \equiv \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$$

Sequences of Eigenproblems

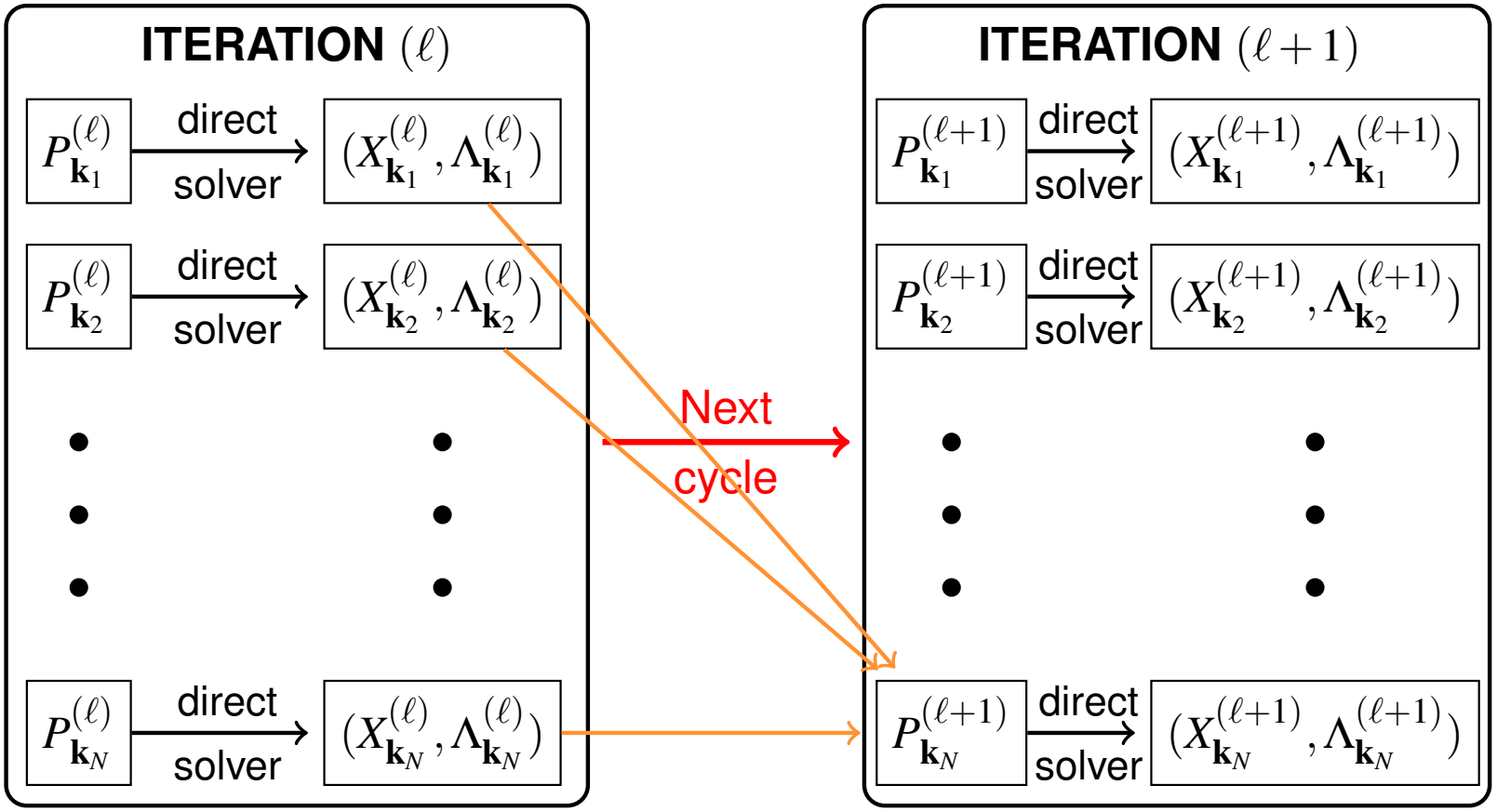
Adjacent iteration cycles



$$X \equiv \{x_1, \ldots, x_n\} \qquad \Lambda \equiv \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$$

Sequences of Eigenproblems

Adjacent iteration cycles



$$X \equiv \{x_1, \ldots, x_n\} \qquad \Lambda \equiv \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$$

Sequences of eigenproblems

Definitions and solving strategies

Definition: Eigenproblem sequence

A sequence of eigenproblems is a finite and index-ordered set of problems $\{P\}_N \doteq P^{(1)} \dots P^{(\ell)} \dots P^{(N)}$ with same size $= n$ such that the eigenpairs of $P^{(\ell)}$ are used (directly or indirectly) to initialize $P^{(\ell+1)}$.

Current solving strategy

- The set of generalized eigenproblems $P^{(1)} \dots P^{(\ell)} P^{(\ell+1)} \dots P^{(N)}$ is handled as a set of disjoint problems $N \times P$;
- Each problem $P^{(\ell)}$ is solved independently using a direct solver as a black-box from a standard library (i.e. ScaLAPACK).

Correlation between eigenproblems

Definition and solving strategies

Definition: Correlation

Two adjacent problems $P^{(\ell+1)}$ and $P^{(\ell)}$ are said to be correlated when the eigenpairs $(X^{(\ell+1)}, \Lambda^{(\ell+1)})$ have some relation with the eigenpairs $(X^{(\ell)}, \Lambda^{(\ell)})$.

