Iterative eigensolvers for DFT

Clémentine Barat

CEA, DAM, DIF, F-91297 Arpajon, France

Université Paris-Saclay, CEA, Laboratoire Matière en Conditions Extrêmes, 91680 Bruyères-le-Châtel, France

January 29, 2024

Contents

Eigensolver in the Self-consistent field

In the self consistent field, one needs to solve an eigenvalue problem at each step. This is a costly operation that can be optimized using iterative eigensolver, which can be mixed with the self-consistent iterations.

Algorithm Self-consistent iteration with an exact eigensolver

$H \leftarrow$ RandomHamiltonian
while H not converged **do**
  $\lambda, X \leftarrow$ EigenSolver(H)
  $H \leftarrow$ Hamiltonian(X)
end while

Algorithm Self-consistent iteration with an iterative eigensolver

$X \leftarrow$ RandomVectors(N, m)
$H \leftarrow$ Hamiltonian(X)
while H not converged **do**
  $\lambda, X \leftarrow$ EigenSolverStep(H, X)
  $H \leftarrow$ Hamiltonian(X)
end while

The problem

We want to find the m smallest eigenvalues of a Hermitian matrix $H \in \mathbb{C}^{N \times N}$ (the discretized Hamiltonian) and the associated eigenvectors.
We denote :

- $\lambda_1, \ldots, \lambda_N$ the eigenvalues of H in ascending order.
- $u_1, \ldots, u_N$ some associated (orthogonal) eigenvectors.
- $\mathcal{U} = \text{Span}\{u_1, \ldots, u_m\}$ the subspace of the eigenvectors we are looking for.

Iterative Eigensolvers

Abinit uses two types of iterative eigensolvers :

- Iterativ minimization on vectors : An iterative minimization method is used to minimize the energy, yielding a set of minimizing vectors.
- Iterativ methods on subspaces : The sought eigensubspace is computed iteratively.

In both cases, the algorithms have the same structure :

1. A test vectors or a test subspace is build from the previous iteration.
2. The Rayleigh-Ritz method is used to retrieve the eigenvalues from the set of minimizing vectors or estimated eigensubspace.

Rayleigh-Ritz method

The Rayleigh-Ritz method gives an approximation of the eigenvalues and eigenvectors in a given subspace.

Algorithm Rayleigh-Ritz

function RayleighRitz(H, X)
  $X \leftarrow \text{Ortho}(X)$
  $R \leftarrow X^H H X$
  $\lambda, Y \leftarrow \text{Eigen}(R)$
  $U \leftarrow XY$
  return λ, U
end function

Diagonalization of the matrix H projected onto the subspace defined by X.

The full diagonalization of the reduced matrix $X^H H X$ is done with iterative methods that usually scale as $\mathcal{O}(m^3)$ (where m is the matrix size).

Minimization algorithms for eigenvalue problems

- Eigenvalue problem : Find the m smallest eigenpairs $(\lambda_1, x_1), \ldots, (\lambda_m, x_m)$ in $\mathbb{R} \times \mathbb{C}^N$ such that for $1 \leq i \leq m$,

$$Hx_i = \lambda_i x_i \tag{1}$$

- Constrained minimization problem :

$$\begin{aligned} \min_{X \in \mathbb{C}^{N \times m}} \quad & E(X) = X^H H X \\ \text{s.t.} \quad & X^H X = I_m \end{aligned} \tag{2}$$

The solutions of (2) represent the same subspace as the solutions of (1).

The eigenvalue problem can be solved using an optimization algorithm and the the Rayleigh-Ritz procedure to retrieve the eigenvectors from the minimizing X.

Standard conjugate gradient algorithm for quadratic minimization

Quadratic optimization problem

$$\min_{x \in \mathbb{R}^N} q(x) = \frac{1}{2} x^T A x - bx$$

with $A \in \mathbb{R}^{N \times N}$ positive definite and $b \in \mathbb{R}^N$.

