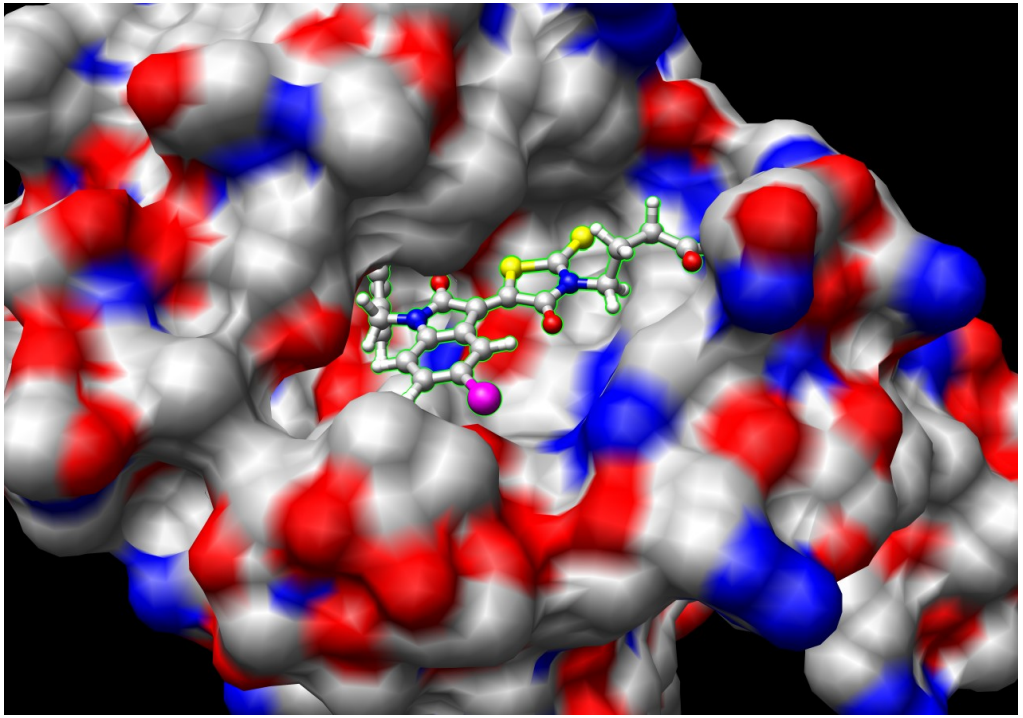
Diagonalisation parallèle pour le calcul de structure électronique

Antoine Levitt

Laboratoire Jacques-Louis Lions, UPMC
Collaboration avec Marc Torrent, CEA/DAM Bruyères-Le-Châtel
doi:10.1016/j.cpc.2014.10.015

Mercredi 5 mai 2015
Séminaire de l'équipe PEQUAN, LIP6, UPMC

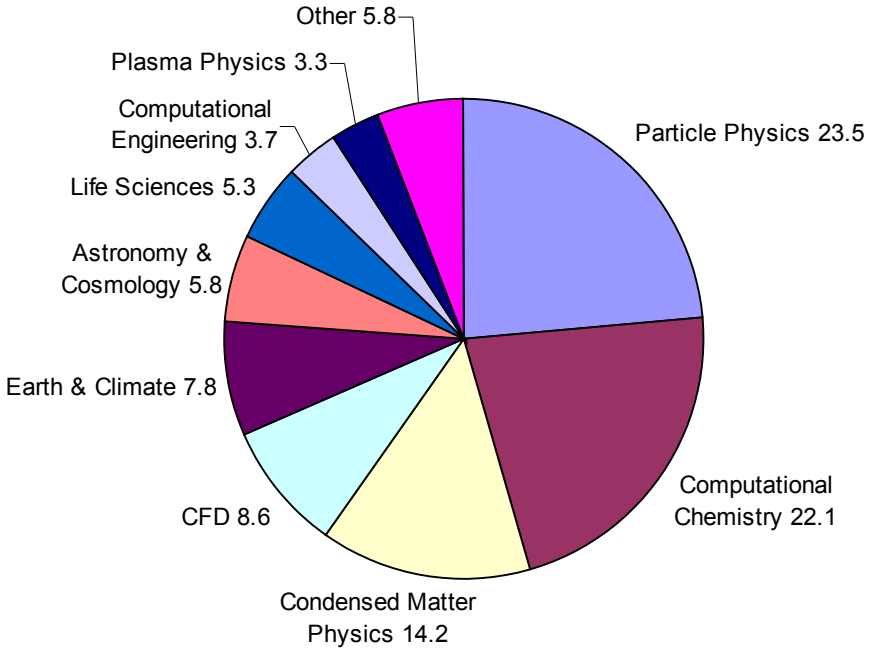
Applications



- Drug design (docking)
- Materials research
  - Crystalline structures
  - Elastic constants, dislocations
  - Spectroscopy
  - Thermal, electrical conduction