Correlation between eigenproblems

Definition and solving strategies

Definition: Correlation

Two adjacent problems $P^{(\ell+1)}$ and $P^{(\ell)}$ are said to be correlated when the eigenpairs $(X^{(\ell+1)}, \Lambda^{(\ell+1)})$ have some relation with the eigenpairs $(X^{(\ell)}, \Lambda^{(\ell)})$.

Uncovering the correlation $\longrightarrow$ extracting information from simulations

- Extracted the matrices of eigenproblems $P^{(1)}, \ldots, P^{(N)}$ from the FLAPW code by running a full simulation;
- Computed the solutions of the full sequence,
  - collected data on **angles** b/w eigenvectors of adjacent eigenproblems;

$$\Theta^{(\ell)}_{\mathbf{k}_i} \equiv \{\theta_1, \ldots, \theta_n\} = \mathrm{diag}\left(\mathbb{1} - \langle X^{(\ell-1)}_{\mathbf{k}_i}, \tilde{X}^{(\ell)}_{\mathbf{k}_i} \rangle\right)$$

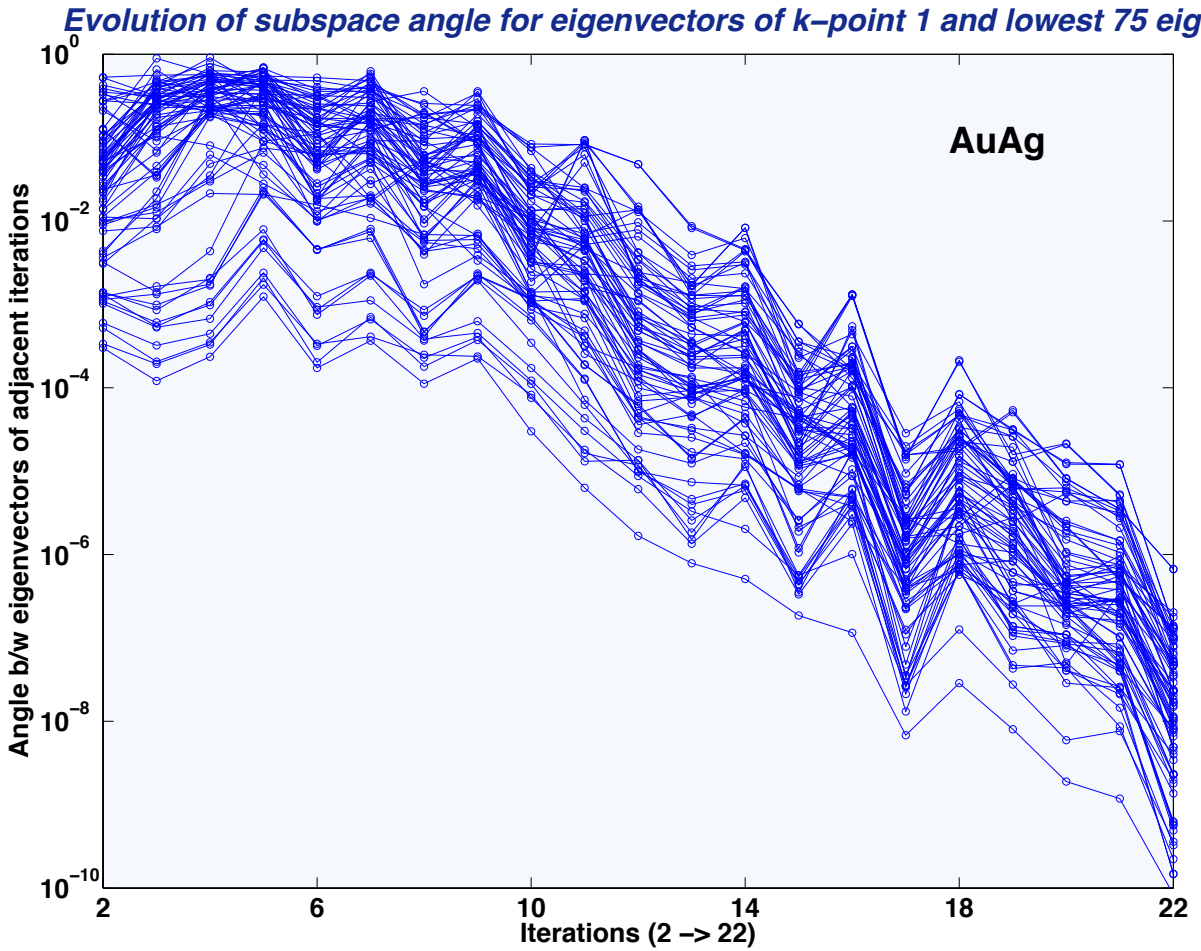
  - uncovered **evolution** of eigenvectors along the sequence

$$\text{for fixed } \mathbf{k}_i \quad \theta^{(2)}_j \gtrsim \theta^{(3)}_j \gtrsim \cdots \gtrsim \theta^{(N)}_j : \quad \theta^{(2)}_j \gg \theta^{(N)}_j$$