For a basis $d_1, \ldots, d_N$ of $\mathbb{R}^N$ with mutually A-conjugate vectors ($d_i^T A d_j = 0$ for $i \neq j$), q can be independently minimized on each direction d_i.

- Initialisation step :
  - Initial guess x_0.
  - Initial line search direction, in the steepest descent direction $d_0 = -g_0$, $g_0 = Ax_0 - b$.
- Iterations :
  - New estimation by minimizing q along the line search direction

$$\alpha_{n+1} = \arg\min_{\alpha \in \mathbb{R}} q(x_n + \alpha d_n) = \frac{-d_n^T g_n}{d_n^T A d_n}$$

$$x_{n+1} = x_n - \alpha_{n+1} d_n$$

  - New line search direction : the steepest descent direction modified to be A-conjugate to the already minimized directions

$$g_{n+1} = A x_{n+1} - b$$

$$\beta_{n+1} = \frac{g_{n+1}^T A d_n}{d_n^H A d_n}$$

$$d_{n+1} = -g_{n+1} + \beta_{n+1} d_n$$

Conjugate gradient algorithm for eigenvalue problems

Eigenvalue problem

$$\min_{x \in \mathbb{C}^N} \lambda(x) = \frac{x^H H x}{x^H x}$$

with $H \in \mathbb{C}^{N \times N}$ hermitian.

The conjugate gradient algorithm can be applied to this non quadratic optimization problem, with the following modifications :

- The gradient of λ in x_n is now $g_n = Hx_n - \lambda(x_n)x_n$.
- $d_n = -g_n + \beta_n d_{n-1}$ with several options for β_n.The directions d_n are "conjugated" with the previous direction d_{n-1} in a sense that depends on the choice of β.
- $\alpha_{n+1} = \arg\min_{\alpha \in \mathbb{C}} \lambda(x_n + \alpha d_n)$ is now the resolution of a 2 dimension eigenvalue problem (The Rayleigh-Ritz method on the subspace spanned by x_n and d_n).
  - 3-term variant : x_{n+1} obtained using Rayleigh-Ritz on Span$\{x_n, d_n, x_{n-1}\}$.

Preconditioning

Ideally, we would have the search direction directly proportional to the error $x_n - u_1$.

- $x_n = \sum a_i u_i$
- $g_n = Hx_n - \lambda(x_n)x_n = \sum a_i(\lambda_i - \lambda(x_n))u_i$
- $Tg_n = \sum a_i(\lambda_i - \lambda(x_n))Tu_i$

If for $i \neq 1$, $Tu_i \approx \frac{1}{\lambda_i - \lambda_1} u_i$, Tg_n will be close to be proportional to $x_n - u_1$. A preconditioner is an invertible matrix $T \in \mathbb{C}^{N \times N}$ such that $Tu_i \approx \frac{1}{\lambda_i - \lambda_1} u_i$ for $i \neq 1$.

Projected Conjugate Gradient algorithm

Entire constrained optimization problem :

$$\min_{X \in \mathbb{C}^{N \times m}} E(X) = X^H H X$$
$$\text{s.t. } X^H X = I_m$$

The bands are computed successively using the conjugate gradient method with n_{line} iterations, with the additional condition that the line search directions d_n must also be orthogonal to the already computed bands.

Iterations over the bands $1 \leq i \leq m$:

- Initialisation step :
  - Initial guess x_0^i orthogonal to $x^1, \ldots, x^{m-1}$
  - Initial line search direction orthogonal to the already computed bands
    $d_0^i = -Tg_0^i + \gamma^1 x^1 + \ldots + \gamma^{i-1} x^{i-1}$
    ($\gamma^1, \ldots, \gamma^{i-1}$ Gram-Schmidt coefficients).
- Iterations $1 \leq n+1 \leq n_{\text{line}}$:
  - New estimate (2D or 3D Rayleigh-Ritz)
    $$x_{n+1}^i = \alpha_1 x_n^i + \alpha_2 d_n \ (+\alpha_3 x_{n-1})$$
  - New line search direction orthogonal to the already computed bands
    $$d_{n+1}^i = -Tg_{n+1}^i + \beta d_n^i + \gamma^1 x^1 + \ldots + \gamma^{i-1} x^{i-1}$$
- $x^i = x_{n_{\text{line}}}^i$

LOBPCG : Locally Optimised Block Preconditioned Conjugate Gradient

LOBPCG is similar to the Projected Conjugate Gradient but the bands are computed in blocks instead of individually.