Importance of electronic structure computations

Distribution of LEFs by scientific area

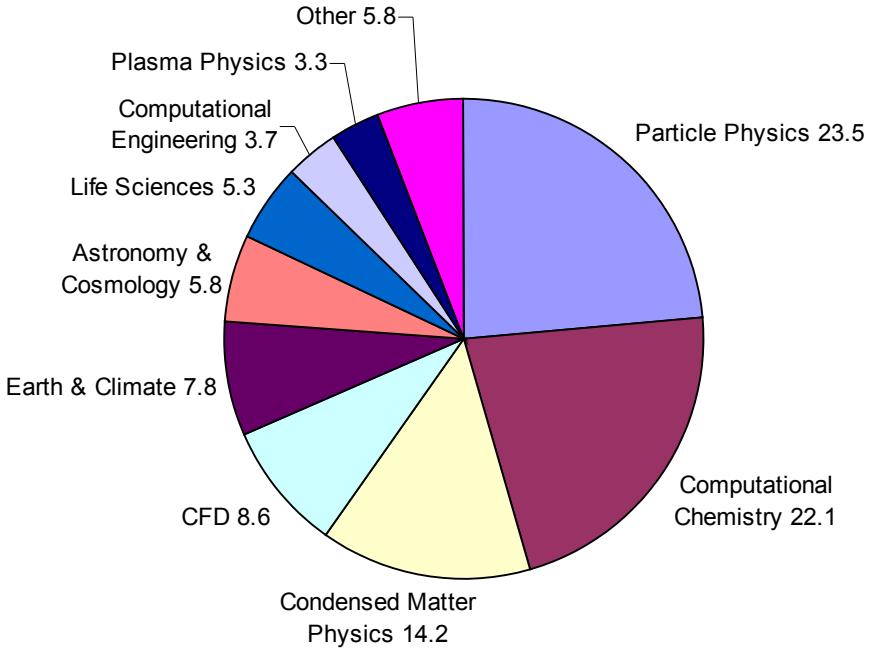


EPSRC Engineering and Physical Sciences Research Council

epcc

Importance of electronic structure computations

Distribution of LEFs by scientific area



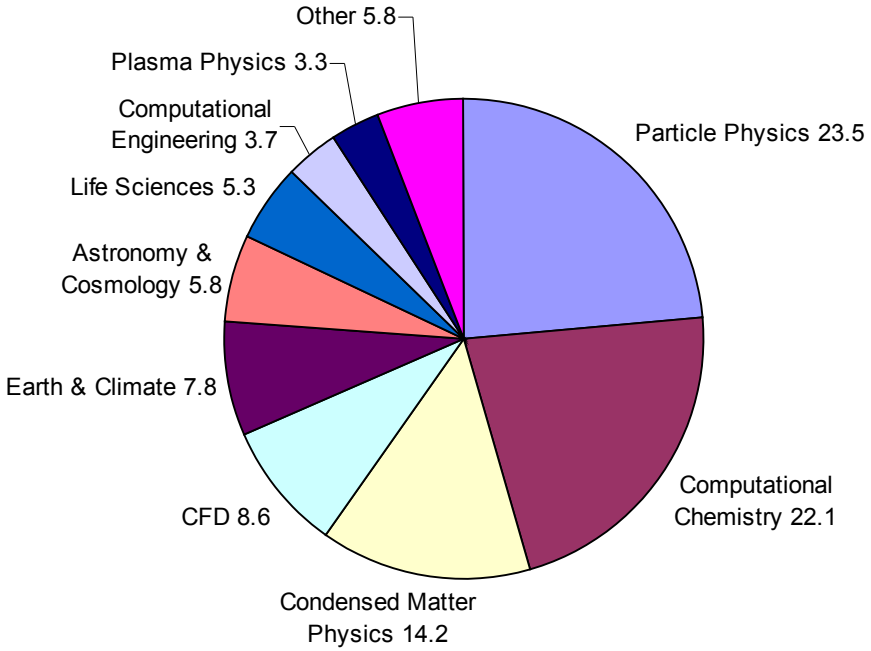
EPSRC Engineering and Physical Sciences Research Council

epcc

Keyword	Google Scholar	MathSciNet
Navier Stokes	537,000	8,220
Density Functional Theory		

Importance of electronic structure computations

Distribution of LEFs by scientific area



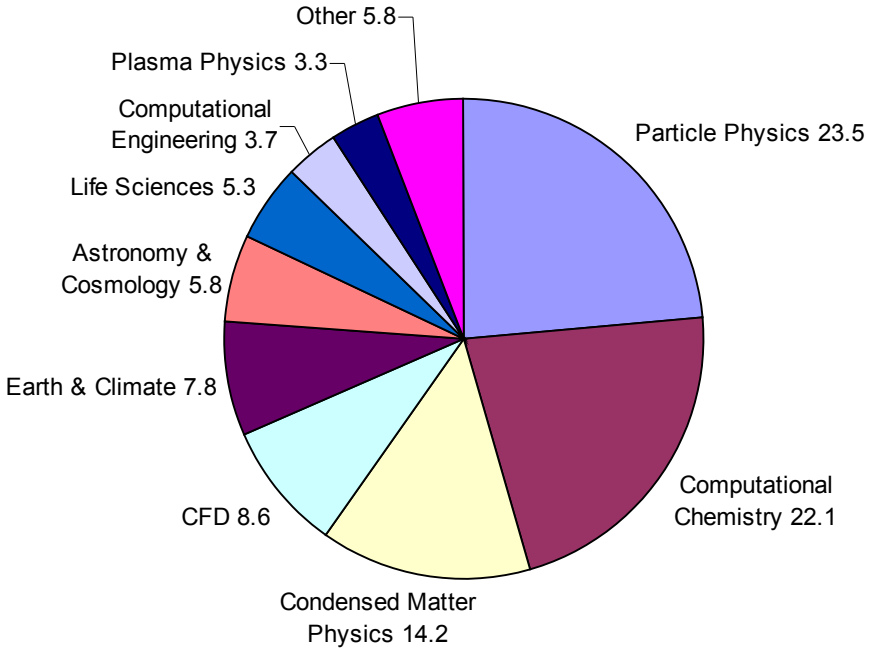
EPSRC Engineering and Physical Sciences Research Council

epcc

Keyword	Google Scholar	MathSciNet
Navier Stokes	537,000	8,220
Density Functional Theory	1,200,000	

Importance of electronic structure computations

Distribution of LEFs by scientific area



EPSRC Engineering and Physical Sciences Research Council

epcc

Keyword	Google Scholar	MathSciNet
Navier Stokes	537,000	8,220
Density Functional Theory	1,200,000	124

Mathematicians and electronic structure

- Collaboration between chemists/physicists and mathematicians is important
  1. For us (need to understand relevant current models)
  2. For them (numerical methods, HPC)
- Strong collaboration between mathematicians and chemists at UPMC (people : E. Cancès, L. Lagardère, F. Lipparini, Y. Maday, B. Menucci, J-P. Piquemal, B. Stamm)
  1. 2013 : Domain decomposition for solvation models (solve subproblems analytically, parallelization)
  2. 2014 : Polarizable force fields (use CG instead of Jacobi, parallelization)
  3. ...

This talk:
Joint work with Marc Torrent, condensed matter lab, CEA/DAM

Summary

Quantum mechanics in one slide

- Matter at the atomic level is described by quantum mechanics
- Main equation : time-independent Schrödinger equation

$$H\psi = E\psi$$

- Eigenvalue equation for the self-adjoint operator H on a Hilbert space $\mathcal{H}$
- $\psi \in \mathcal{H}$ is the complex wavefunction
- Entanglement: if subsystems A and B are represented by $\mathcal{H}_A$ and $\mathcal{H}_B$, then $A \cup B$ is represented by $\mathcal{H}_A \otimes \mathcal{H}_B$ (contrast with classical $\mathcal{H}_A \oplus \mathcal{H}_B$)
- For N electrons, $\mathcal{H} \subset L^2(\mathbb{R}^{3N})$ (compare with classical $\mathcal{H} \subset \mathbb{R}^{6N}$)
- If using 10 d.o.f. per electron, $\mathcal{H} \subset R^{10^{3N}}$: much too big!

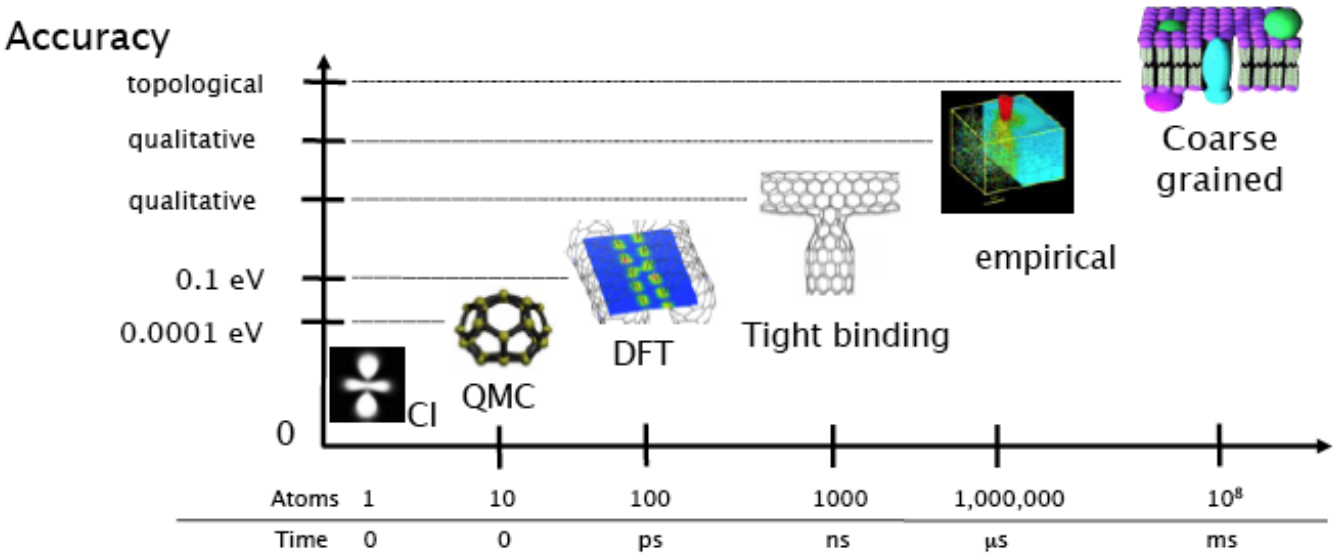
The Schrödinger equation

N electrons in a potential V. H is the total energy (kinetic + electron-nuclei + electron-electron)

$$(H\psi)(x) = \sum_{i=1}^{N} \left(\left(-\frac{1}{2}\Delta_i \psi \right)(x) + V(x_i)\psi(x) \right) + \sum_{1 \le i < j \le N} \frac{1}{|x_i - x_j|} \psi(x),$$

with $x \in \mathbb{R}^{3N} = (x_i)_{1 \le i \le N}$.