Angles evolution

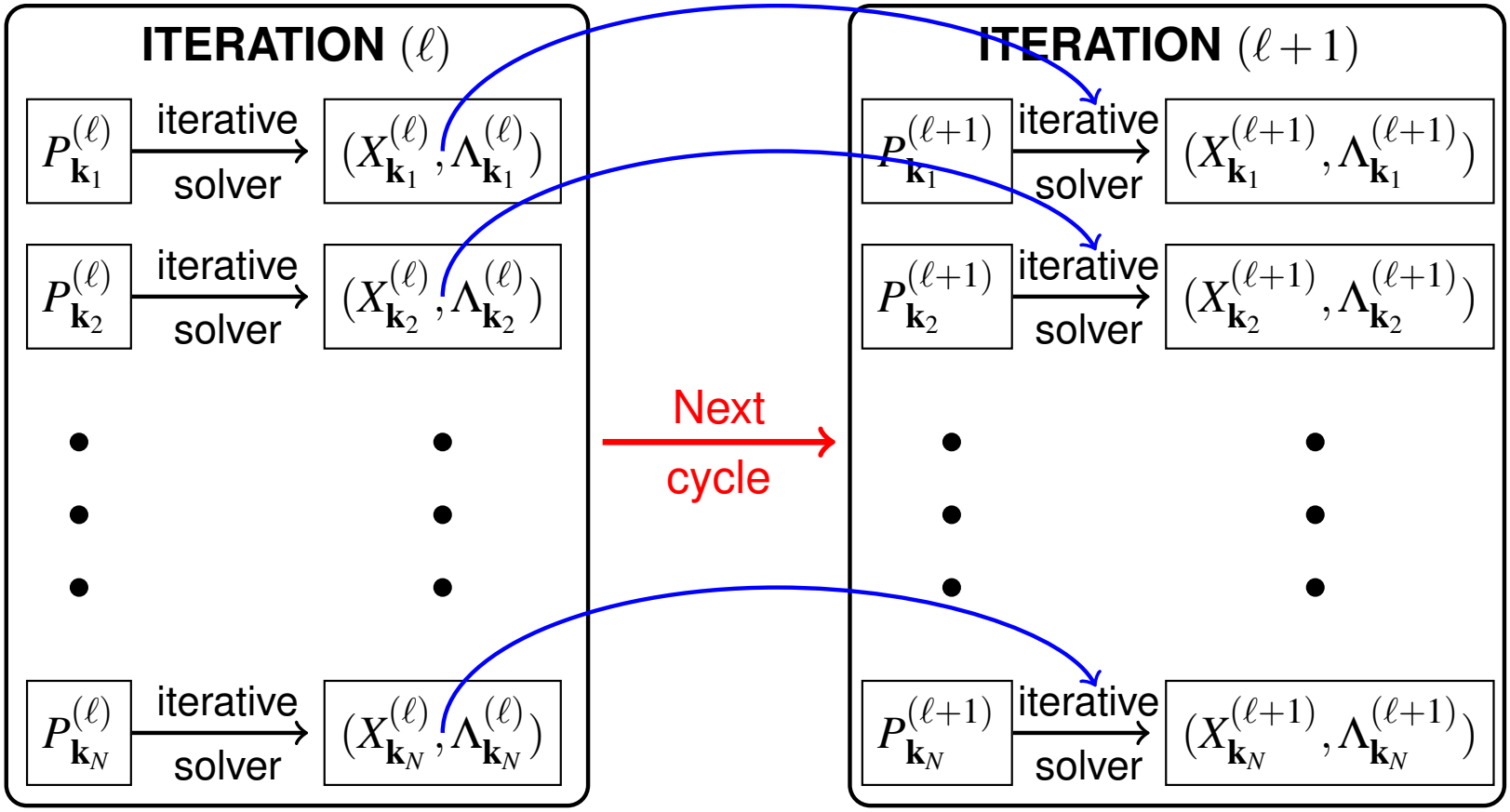
An example

Example: a metallic compound at fixed $\mathbf{k}$



An alternative solving strategy

Adjacent cycles



$$X \equiv \{x_1, \ldots, x_n\} \qquad \Lambda \equiv \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$$

Outline

Sequences of correlated eigenproblems

The algorithm: Chebyshev Filtered Sub-space Iteration method (**ChFSI**)

ChFSI parallelization and numerical tests

Chebyshev Filtered Subspace Iteration method (ChFSI)

Main properties

- eigenproblem needs to be in standard form $A' = L^{-1}AL^{-T}$ with $B = LL^T$
- it accepts the full set of multiple starting vectors $Z_0 \equiv X_{\mathbf{k}_i}^{(\ell-1)}(:, 1 : \mathrm{NEV})$;
- the capacity to solve simultaneously for a substantial portion of eigenpairs;
- augmented with the Chebyshev polynomial filter it has a much faster convergence rate;
- it maximally exploits fast BLAS 3 kernels (`xGEMM`);
- it avoids stalling when facing small clusters of eigenvalues;
- converged eigenvectors can be easily locked;
- the degree of the polynomial can be opportunely optimized.

ChFSI pseudocode

INPUT: Hamiltonian, approximate eigenvectors – Z_0, extreme eigenvalues $\{\lambda_1, \lambda_{NEV}\}$, TOL, DEG.

OUTPUT: NEV wanted eigenpairs (Λ, W).

1. *Lanczos step.* Identify the bounds for the **eigenspectrum interval** corresponding to the wanted eigenspace.