Iterations over the blocks $1 \leq i \leq n_{\text{blocks}}$:

- Initialisation step :
  - Initial guess $x_0^{i,1}, \ldots, x_0^{i,m_b}$
  - Initial search directions orthogonal to the already computed blocks

$$d_0^{i,j} = -T(Hx_0^{i,j} - \lambda(x_0^{i,j})x_0^{i,j}) + \sum_{l=1}^{i-1}\sum_{k=1}^{m_b} \gamma^{l,k} x^{l,k}$$

- Iterations $1 \leq n+1 \leq n_{\text{line}}$:
  - New estimate $x_{n+1}^{i,1}, \ldots, x_{n+1}^{i,m_b}$ obtained with a Rayleigh-Ritz over $\text{Span}\{x_n^{i,1}, \ldots, x_n^{i,m_b}, d_n^{i,1}, \ldots, d_n^{i,m_b}, x_{n-1}^{i,1}, \ldots, x_{n-1}^{i,m_b}\}$ (dimension $3m_b$).
  - New search directions for $1 \leq j \leq m_b$:

$$d_{n+1}^{i,j} = -T(Hx_{n+1}^{i,j} - \lambda(x_{n+1}^{i,j})x_{n+1}^{i,j}) + \sum_{k=1}^{m_b} \beta_k d_n^{i,k} + \sum_{l=1}^{i-1}\sum_{k=1}^{m_l} \gamma^{l,k} x^{l,k}$$

- $x^{i,j} = x_{n_{\text{line}}}^{i,j}$ for $1 \leq j \leq m_b$

Notes on LOBPCG

- Convergence rate is determined by n_{line}, which is the number of time the Hamiltonian operator H is applied.
- Parallelization possibilities in each block : The m_b new search directions can be computed in parallel for each block.
- n_{line} Rayleigh-Ritz in dimension $3m_b$ per blocks.
- The different blocks must be calculated one at a time.
- A balance must be found between parallelization of the Hamiltonian application and higher dimension Rayleigh-Ritz.
- Fast convergence thanks to preconditioning.
- One Rayleigh-Ritz in dimension m per SCF step.

Using LOBPCG in Abinit

Abinit keywords :

- `wfoptalg` : set to 4, 14 or 114 to select LOBPCG.
- `bandpp` : number of band per processor.
- `npband` : number of processor used to parallelize over bands.
- `nline` : number n_{line} of local search per SCF step.

Block size : `npband` $\times$ `bandpp`
Number of blocks : `nband`/(`npband` $\times$ `bandpp`)

RMM-DIIS : Residual Minimization Method - Direct Inversion in the Iterative Subspace

Minimizing the residual of each band $\|Hx - \lambda(x)x\|$ instead of the Rayleigh quotient $x^H Hx / x^H x$ to avoid orthogonalization at each step.

The residual has local minima at each eigenvectors. Once the m first local minima are roughly located, they can each be computed in parallel with a local minimization technique.

1. Perform a few iterations of an iterative eigensolver (e.g. LOBPCG)
2. Perform an iterative minimization of each residual $\|Hx_i - \lambda(x_i)x_i\|$ independently.

- The method is inherently parallel as each band can be computed independently.
- The method is unstable as the final result depends of the initial guess. Some eigenvalues could be missed.
- No final Rayleigh-Ritz diagonalization needed.

Subspace iteration methods

In subspace iteration methods, we try to approximate directly the subspace $\mathcal{U}$ spanned by sought eigenvectors.