- The electron-electron term couples (*entangles*) the N electrons
- Direct simulation impossible ($3N$-dimensional PDE)



The Kohn-Sham equations

- Approximate the Schrödinger equation by a system of N non-interacting electrons satisfying

$$-\frac{1}{2}\Delta\psi_i + V_{\text{eff}}[\rho]\psi_i = \lambda_i\psi_i$$

- Effective potential

$$V_{\text{eff}}[\rho] = V_{\text{ext}} + V_{\text{Hartree}}[\rho] + V_{XC}[\rho]$$

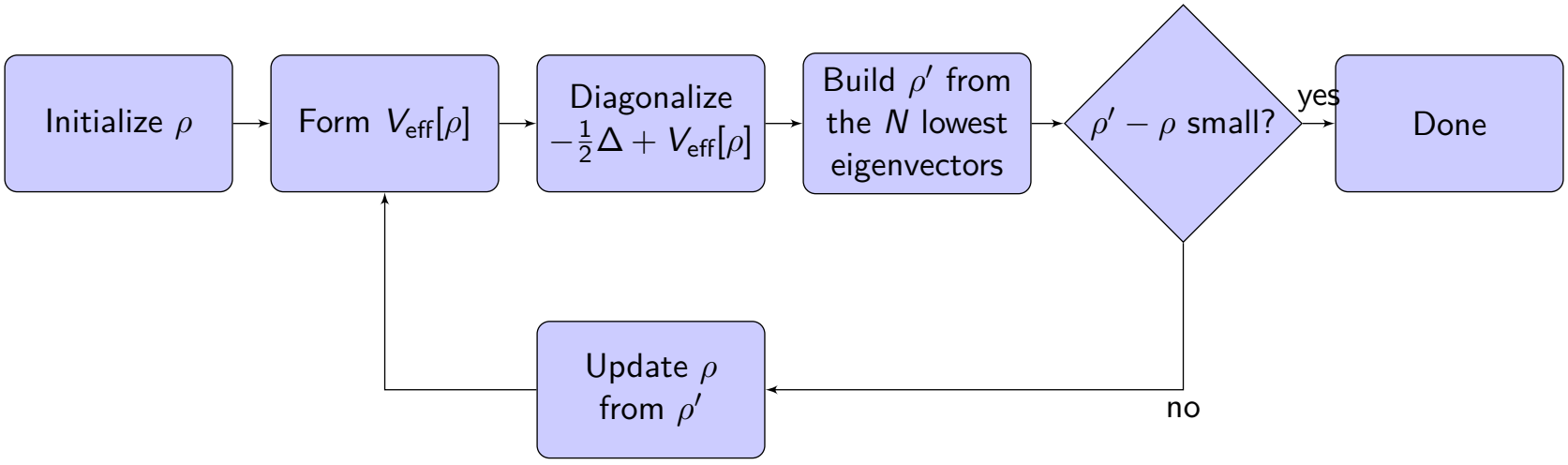
depending only on the electronic density $\rho(y) = \sum_{i=1}^{N} |\psi_i|^2(y)$

- V_{ext} is the electron-nuclei interaction potential
- Mean-field potential

$$V_{\text{Hartree}}[\rho](x) = \int_{y\in\mathbb{R}^3} \frac{\rho(y)}{|x-y|}$$

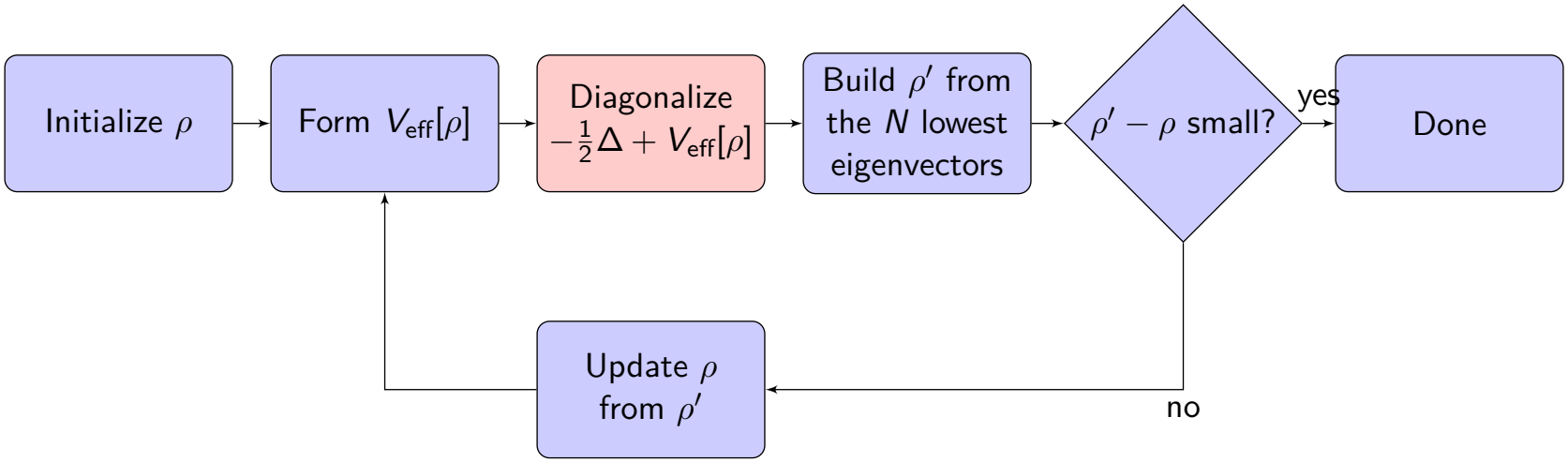
- Exchange-correlation potential $V_{XC}[\rho](x)$ approximated using various schemes (LDA, GGA ...) from the density ρ

The Self-Consistent Field (SCF) cycle



- Many important complications not discussed here
  1. Types of systems: crystals, metals ...
  2. More accurate physics: spin, XC functional, relativistic effects, perturbation theory ...
  3. Implementation: discretization, pseudopotentials ...
  4. Derived properties: geometry optimization, molecular dynamics, optical and mechanical constants, excited states ...

The Self-Consistent Field (SCF) cycle



- Many important complications not discussed here
  1. Types of systems: crystals, metals ...
  2. More accurate physics: spin, XC functional, relativistic effects, perturbation theory ...
  3. Implementation: discretization, pseudopotentials ...
  4. Derived properties: geometry optimization, molecular dynamics, optical and mechanical constants, excited states ...

