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

Initial guess for charge density $n_{\text{start}}(\mathbf{r})$

Compute discretized Kohn-Sham equations

Solve a set of eigenproblems $P_{\mathbf{k}_1}^{(\ell)} \ldots P_{\mathbf{k}_N}^{(\ell)}$

Converged?

Yes

OUTPUT Electronic structure, ... | $|n^{(\ell)} - n^{(\ell-1)}| < \eta$

No

Compute new charge density $n^{(\ell)}(\mathbf{r})$

1. every $P_{\mathbf{k}}^{(\ell)} : A_{\mathbf{k}}^{(\ell)} \mathbf{x} = B_{\mathbf{k}}^{(\ell)} \lambda \mathbf{x}$ 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: $\ell = 1 : 20 \div 50$. 

Copper Mountain, Colorado, USA. April 7th

M. Berljafa and E. Di Napoli
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

\[
\begin{align*}
\text{ITERATION } (\ell) &:&
\begin{array}{c}
P^{(\ell)}_{k_1} \quad \text{direct solver} \\
P^{(\ell)}_{k_2} \quad \text{direct solver}
\end{array} &:
\begin{array}{c}
(X^{(\ell)}_{k_1}, \Lambda^{(\ell)}_{k_1}) \\
(X^{(\ell)}_{k_2}, \Lambda^{(\ell)}_{k_2})
\end{array}
\vspace{0.5cm}
\vdots &:& \vdots
\vspace{0.5cm}
\vdots &:& \vdots
\vspace{0.5cm}
\vdots &:& \vdots
\vspace{0.5cm}
\vdots &:& \vdots
\vspace{0.5cm}
\vdots &:& \vdots
\end{align*}
\]

\[X \equiv \{x_1, \ldots, x_n\}\]

\[
\begin{align*}
\text{ITERATION } (\ell + 1) &:&
\begin{array}{c}
P^{(\ell+1)}_{k_1} \quad \text{direct solver} \\
P^{(\ell+1)}_{k_2} \quad \text{direct solver}
\end{array} &:
\begin{array}{c}
(X^{(\ell+1)}_{k_1}, \Lambda^{(\ell+1)}_{k_1}) \\
(X^{(\ell+1)}_{k_2}, \Lambda^{(\ell+1)}_{k_2})
\end{array}
\vspace{0.5cm}
\vdots &:& \vdots
\vspace{0.5cm}
\vdots &:& \vdots
\vspace{0.5cm}
\vdots &:& \vdots
\vspace{0.5cm}
\vdots &:& \vdots
\vspace{0.5cm}
\vdots &:& \vdots
\end{align*}
\]

\[\Lambda \equiv \text{diag}(\lambda_1, \ldots, \lambda_n)\]

Next cycle
Sequences of Eigenproblems

Adjacent iteration cycles

\[ \begin{align*} 
\text{ITERATION (}\ell\text{)} & \quad \rightarrow \\
P^{(\ell)}_{k_1} & \quad \text{direct solver} \\
(X^{(\ell)}_{k_1}, \Lambda^{(\ell)}_{k_1}) & \\
\vdots & \\
\vdots & \\
\vdots & \\
\vdots & \\
(X^{(\ell)}_{k_N}, \Lambda^{(\ell)}_{k_N}) & \quad \rightarrow \\
P^{(\ell+1)}_{k_1} & \quad \text{direct solver} \\
(X^{(\ell+1)}_{k_1}, \Lambda^{(\ell+1)}_{k_1}) & \\
\vdots & \\
\vdots & \\
\vdots & \\
\vdots & \\
(X^{(\ell+1)}_{k_{N}}, \Lambda^{(\ell+1)}_{k_{N}}) & \quad \rightarrow \\
P^{(\ell+1)}_{k_2} & \quad \text{direct solver} \\
(X^{(\ell+1)}_{k_2}, \Lambda^{(\ell+1)}_{k_2}) & \\
\vdots & \\
\vdots & \\
\vdots & \\
\vdots & \\
(X^{(\ell+1)}_{k_{N}}, \Lambda^{(\ell+1)}_{k_{N}}) & \\
P^{(\ell+1)}_{k_N} & \quad \text{direct solver} \\
(X^{(\ell+1)}_{k_N}, \Lambda^{(\ell+1)}_{k_N}) \\
\end{align*} \]

\[ X \equiv \{x_1, \ldots, x_n\} \]

\[ \Lambda \equiv \text{diag}(\lambda_1, \ldots, \lambda_n) \]
Sequences of Eigenproblems
Adjacent iteration cycles