REPEAT UNTIL CONVERGENCE:

2. *Chebyshev filter.* **Filter** a block of vectors $W \longleftarrow Z_0$.
3. Re-orthogonalize the vectors outputted by the filter; $W = QR$.
4. Compute the **Rayleigh quotient** $G = Q^{\dagger} H Q$.
5. Compute the primitive Ritz pairs (Λ, Y) by solving for $GY = Y\Lambda$.
6. Compute the approximate Ritz pairs $(\Lambda, W \leftarrow QY)$.
7. *Check* which one among the Ritz vectors *converged*.
8. *Deflate* and *lock* the converged vectors.

END REPEAT

The core of the algorithm: Chebyshev filter

The basic principle

Theorem

Let $|\gamma| > 1$ and $\mathbb{P}_m$ denote the set of polynomials of degree smaller or equal to m. Then the extremum

$$\min_{p \in \mathbb{P}_m, p(\gamma)=1} \max_{t \in [-1,1]} |p(t)|$$

is reached by

$$p_m(t) \doteq \frac{C_m(t)}{C_m(\gamma)}.$$

where C_m is the Chebyshev polynomial of the first kind of order m, defined as

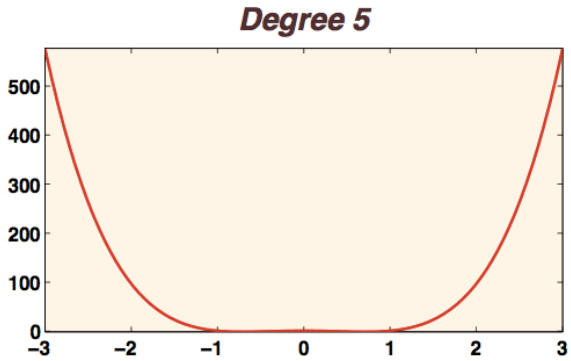
$$C_m(t) = \begin{cases} \cos\left(m \arccos(t)\right), & t \in [-1,1]; \\ \cosh\left(m \operatorname{arccosh}(t)\right), & |t| > 1. \end{cases}$$

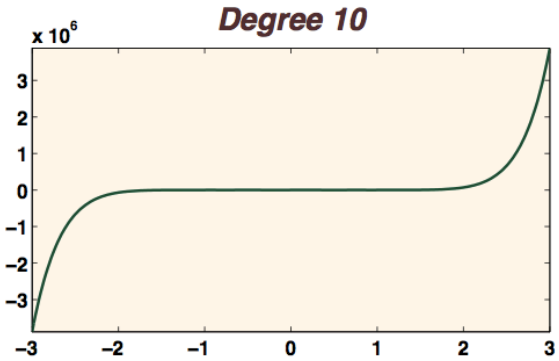
The core of the algorithm: Chebyshev filter

Chebyshev polynomials

A generic vector $v = \sum_{i=1}^{n} s_i x_i$ is very quickly aligned in the direction of the eigenvector corresponding to the extremal eigenvalue λ_1

$$
\begin{aligned}
v^m = p_m(A)v \quad &= \quad \sum_{i=1}^{n} s_i p_m(A) x_i = \sum_{i=1}^{n} s_i p_m(\lambda_i) x_i \\
&= \quad s_1 x_1 + \sum_{i=2}^{n} s_i \frac{C_m\left(\frac{\lambda_i - c}{e}\right)}{C_m\left(\frac{\lambda_1 - c}{e}\right)} x_i \quad \sim \quad \boxed{s_1 x_1}
\end{aligned}
$$





The core of the algorithm: Chebyshev filter

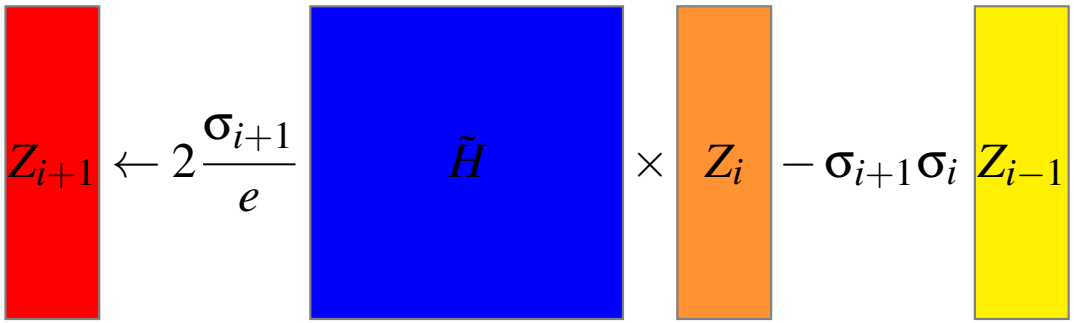
In practice

Three-terms recurrence relation

$$C_{m+1}(t) = 2xC_m(t) - C_{m-1}(t); \qquad m \in \mathbb{N}, \quad C_0(t) = 1, \quad C_1(t) = x$$

$$Z_m \doteq p_m(\tilde{H})\, Z_0 \quad \text{with} \quad \tilde{H} = H - cI_n$$

FOR: $i = 1 \rightarrow \text{DEG} - 1$



xGEMM

END FOR.