Computational bottleneck: get the N lowest eigenvectors (*bands*) of $-\frac{1}{2}\Delta + V_{\text{eff}}[\rho]$

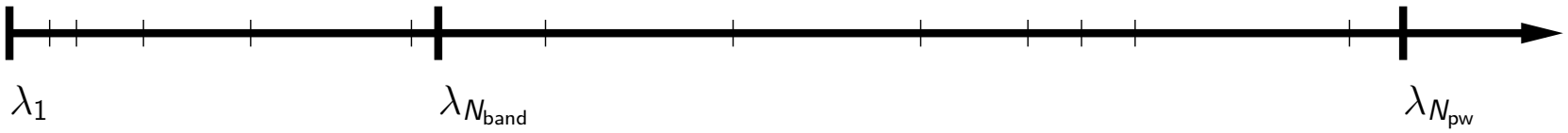
Summary

Overview of Abinit

"Finds the total energy, charge density, and electronic structure of systems made of electrons and nuclei, using pseudopotentials and a plane-wave basis."

- Solves the Kohn-Sham equations discretized in a plane-wave basis (spectral method)
- Represent core electrons (chemically inert) by an effective pseudopotential
- International collaboration led by Université Catholique de Louvain, Belgium, with an important group at CEA/DAM
- Free software (GNU GPL), `http://www.abinit.org`
- Widely used in condensed matter physics, lots of features
- HPC in Abinit: parallel FFT (Goedecker, Boulet, Deutsch, 2003), block eigenvalue solvers (Bottin, Leroux, Knyazev, Zerah, 2006), OpenMP/GPU (ongoing)

The eigenproblem in Abinit



Innermost loop : at ρ fixed, find the N_{band} lowest eigenvectors of the Galerkin projection of

$$H = -\frac{1}{2}\Delta + V_{\text{eff}}[\rho].$$

on

$$\mathcal{H}_{N_{\text{pw}}} = \text{Span}\{e^{i\xi_n \cdot x}\}_{1 \leq n \leq N_{\text{pw}}}$$

- Target medium-size applications : $N_{\text{band}} \approx 1,000$, $N_{\text{pw}} \approx 100,000$
- H too big and dense: don't compute H explicitly, use its action on vectors

Hamiltonian operator

The Hamiltonian

$$H = -\frac{1}{2}\Delta + V_{\text{loc}}[\rho] + V_{\text{nonloc}}[\rho]$$

- The Laplacian is diagonal in our plane-wave basis
- $V_{\text{loc}}[\rho]$ is a real-space multiplication : computed with FFTs
- Nonlocal terms come from pseudopotentials (not discussed in details here), and involve a $N_{\text{pw}} \times N_{\text{projs}}$ set of *projectors* P, $N_{\text{projs}} \approx N_{\text{band}}$
- $V_{\text{nonloc}}[\rho] = PV_{\text{proj}}[\rho]P^T$
- Cost of applying H: $O(N_{\text{pw}} \log N_{\text{pw}} + N_{\text{pw}} N_{\text{projs}})$
- FFT (even 3D) is hard to parallelize beyond 100 cores

Hamiltonian operator

The Hamiltonian

$$H = -\frac{1}{2}\Delta + V_{\text{loc}}[\rho] + V_{\text{nonloc}}[\rho]$$

- The Laplacian is diagonal in our plane-wave basis
- $V_{\text{loc}}[\rho]$ is a real-space multiplication : computed with FFTs
- Nonlocal terms come from pseudopotentials (not discussed in details here), and involve a $N_{\text{pw}} \times N_{\text{projs}}$ set of *projectors* P, $N_{\text{projs}} \approx N_{\text{band}}$
- $V_{\text{nonloc}}[\rho] = PV_{\text{proj}}[\rho]P^T$
- Cost of applying H: $O(N_{\text{pw}} \log N_{\text{pw}} + N_{\text{pw}} N_{\text{projs}})$
- FFT (even 3D) is hard to parallelize beyond 100 cores

Have to find other levels of parallelism (matvecs in parallel)

What is the ideal eigensolver?

- We can compute matvecs efficiently but the Hamiltonian is not sparse: need "**matrix-free**" algorithms
- The linear solver is not an island, but is embedded in multiple loops (k-points, SCF, molecular dynamics, geometry optimization ...) : **reuse** available information (approximate eigenvectors)
- Must be able to **stop early**: there's no point in optimizing the wavefunctions to 10^{-16} if the potential is not converged
- $-\frac{1}{2}\Delta + V$ well approximated by $-\frac{1}{2}\Delta$ in the high-frequency regime : cheap and efficient **preconditionner** available
- Eigenvector-level parallelism: prefer **fixed basis** than growing-basis algorithms, allow concurrent update of approximate eigenvectors, limit communications

Rules out "standard" algorithms (forget Lanczos)

Rayleigh-Ritz(-Galerkin)

- Variational formulation for $H\psi = \lambda\psi$, H symmetric

$$\psi_n = \underset{\psi\in\mathcal{H}, \langle\psi_i,\psi\rangle=\delta_{i,n},\ i=1,\ldots,n}{\arg\min} \langle\psi, H\psi\rangle$$

- Important tool for eigenvalue problems: Rayleigh-Ritz ("variational principle", "subspace rotation", "block diagonalization" ...), using a set of trial vectors $(\tilde{\psi}_i)_{1\leq i\leq N_t}$:

$$\psi_n = \underset{\substack{\psi\in\mathrm{Span}\{(\tilde{\psi}_i)_{1\leq i\leq N_t}\},\\ \langle\psi_i,\psi\rangle=\delta_{i,n},\ i=1,\ldots,n}}{\arg\min} \langle\psi, H\psi\rangle$$

- Solve the eigenvalue problem in $\mathrm{Span}\{(\tilde{\psi}_i)_{1\leq i\leq N_t}\}$
  1: Form $\tilde{H} = \left\langle \tilde{\psi}_i, H\tilde{\psi}_j \right\rangle$, $\tilde{S} = \left\langle \tilde{\psi}_i, \tilde{\psi}_j \right\rangle$
  2: Solve $\tilde{H}X_i = \lambda_i \tilde{S}X_i$ for the lowest eigenvalues
  3: $\psi_i = \sum_{j=1}^{N_t} \tilde{\psi}_j X_{ji}$
- Implementation: manual distributed matrix-multiply + ScaLAPACK

Historic eigensolver : conjugate gradient

- Tetter, Allan, Payne ('89) : "band-by-band conjugate gradient"
- Idea : minimize the Rayleigh quotient $\langle \psi_i, H\psi_i \rangle$ subject to $\langle \psi_i, \psi_j \rangle = \delta_{ij}$, $j \leq i$
- Minimise Rayleigh quotient by conjugate gradients, while maintaining orthogonality by Gram-Schmidt
- Precondition conjugate gradients by diagonal matrix (scale high frequencies by $1/|\xi|^2$)
- Implemented in most plane-wave DFT codes
- ☺ Good convergence, robust, easy to implement, good support for *locking* (not iterating on converged vectors)
- ☹ Many orthogonalizations, not parallel

Historic eigensolver : conjugate gradient

- Tetter, Allan, Payne ('89) : "band-by-band conjugate gradient"
- Idea : minimize the Rayleigh quotient $\langle \psi_i, H\psi_i \rangle$ subject to $\langle \psi_i, \psi_j \rangle = \delta_{ij}$, $j \leq i$
- Minimise Rayleigh quotient by conjugate gradients, while maintaining orthogonality by Gram-Schmidt
- Precondition conjugate gradients by diagonal matrix (scale high frequencies by $1/|\xi|^2$)
- Implemented in most plane-wave DFT codes
- ☺ Good convergence, robust, easy to implement, good support for *locking* (not iterating on converged vectors)
- ☹ Many orthogonalizations, not parallel

How do we run this on 100k cores?