- Initialization : Random initial subspace of dimension m :

$$\mathcal{X}_0 = \text{Span}\{x_1^0, \ldots, x_m^0\}$$

- Iterations : Applying a filter $f : \mathbb{C}^N \to \mathbb{C}^N$ well chosen :

$$\mathcal{X}_{k+1} = f(\mathcal{X}_k) = \text{Span}\{f(x_1^k), \ldots, f(x_m^k)\}$$

  The filter must amplify the components of the vectors in $\mathcal{U}$ and attenuate those in a complement of $\mathcal{U}$ so that $\mathcal{X}_k$ converges to $\mathcal{U}$.

- Final step : Applying the Rayleigh-Ritz method to retrieve the eigenvectors from the subspace estimation.

Polynomial and rational filters

For (λ_i, u_i) an eigenpair, $Hu_i = \lambda_i u_i$ and $H^{-1}u_i = \frac{1}{\lambda_i}u_i$. So for Q a polynomial or a rational fraction, $Q(H)u_i = Q(\lambda_i)u_i$ and for a vector $v = \sum_{i=1}^{N} \alpha_i u_i$, we have

$$Q(H)v = \sum_{i=1}^{N} \alpha_i Q(\lambda_i) u_i.$$

By choosing a large Q on $[\lambda_1, \lambda_m]$ and a small Q on $[\lambda_{m+1}, \lambda_N]$ we obtain a suitable filter $f = Q(H)$ which will allow us to extract from each vector its component on $\mathcal{U}$.

Chebyshev Filtering (ChebFi)

Chebyshev polynomials of the first kind

- $T_n(x) \in [-1, 1]$ for $x \in [-1, 1]$
- T_n grows rapidly outside of $[-1, 1]$

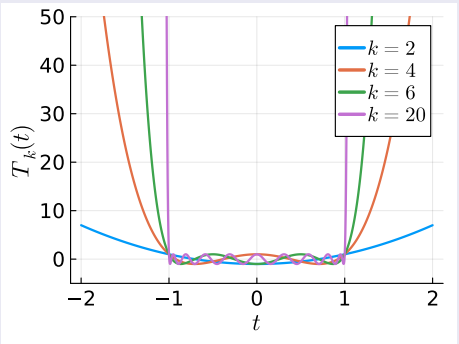


Figure – Chebyshev polynomials

Chebyshev Filtering consists in translating $[\lambda_{m+1}, \lambda_N]$ (the part of the spectrum we don't want to amplify) onto $[-1, 1]$ and then applying a Chebyshev polynomial :

$$f = T_n\left(\frac{1}{r}(H - cI)\right) \text{ with } c = \frac{\lambda_{m+1} + \lambda_N}{2},\ r = \frac{\lambda_N - \lambda_{m+1}}{2}$$

ChebFi in the Self-Consistent Field

Algorithm Chebyshev filtering in the SCF cycle

$X \leftarrow \text{RandomVectors}(N, m)$
$H \leftarrow \text{Hamiltonian}(X)$
while H not converged **do**
  $c \leftarrow \frac{1}{2}(E_{\text{cut}} + \lambda_m)$, $c \leftarrow \frac{1}{2}(E_{\text{cut}} - \lambda_m)$
  $X \leftarrow T_n(\frac{1}{r}(H - cI))(X)$
  $\lambda, X \leftarrow \text{RayleighRitz}(H, X)$
  $H \leftarrow \text{Hamiltonian}(X)$
end while

Notes on Chebyshev Filtering

- Convergence rate is linked to the polynomial degree, which is the number of time the Hamiltonian operator H needs to be applied to a vector (usually between 2 and 10, Abinit default value is 4).
- Significant possibilities of parallelization : The filter can be applied in parallel to each vector.
- One Rayleigh-Ritz in dimension m per SCF step.
- The last bands converge more slowly so we need to compute more bands than necessary.
- No preconditioning is possible.