<table>
<thead>
<tr>
<th>ITERATION ((\ell))</th>
<th>ITERATION ((\ell + 1))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(P_{k_1}^{(\ell)})</td>
<td>(P_{k_1}^{(\ell+1)})</td>
</tr>
<tr>
<td>(P_{k_2}^{(\ell)})</td>
<td>(P_{k_2}^{(\ell+1)})</td>
</tr>
<tr>
<td>(P_{k_N}^{(\ell)})</td>
<td>(P_{k_N}^{(\ell+1)})</td>
</tr>
</tbody>
</table>

\[ X \equiv \{x_1, \ldots, x_n\} \]

\[ \Lambda \equiv \text{diag}(\lambda_1, \ldots, \lambda_n) \]
Sequences of eigenproblems
Definitions and solving strategies

Definition: Eigenproblem sequence
A sequence of eigenproblems is a finite and index-ordered set of problems
\( \{P\}_N \equiv P^{(1)} \ldots P^{(\ell)} \ldots P^{(N)} \) 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 \( P^{(1)} \ldots P^{(\ell)} P^{(\ell+1)} \ldots P^{(N)} \) 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 $(X^{(\ell)}, \Lambda^{(\ell)})$. 

---

**Uncovering the correlation**

→ Extracting information from simulations

Extracted the matrices of eigenproblems $P^{(1)},...,P^{(N)}$ 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; $\Theta^{(\ell)}_k \equiv \{\theta_1,...,\theta_n\} = \text{diag}(1-\langle X^{(\ell-1)}_k, \tilde{X}^{(\ell)}_k \rangle)$ uncovered evolution of eigenvectors along the sequence for fixed $k_i$; $\theta^{(2)}_j \gg \theta^{(3)}_j \gg \cdots \gg \theta^{(N)}_j$: $\theta^{(2)}_j \gg \theta^{(N)}_j$. 

Copper Mountain, Colorado, USA. April 7th

M. Berljafa and E. Di Napoli
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 → 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;
    \[ \Theta_{k_i}^{(\ell)} \equiv \{\theta_1, \ldots, \theta_n\} = \text{diag} \left( 1 - \langle X_{k_i}^{(\ell-1)}, \tilde{X}_{k_i}^{(\ell)} \rangle \right) \]
  - uncovered evolution of eigenvectors along the sequence
    \[ \text{for fixed } k_i, \quad \theta_j^{(2)} \gtrsim \theta_j^{(3)} \gtrsim \cdots \gtrsim \theta_j^{(N)} : \quad \theta_j^{(2)} \gg \theta_j^{(N)} \]
Angles evolution

An example

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

Evolution of subspace angle for eigenvectors of $k$-point 1 and lowest 75 eigs

Angle b/w eigenvectors of adjacent iterations
An alternative solving strategy

Adjacent cycles

\[ P^{(\ell)}_{k_1} \xrightarrow{\text{iterative solver}} (X^{(\ell)}_{k_1}, \Lambda^{(\ell)}_{k_1}) \]
\[ P^{(\ell)}_{k_2} \xrightarrow{\text{iterative solver}} (X^{(\ell)}_{k_2}, \Lambda^{(\ell)}_{k_2}) \]
\[ P^{(\ell)}_{k_N} \xrightarrow{\text{iterative solver}} (X^{(\ell)}_{k_N}, \Lambda^{(\ell)}_{k_N}) \]

\[ \Lambda \equiv \text{diag}(\lambda_1, \ldots, \lambda_n) \]

\[ X \equiv \{x_1, \ldots, x_n\} \]
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 $A' = L^{-1} A L^{-T}$ with $B = L L^T$
- it accepts the full set of multiple starting vectors $Z_0 \equiv X^{(\ell-1)}_{k_i} (:, 1 : \text{NEV})$;
- 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 – Z₀, extreme eigenvalues \{\lambda₁, \lambda_{\text{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 ←− Z₀.