Block algorithms : LOBPCG

- **L**ocally **O**ptimal **B**lock **P**reconditionned **C**onjugate **G**radient (Knyazev '01)
- Iteratively update all the eigenvectors at the same time
- Use a single Rayleigh-Ritz procedure on the $3N_{\text{band}}$-dimensional space $\{\psi_i^{n-1}, \psi_i^n, P^{-1}H\psi_i^n\}_{i=1,\dots,N_{\text{band}}}$
- ☺ Better convergence than CG, can compute $H\psi_i^n$ in parallel!
- ☹ Higher costs for the Rayleigh-Ritz procedure
- Split in blocks to avoid the Rayleigh-Ritz costs
- Acceptable parallel scaling up to 500-1k cores for large problems
- 1k cores is still not enough (target computers at CEA/DAM : 100k cores, much more in the future)

Block algorithms : LOBPCG

- **L**ocally **O**ptimal **B**lock **P**reconditionned **C**onjugate **G**radient (Knyazev '01)
- Iteratively update all the eigenvectors at the same time
- Use a single Rayleigh-Ritz procedure on the $3N_{\text{band}}$-dimensional space $\{\psi_i^{n-1}, \psi_i^n, P^{-1}H\psi_i^n\}_{i=1,\dots,N_{\text{band}}}$
- ☺ Better convergence than CG, can compute $H\psi_i^n$ in parallel!
- ☹ Higher costs for the Rayleigh-Ritz procedure
- Split in blocks to avoid the Rayleigh-Ritz costs
- Acceptable parallel scaling up to 500-1k cores for large problems
- 1k cores is still not enough (target computers at CEA/DAM : 100k cores, much more in the future)

Can we do better ?

Summary

Filtering algorithm

- Goal: avoid frequent communications (orthogonalizations, Rayleigh-Ritz)
- Some global communication between eigenvectors is unavoidable (otherwise it's an interior eigenvalue problem - danger)

```
1: while not converged do
2:     for each band i do
3:         ψ_i ←???
4:     end for
5:     Apply the Rayleigh-Ritz procedure to the ψ_i
6: end while
```

- The Rayleigh-Ritz step liberates us from the need to converge every eigenvector separately
- We only need to filter out the unwanted eigencomponents

Filtering algorithm

- Goal: avoid frequent communications (orthogonalizations, Rayleigh-Ritz)
- Some global communication between eigenvectors is unavoidable (otherwise it's an interior eigenvalue problem - danger)

```
1: while not converged do
2:     for each band i do
3:         ψ_i ← L(ψ_i)
4:     end for
5:     Apply the Rayleigh-Ritz procedure to the ψ_i
6: end while
```

- The Rayleigh-Ritz step liberates us from the need to converge every eigenvector separately
- We only need to filter out the unwanted eigencomponents

Filtering algorithms

Choice of L

- L must approximate the spectral projector on the first N_{band} eigenvectors
- If $H = Q\Lambda Q^T$, then $f(H) = Qf(\Lambda)Q^T$: find a computable matrix function f that approximates $\chi_{[\lambda_1,\lambda_N]}$
- Computable functions of matrices? Polynomials
- Hard to approximate a discontinuous function with polynomials. Rational functions better but require inversion (too expensive in our case) : focus on polynomials
- p must be large on $[\lambda_1, \lambda_{N_{\text{band}}}]$, small on $[\lambda_{N_{\text{band}}+1}, \lambda_{N_{\text{pw}}}]$

Choice of L

- L must approximate the spectral projector on the first N_{band} eigenvectors
- If $H = Q\Lambda Q^T$, then $f(H) = Qf(\Lambda)Q^T$: find a computable matrix function f that approximates $\chi_{[\lambda_1,\lambda_N]}$
- Computable functions of matrices? Polynomials
- Hard to approximate a discontinuous function with polynomials. Rational functions better but require inversion (too expensive in our case) : focus on polynomials
- p must be large on $[\lambda_1, \lambda_{N_{\text{band}}}]$, small on $[\lambda_{N_{\text{band}}+1}, \lambda_{N_{\text{pw}}}]$

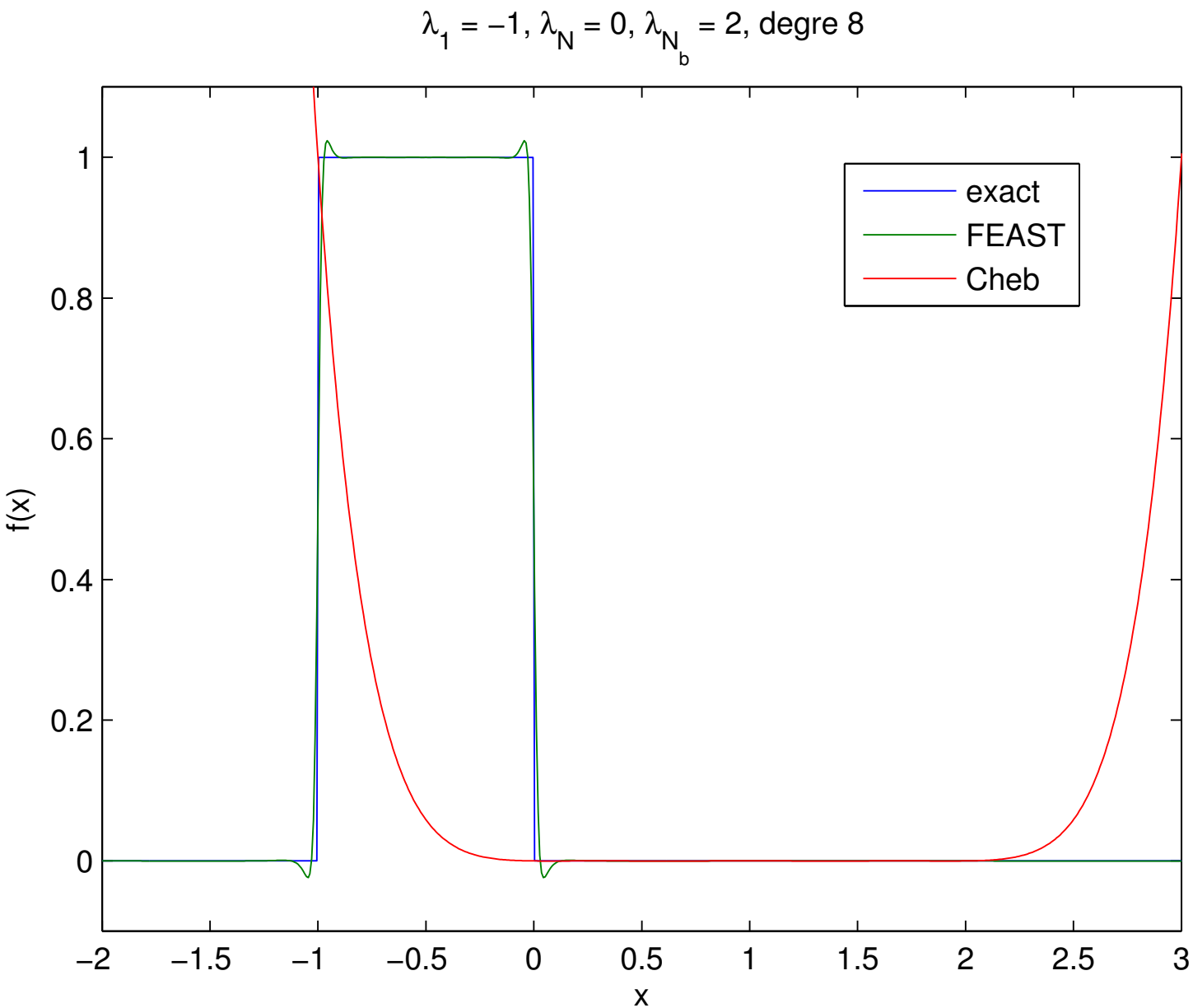
What is the smallest polynomial of a given degree on $[\lambda_{N_{\text{band}}+1}, \lambda_{N_{\text{pw}}}]$?