Polynomial degree optimization

Convergence ratio and residuals

Definition

The **convergence ratio** for the eigenvector x_i corresponding to eigenvalue $\lambda_i \notin [\alpha, \beta]$ is defined as

$$\tau(\lambda_i) = |\rho_i|^{-1} = \min_{\pm} \left| \frac{\lambda_i - c}{e} \pm \sqrt{\left(\frac{\lambda_i - c}{e}\right)^2 - 1} \right|.$$

The further away λ_i is from the interval $[\alpha, \beta]$ the smaller is $|\rho_i|^{-1}$ and the faster the convergence to x_i is.

For a set of input vectors $V = \{v_1, v_2, \ldots, v_{\text{nev}}\}$

Residuals are a function of m and $|\rho|$

$$\text{Res}(v_1^m) \sim \quad \text{Res}(v_1^{m_0}) \left| \frac{1}{\rho_1} \right|^{(m-m_0)}$$

$$\text{Res}(v_i^m) \sim \quad \text{Res}(v_i^{m_0}) \left| \frac{1}{\rho_i} \right|^{(m-m_0)} + \langle \text{eps} \rangle \frac{|\rho_1|^{(m-m_0)}}{|\rho_i|^{(m-m_0)}} \quad ; \quad k \geq i \geq 2.$$

ChFSI Single Optimization pseudocode

1. *Chebyshev filter.* Initial filter $W \longleftarrow Z_0$. with DEG$= m_0$.
2. Re-orthogonalize $W = QR$ & compute the Rayleigh quotient $G = Q^{\dagger} H Q$.
3. Solve the reduced problem $GY = Y\Lambda$ and compute the approximate Ritz pairs $(\Lambda, W \leftarrow QY)$ and store their residuals $\mathrm{Res}(w_i)$.

REPEAT UNTIL CONVERGENCE:

4. *Optimizer.* Compute the polynomial degrees $m_i \geq \ln \left| \frac{\mathrm{TOL}}{\mathrm{Res}(w_i)} \right| / \ln \|\rho_i\|$.
5. *Chebyshev filter.* Filter $W \longleftarrow Z_0$ with DEG$= m_i$.
6. Re-orthogonalize $W = QR$ & compute the Rayleigh quotient $G = Q^{\dagger} H Q$.
7. Solve the reduced problem $GY = Y\Lambda$ and compute the approximate Ritz pairs $(\Lambda, W \leftarrow QY)$.
8. *lock* the converged vectors.
9. Store the residuals $\mathrm{Res}(w_i)$ of the unconverged vectors.

END REPEAT

Outline

Sequences of correlated eigenproblems

The algorithm: Chebyshev Filtered Sub-space Iteration method (**ChFSI**)

ChFSI parallelization and numerical tests

Experimental tests setup

C++ implementation of ChFSI

- OMPChFSI – OpenMP parallelization for shared memory
- EleChFSI – Elemental (MPI) parallelization for distributed memory

Matrix sizes: 2,600 $\div$ 13,300.

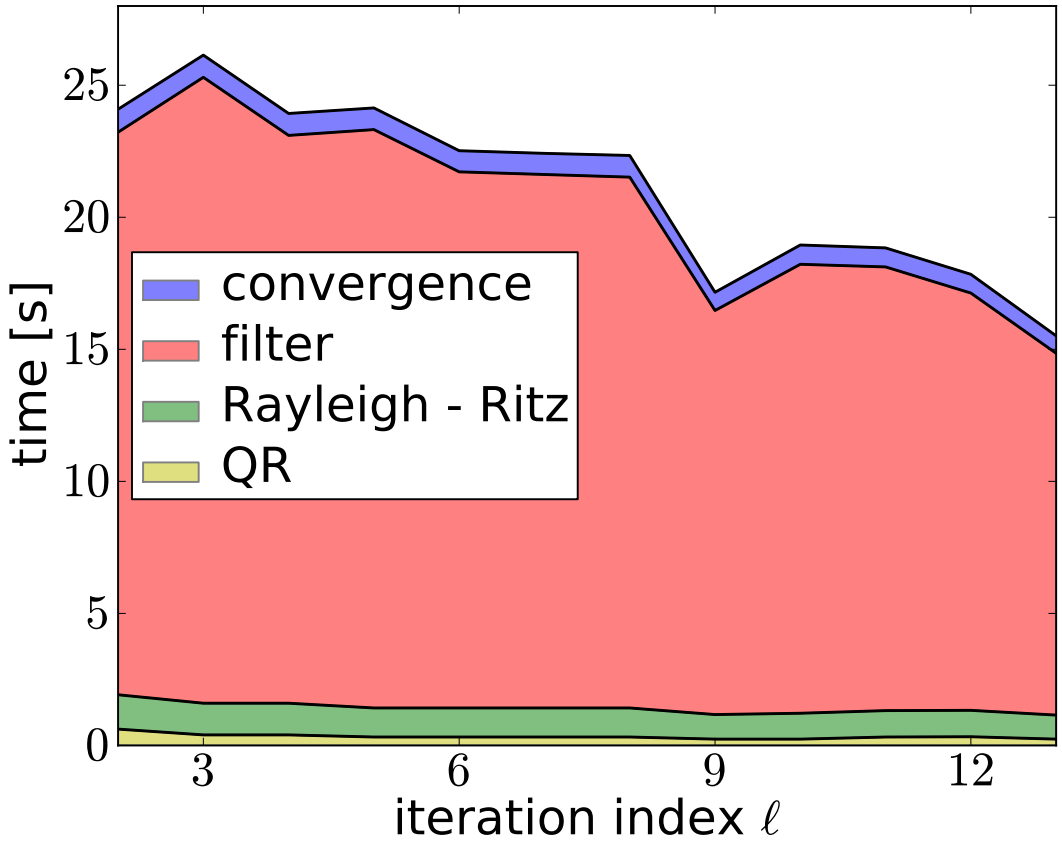
Tests were performed on the JUROPA cluster.