3. Re-orthogonalize the vectors outputted by the filter; W = QR.

4. Compute the **Rayleigh quotient** \( G = Q^\dagger HQ \).

5. Compute the primitive Ritz pairs (Λ, Y) by solving for \( GY = YΛ \).

6. Compute the approximate Ritz pairs (Λ, W ← 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

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

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

$$C_m(t) = \begin{cases} \cos (m \arccos(t)), & t \in [-1, 1]; \\ \cosh (m \arccosh(t)), & |t| > 1. \end{cases}$$
The core of the algorithm: Chebyshev filter

Chebyshev polynomials

A generic vector \( v = \sum_{i=1}^{n} s_ix_i \) is very quickly aligned in the direction of the eigenvector corresponding to the extremal eigenvalue \( \lambda_1 \)

\[
v^m = p_m(A)v = \sum_{i=1}^{n} s_ip_m(A)x_i = \sum_{i=1}^{n} s_ip_m(\lambda_i)x_i
\]

\[
= s_1x_1 + \sum_{i=2}^{n} s_i \frac{C_m(\frac{\lambda_i-c}{e})}{C_m(\frac{\lambda_1-c}{e})}x_i \sim s_1x_1
\]
The core of the algorithm: Chebyshev filter

In practice

Three-terms recurrence relation

\[ C_{m+1}(t) = 2x C_m(t) - C_{m-1}(t); \quad m \in \mathbb{N}, \quad C_0(t) = 1, \quad C_1(t) = x \]

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

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

\[ Z_{i+1} \leftarrow 2 \frac{\sigma_{i+1}}{e} \tilde{H} Z_i \quad - \sigma_{i+1} \sigma_i Z_{i-1} \]

\text{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

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

The further away $\lambda_i$ 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 $V = \{v_1, v_2, \ldots, v_{nev}\}$

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

$$\text{Res}(v^m_1) \sim \text{Res}(v^{m_0}_1) \left| \frac{1}{\rho_1} \right|^{(m-m_0)}$$

$$\text{Res}(v^m_i) \sim \text{Res}(v^{m_0}_i) \left| \frac{1}{\rho_i} \right|^{(m-m_0)} + \langle \text{eps} \rangle \frac{|\rho_1|^{(m-m_0)}}{|\rho_i|^{(m-m_0)}} ; \quad k \geq i \geq 2.$$
ChFSI Single Optimization pseudocode

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

**Repeat Until Convergence:**

4. *Optimizer.* Compute the polynomial degrees $m_i \geq \ln \left| \frac{\text{TOL}}{\text{Res}(w_i)} \right| / \ln \|\rho_i\|$.
5. *Chebyshev filter.* Filter $W \leftarrow Z_0$ with DEG$= m_i$.
6. Re-orthogonalize $W = QR$ & compute the Rayleigh quotient $G = Q^\dagger HQ$.
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 $\text{Res}(w_i)$ 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 ÷ 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 $\text{Res}(x_i) = 10^{-10}$;
- 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 $\ell$ for a system of size $n = 9,273$. 
Speed-up

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

\[
\text{Au}_{98}\text{Ag}_{10} - n = 13,379 - 128 \text{ cores.}
\]

Copper Mountain, Colorado, USA. April 7th

M. Berljafa and E. Di Napoli
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.

![Graph showing the relationship between the number of cores and time for different systems. The y-axis represents time in seconds, and the x-axis represents the number of cores, ranging from 16 to 256. The systems include $Au_{98}Ag_{10}$, $TiO_2$, $Na_{15}Cl_{14}Li$.](image)
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 ⇒ 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. Berljafa
   *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*

3. EDN, and M. Berljafa
   *Block Iterative Eigensolvers for Sequences of Correlated Eigenvalue Problems*

4. EDN, P. Bientinesi, and S. Blügel,
   *Correlation in sequences of generalized eigenproblems arising in Density Functional Theory,*