Choice of L

- L must approximate the spectral projector on the first N_{band} eigenvectors
- If $H = Q\Lambda Q^T$, then $f(H) = Qf(\Lambda)Q^T$: find a computable matrix function f that approximates $\chi_{[\lambda_1,\lambda_N]}$
- Computable functions of matrices? Polynomials
- Hard to approximate a discontinuous function with polynomials. Rational functions better but require inversion (too expensive in our case) : focus on polynomials
- p must be large on $[\lambda_1, \lambda_{N_{\text{band}}}]$, small on $[\lambda_{N_{\text{band}}+1}, \lambda_{N_{\text{pw}}}]$

What is the smallest polynomial of a given degree on $[\lambda_{N_{\text{band}}+1}, \lambda_{N_{\text{pw}}}]$?
Chebyshev polynomials

Choice of L


λ1 = −1, λN = 0, λNb = 2, degre 8
exact
FEAST
Cheb
f(x)
x
1
0.8
0.6
0.4
0.2
0
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2
2.5
3

Chebyshev filtering

- Old ideas (Rutishauser, 1969), applied to DFT by Zhou, Saad, Tiago, Chelikowsky (2006), gaining popularity in DFT

1: **while** not converged **do**
2: **for** each band i **do**
3: $\psi_i \leftarrow T_n(\psi_i)$
4: **end for**
5: Apply the Rayleigh-Ritz procedure to the ψ_i
6: **end while**

where T_n is the Chebyshev polynomial on $[\lambda_{N_{\text{band}}+1}, \lambda_{N_{\text{pw}}}]$.

- Need approximation of $\lambda_{N_{\text{band}}+1}, \lambda_{N_{\text{pw}}}$, easily done
- Very good parallel properties (much less Rayleigh-Ritz than in LOBPCG)

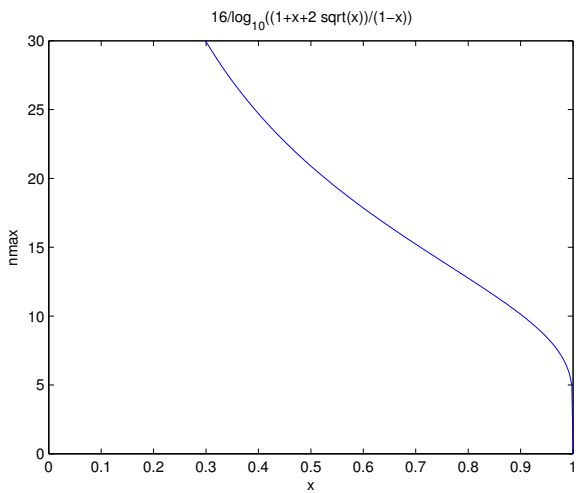
Choice of degree

- If n is too small, too many communications
- If n is too large, every ψ_i converges to the first eigenvector!
- How large is too large? Amplification factor:

$$T_n(\lambda_1) \sim_{n \gg 1} \left(\frac{1+\sqrt{f}}{1-\sqrt{f}} \right)^n ,$$

$$f = \frac{\lambda_N - \lambda_1}{\lambda_{N_b} - \lambda_1} \text{ fraction of the spectrum to compute.}$$

- Heuristic: $T_n(\lambda_1) \ll 10^{16}$



Not an issue in practice

Convergence analysis

- Remember convergence theory for CG

$$
\begin{aligned}
\|e_n\|_A &= \min_{e \in e_0 + K_n(Ae_0)} \|e\|_A \\
&= \min_{P \in P_n, P(0)=1} \max_{\lambda \in \Lambda(A)} P(\lambda) \|e_0\|_A \\
&\leq \max_{\lambda \in \Lambda(A)} T_n(\lambda) \|e_0\|_A \\
&\leq \left(2 \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^n \|e_0\|_A ,
\end{aligned}
$$

  where T_n is a suitably rescaled Chebyshev polynomial
- Same applies for LOBPCG: Chebyshev acts as LOBPCG's worst case
- Crucial difference: LOBPCG uses preconditioning, Chebyshev cannot

Convergence analysis

- Remember convergence theory for CG

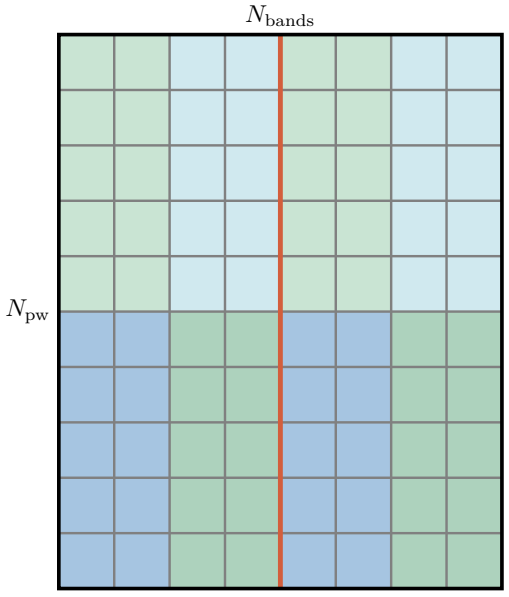
$$
\begin{aligned}
\|e_n\|_A &= \min_{e \in e_0 + K_n(Ae_0)} \|e\|_A \\
&= \min_{P \in P_n, P(0)=1} \max_{\lambda \in \Lambda(A)} P(\lambda) \|e_0\|_A \\
&\leq \max_{\lambda \in \Lambda(A)} T_n(\lambda) \|e_0\|_A \\
&\leq \left(2 \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^n \|e_0\|_A ,
\end{aligned}
$$

  where T_n is a suitably rescaled Chebyshev polynomial
- Same applies for LOBPCG: Chebyshev acts as LOBPCG's worst case
- Crucial difference: LOBPCG uses preconditioning, Chebyshev cannot

Is the additional scalability worth it? Only one way to find out ...