- 2 Intel Xeon 5570 (Nehalem-EP) quad-core processors/node, 2.93GHz;
- 24 GB/node;
- THEORETICAL PEAK PERFORMANCE/CORE=11.71 GFLOPS;
- Minimum absolute tolerance of residuals $\mathrm{Res}(x_i) = 10^{-10}$;
- All numerical data are MEDIAN values over 12 distinct measurements.

ChFSI time profile

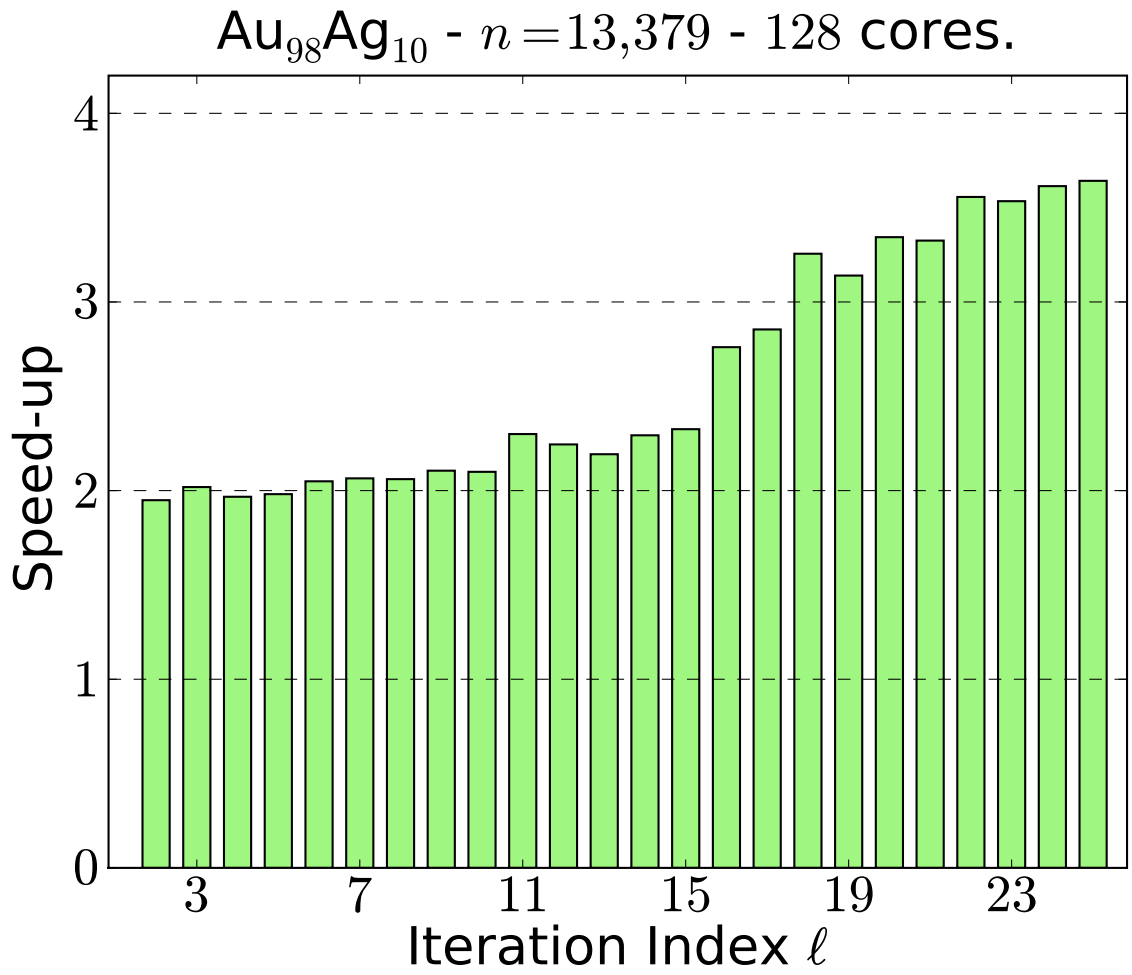
As a function of iteration cycles

Time spent in each stage of the algorithm as a function of the iteration index ℓ for a system of size $n = 9,273$.