Using ChebFi in Abinit

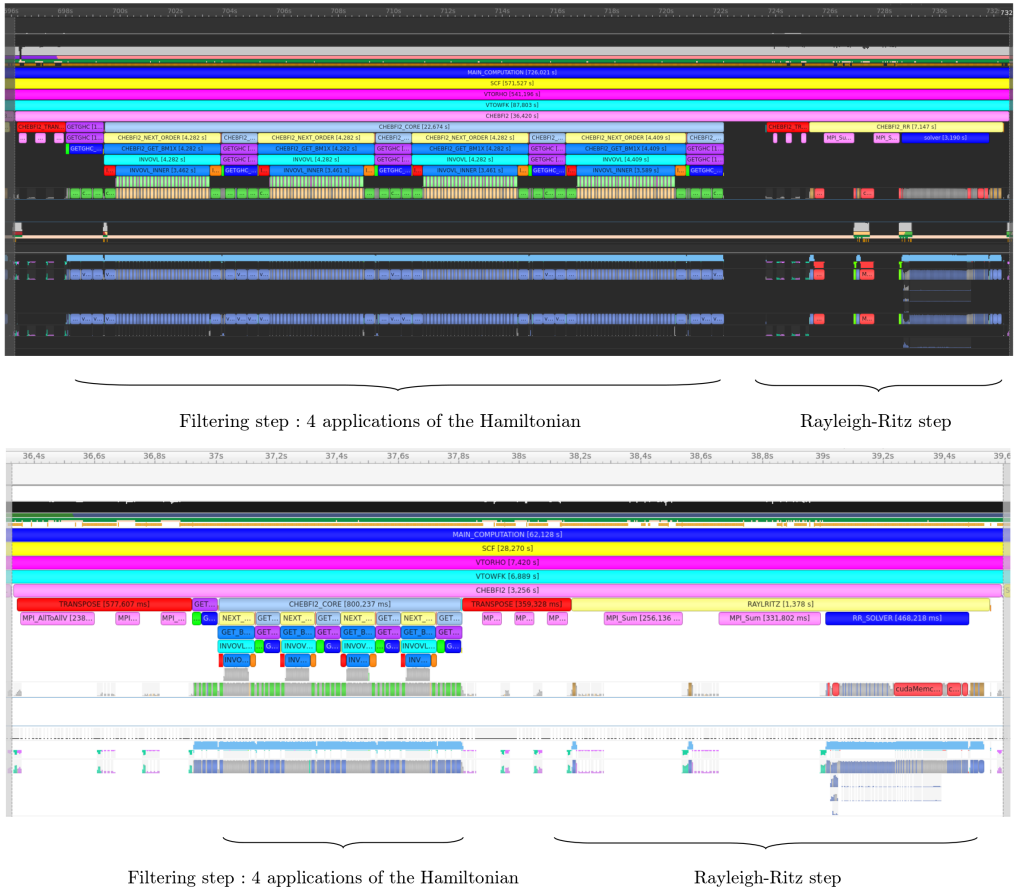
Abinit keywords :

- `wfoptalg` : set to 1 to select ChebFi.
- `npband` : number of processor used to parallelize over bands.
- `nline` : degree of the Chebyshev polynomial (should be between 2 and 10).

Comparison of the different algorithms

	LOBPCG	ChebFi
Cost per SCF step	$\mathcal{O}(n_{\text{block}} \cdot n_{\text{line}} \cdot (T_H \cdot m_b + (3m_b)^3) + m^3)$	$\mathcal{O}(n_{\text{deg}} \cdot T_H \cdot m + m^3)$
Advantages	• Fast convergence thanks to preconditioning. • Stable and well understood.	• Good scalability.
Drawbacks	• Fewer parallelization possibilities. • Many calls to Rayleigh-Ritz that scales very poorly	• No preconditioning possible. • Last bands are poorly converged.

Outlook



(a) CPU

(b) CPU + GPU

Figure – Time spent in the filtering step and in the Rayleigh-Ritz step on CPU computers and CPU+GPU computers.

Thanks for listening !