Summary

Parallelism



- 2D MPI grid
- Vertical communication for Hamiltonian application, horizontal communication for Rayleigh-Ritz
- Load balancing problem (extremal eigenvectors converge faster): use cyclic distribution
- Scalability essentially limited by the Rayleigh-Ritz step (diagonalizing a 1000×1000 dense matrix stops scaling around 100 processors, even with state-of-the-art libraries)

Wonders of BLAS

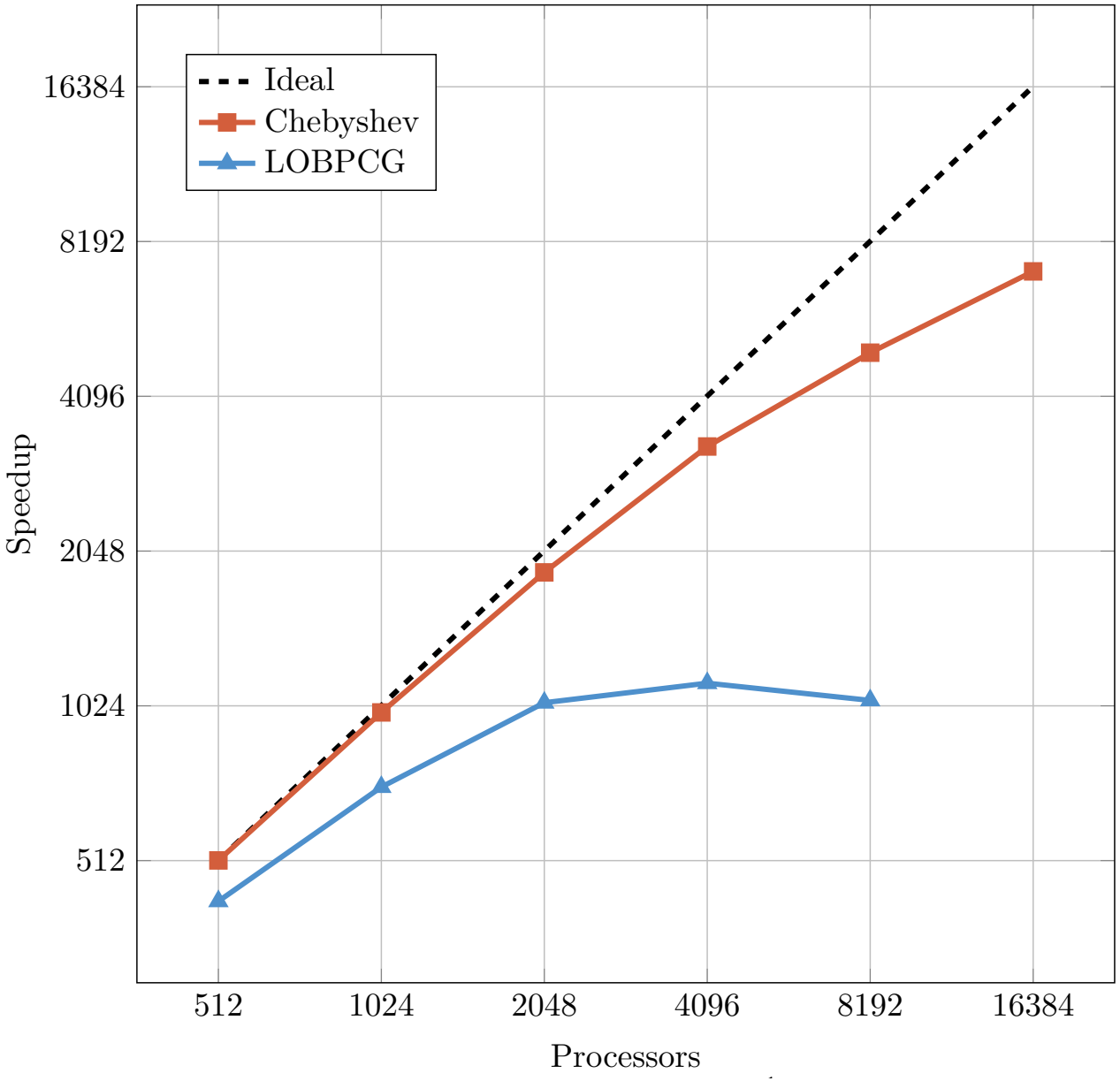
- Important part of the computation: operations like $P^T \Psi$, with P a $N_{\text{pw}} \times N_{\text{projs}}$ matrix of projectors, and Ψ a $N_{\text{pw}} \times N_{\text{band}}$ matrix of wavefunctions
- Previous implementation in Abinit:

```
1: for iband=1,nband do
2:    for ia=1,natom do
3:       for ilmn=1,nlmn do
4:          for ipw=1,npw do
5:             ...
6:          end for
7:       end for
8:    end for
9: end for
```

- Replaced by BLAS3: up to x5 speedup (but more memory)
- Physicists still have a "count the FLOPS" culture
- Progress = physicists + mathematicians + HPC experts working together