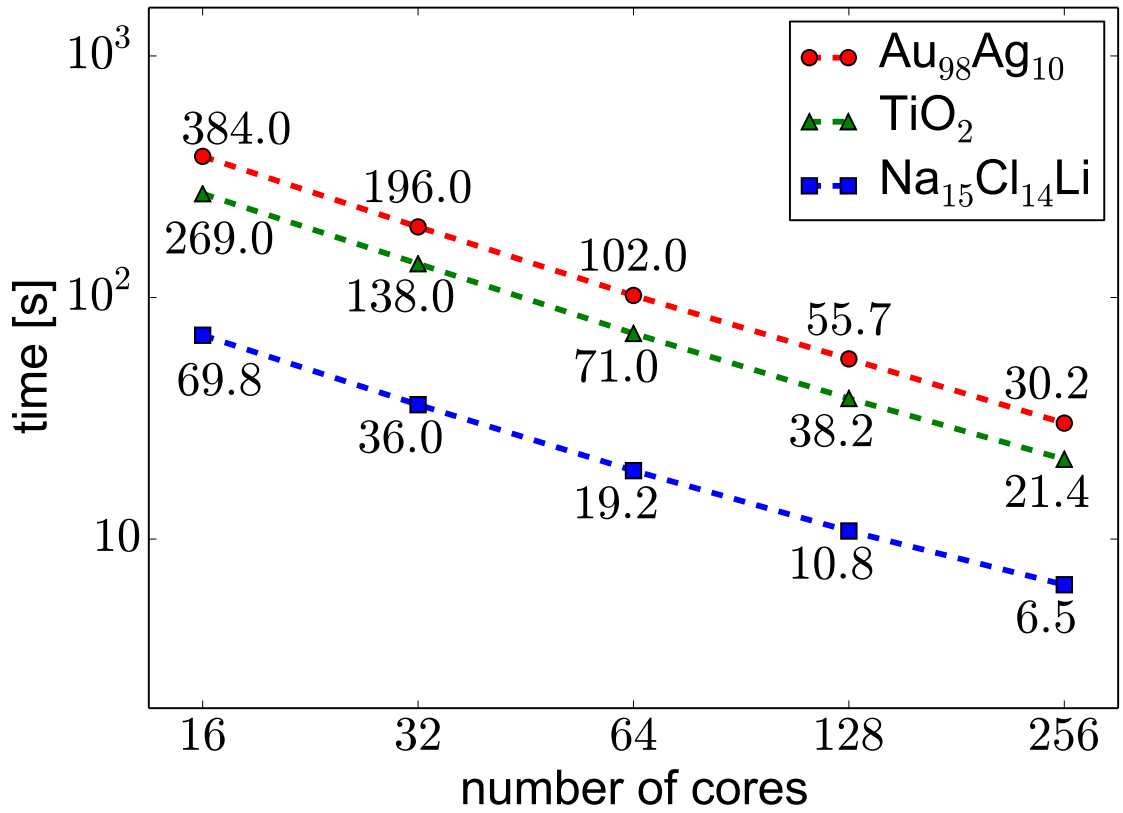
Speed-up

$$\text{Speed-up} = \frac{\text{CPU time (input random vectors)}}{\text{CPU time (input approximate eigenvectors)}}$$



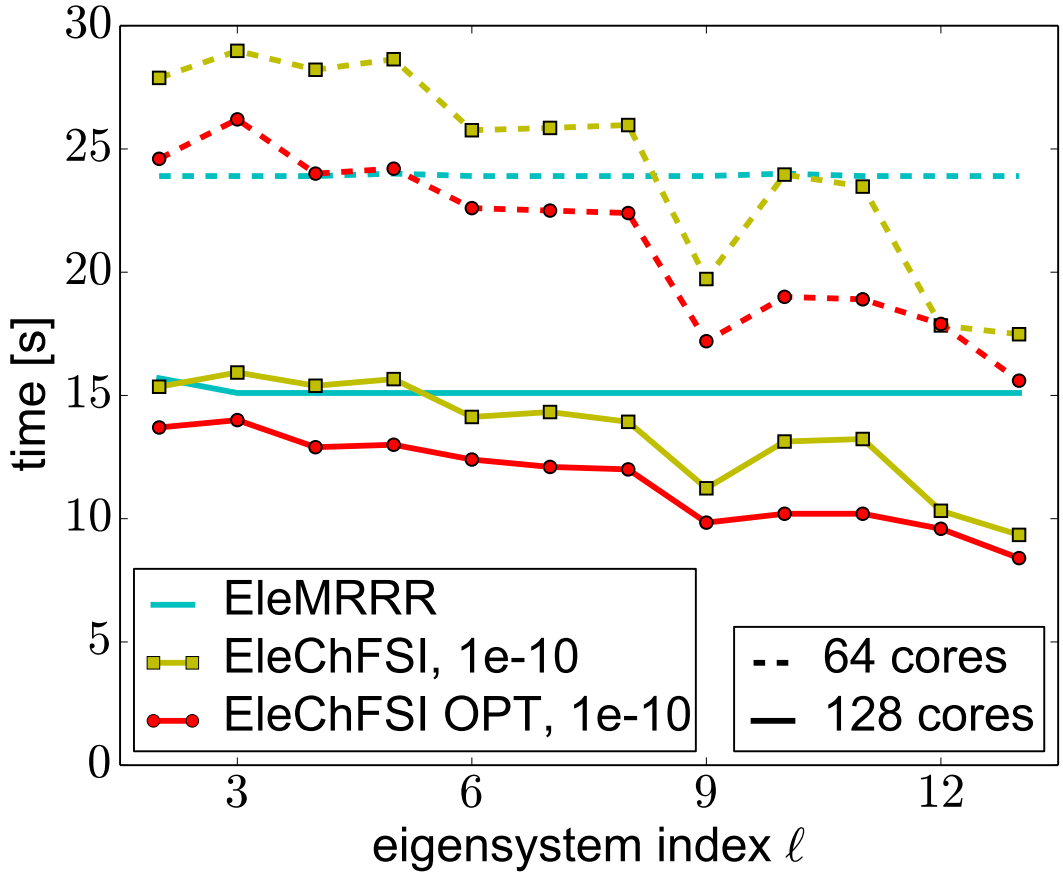
Scalability

Strong Scalability (the size of the eigenproblems are kept fixed while the number of cores is progressively increased) for EleChFSI over two systems of size $n = 13,379 - 12,455 - 9,273$ respectively.



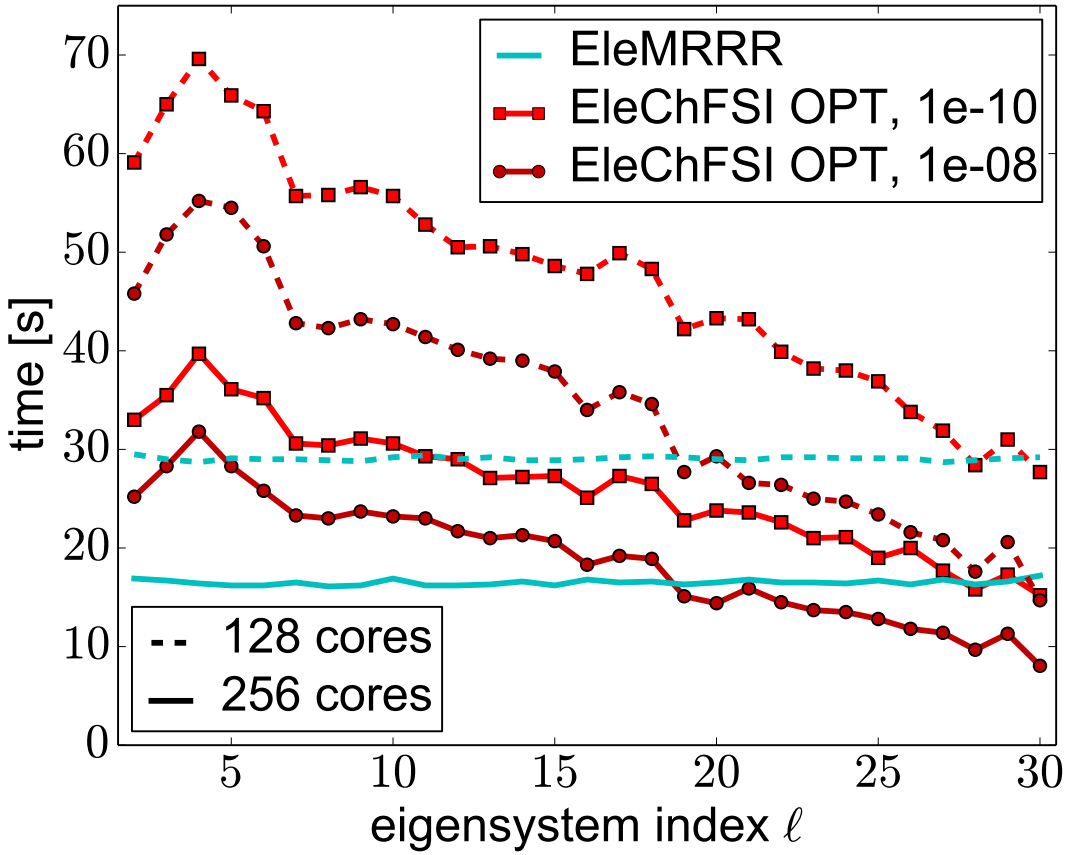
EleChFSI versus direct solvers (parallel MRRR)

For the size of eigenproblems here tested the ScaLAPACK implementation of BXINV is comparable with EleMRRR. For this reason a direct comparison with ScaLAPACK is not included.



EleChFSI versus direct solvers (parallel MRRR)

For the size of eigenproblems here tested the ScaLAPACK implementation of BXINV is comparable with EleMRRR. For this reason a direct comparison with ScaLAPACK is not included.



Conclusions and future work

Algorithmic strategy

Sequences of “correlated” eigenproblems $\Rightarrow$ Tailored algorithms

- Exploiting the correlation of the eigenproblem sequence to speedup the solution of each $P^{(\ell)}$ is a successful strategy;
- Combining iterative methods with kernels for dense linear algebra can pay off.
- The parallelization shows great potential for scalability and parallel efficiency;
- Uncovering information can lead to further algorithmic optimizations;

ONGOING AND FUTURE WORK

1. Fine-tuning filter optimization by profiling eigenvector degrees to convergence along the sequence so as to further reduce complexity;
2. Exploiting different architectures for parallelization (GPUs, Xeon Phi).

References

1. EDN and M.Berliafa
   An Optimized and Scalable Eigensolver for Sequences of Eigenvalue Problems
   To be submitted to Concurrency and Computation: Practice and Experience.
2. EDN, and M. Berljafa
   A Parallel and Scalable Iterative Solver for Sequences of Dense Eigenproblems Arising in FLAPW
   Parallel Processing and Applied Mathematics, Lecture Notes in Computer Science, Vol 8385. [arXiv:1305.5120]
3. EDN, and M. Berljafa
   Block Iterative Eigensolvers for Sequences of Correlated Eigenvalue Problems
   Comp. Phys. Comm. 184 (2013), pp. 2478-2488, [arXiv:1206.3768].
4. EDN, P. Bientinesi, and S. Blügel,
   Correlation in sequences of generalized eigenproblems arising in Density Functional Theory,
   Comp. Phys. Comm. 183 (2012), pp. 1674-1682, [arXiv:1108.2594].