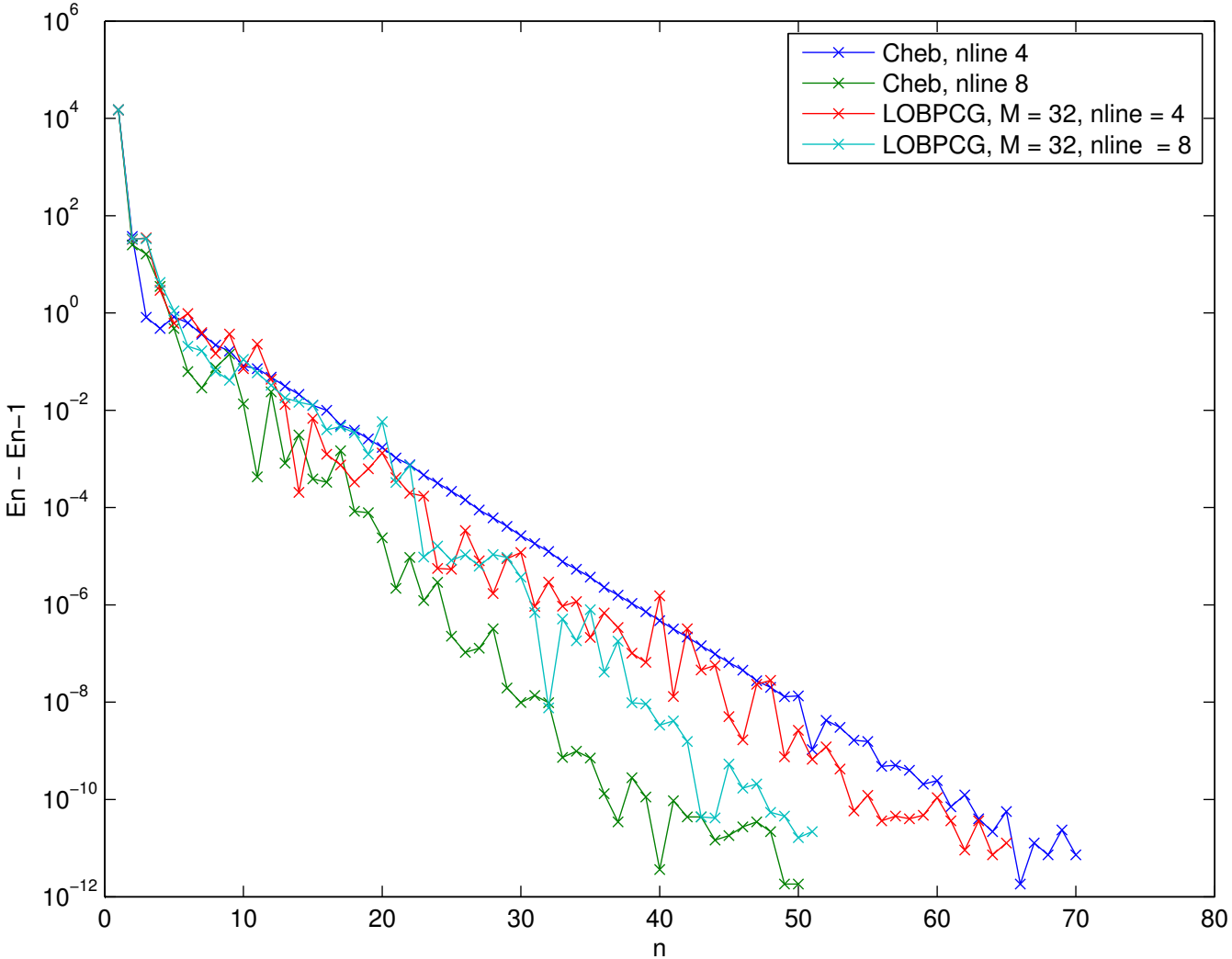
Strong scaling

Speedup for 512 atoms of Ti, Curie. 4k bands, 170k plane waves



SCF Convergence

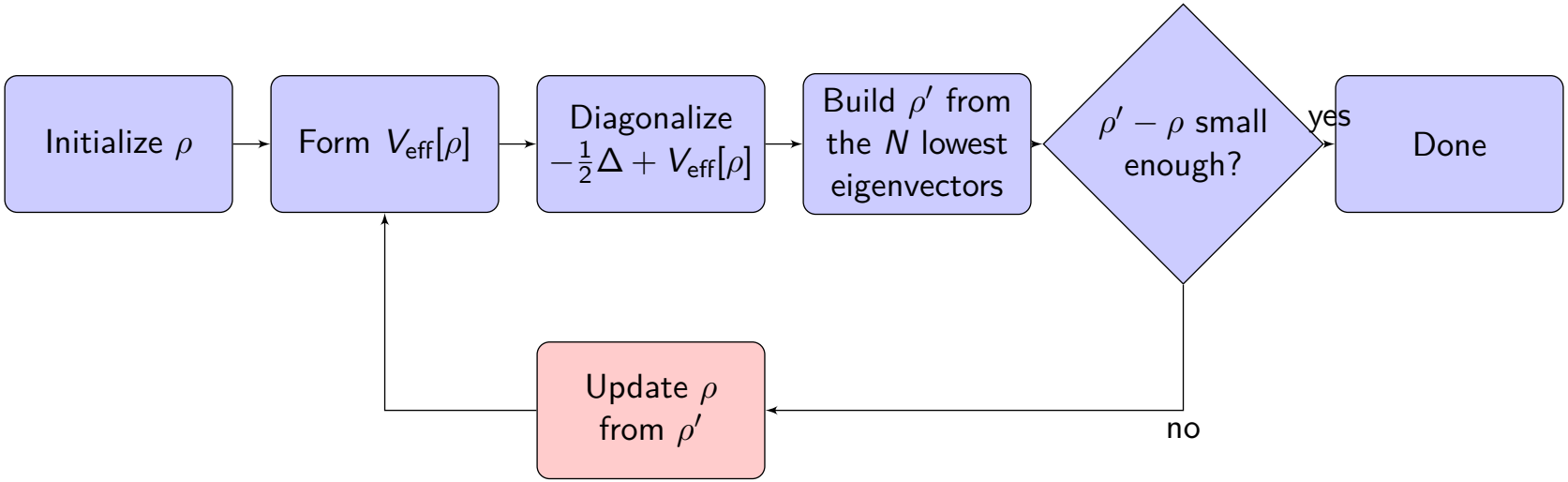
SCF (nonlinear) convergence, 256 Titanium atoms



Chebyshev sensibly identical to LOBPCG. Choice of nline is not trivial

Summary

Convergence acceleration



- Task: find the fixed point of the mapping $\rho \to \rho'$
- General setting: find a fixed point of $g(x)$, ie a solution of $f(x) = g(x) - x = 0$
- We can compute g at a cost of $O(N^3)$, but not its derivatives: no Newton
- Can we do better than $x_{n+1} = g(x_n)$?

Anderson acceleration

- Anderson, '65 :

$$x_{n+1} = \sum_{i=n-m}^{n} \alpha_i g(x_i),$$

  with $\sum_{i=n-m}^{n} \alpha_i = 1$
- α chosen to minimize the linearized residual of $\sum_{i=n-m}^{n} \alpha_i x_i$:

$$\alpha = \underset{\alpha \in \mathbb{R}^{m+1}, \sum_i \alpha_i = 1}{\arg\min} \left\| \sum_{i=n-m}^{n} \alpha_i (g(x_i) - x_i) \right\|$$

- Requires the solution of a $N \times m$ linear least square problem, can be implemented efficiently (incremental QR factorisation, each step is $O(Nm)$, usually negligible compared to the cost of computing g)
- Usually m fixed, around 10 (keep 10 last iterations)
- Used systematically in the electronic structure community, but not much use outside

Numerical analysis

- Known as DIIS (Pulay, '82) among chemists
- Related but not equivalent to convergence acceleration methods studied by numerical analysts (Reduced-Rank Extrapolation, Minimal Polynomial Extrapolation ...). Which is better?
- Studied only recently by mathematicians
  1. Fang-Saad '08: equivalent to a multisecant method
  2. Walker-Ni '11: equivalent to GMRES in the linear case when $m = \infty$
  3. Toth-Kelley '13: convergence in the nonlinear case, when g is a contraction and the α remain bounded

Ongoing work (joint with B. Stamm and Y. Maday)

- Empirically, even when the convergence is linear, the convergence rate depends on initial conditions: analysis complicated, similar to restarted GMRES
- In the linear case $g(x) = Ax + b$, convergence no worse than $||A||^n$ (compare with $\rho(A)^n$ for the fixed point iteration)
- Explicit examples for symmetric 2D matrices with $m = 1$ where $\rho(A)^n$ is sharp $\rightarrow$ no acceleration
- Explicit examples for non-symmetric 2D matrices with $m = 1$ where $||A||^n$ is sharp $\rightarrow$ worse than fixed point!
- Work in progress

Summary

The Future Of Plane-Wave DFT

- Present capabilities: 10k electrons, 0.1 Petaflops, 10k procs
- If we want to do 100k electrons (and we do), need 0.1 exaflop, 10m procs (in a supercomputer near you by 2020-2025?)
- Towards exascale: compute individual parts of spectrum in parallel?
- Can we solve an interior eigenvalue problem without inversions?
- Use "windowing" polynomials, but need very high degree (Schofield, Chelikowsky, Saad, 2011)
- Use iterative methods to invert shifted systems? Would need a better preconditionner.

Conclusion: mathematical open problems

- Analysis of CG? LOBPCG?
- Rigorous numerical analysis of the stability of ChebFi
- Optimal basis size M (cost model to balance increased cost and accelerated convergence)
- Optimal degree?
- What is the polynomial/rational/Padé filter that optimizes the convergence rate?
- Is there a way incorporate preconditioning in non-optimisation based eigenvalue solvers?
- Conventional wisdom in numerical analysis: Krylov good, Chebyshev bad. But Krylov methods need orthogonality = global communications. Time to dust off old textbooks?

Opportunities for math/chemistry/CS collaborations

- Software engineering
  - Accuracy, stability? (CADNA?)
  - Reproducibility?
  - Automatic differentiation?
- High performance computing
  - Dense linear algebra
  - Heterogeneous computing units
- What to do with simulations output?
  - Visualization
  - Fitting
- Machine learning: learn the function that maps atomic configuration to energies?

Thank you!