Large scale electronic structure calculations with the LOBPCG method

François Bottin[1], Stéphane Leroux[1], Andrew Knyazev[2], Gilles Zérah[1]

1 Département de Physique Théorique et Appliquée, CEA-DAM Ile de France
2 Department of Mathematical Sciences, University of Colorado at Denver

Copper Mountain Conference 2008

Parallelization: one of the main computational tasks

- Large supercell *ab initio* calculations are very time consuming.
- Increase of the power and number of processors of supercomputers

Site	Computer	kProc	Tflops
DOE/NNSA/LLNL (US)	BlueGene	212	478
Forschungszentrum Juelich (D)	Blue Gene/P	65	167
New Mexico Computing Center(US)	SGI	14	126
Computational Res. Lab.(India)	EKA Cluster Xeon	14	117
Government Agency (Sweden)	Cluster, Xeon	13	102

- Aim: to use and to be efficient on these supercomputers.

The Norm-Conserving method (NC)

Hamiltonian

$$\tilde{\mathcal{H}} = -\frac{\Delta}{2} + \mathrm{v_{loc}} + \sum_{\mathrm{I}} \sum_{\mathrm{lm}\in\mathrm{I}} \frac{|\tilde{\mathcal{P}}^{\mathrm{I}}_{\mathrm{lm}}\rangle\langle\tilde{\mathcal{P}}^{\mathrm{I}}_{\mathrm{lm}}|}{\langle\tilde{\mathcal{P}}^{\mathrm{I}}_{\mathrm{lm}}|\tilde{\Phi}^{\mathrm{I}}_{\mathrm{lm}}\rangle} \quad \text{with} \quad \tilde{\mathcal{H}}\tilde{\Psi}_{\mathrm{nk}} = \epsilon_{\mathrm{nk}}\tilde{\Psi}_{\mathrm{nk}}$$

Density

$$\tilde{\mathrm{n}}(\mathbf{r}) = \sum_{\mathrm{nk}} \mathrm{f_{nk}} |\tilde{\Psi}_{\mathrm{nk}}(\mathbf{r})|^2$$

The Projector Augmented-Wave method (PAW)

Hamiltonian

$$\tilde{\mathcal{H}} = -\frac{\Delta}{2} + \mathrm{v_{loc}} + \sum_{\mathrm{I}} \sum_{\mathrm{ij}\in\mathrm{I}} |\tilde{\mathcal{P}}_{\mathrm{i}}^{\mathrm{I}}\rangle \mathrm{D}_{\mathrm{ij}}^{\mathrm{I}} \langle\tilde{\mathcal{P}}_{\mathrm{j}}^{\mathrm{I}}| \quad \text{with} \quad \tilde{\mathcal{H}}\tilde{\Psi}_{\mathrm{nk}} = \epsilon_{\mathrm{nk}} \mathcal{O} \tilde{\Psi}_{\mathrm{nk}}$$

Densities – Overlap operator

$$\hat{\mathrm{n}}(\mathbf{r}) \quad \text{and} \quad \rho_{\mathrm{ij}}^{\mathrm{I}} = \sum_{\mathrm{nk}} \mathrm{f_{nk}} \langle\tilde{\Psi}_{\mathrm{nk}}|\tilde{\mathcal{P}}_{\mathrm{i}}^{\mathrm{I}}\rangle\langle\tilde{\mathcal{P}}_{\mathrm{j}}^{\mathrm{I}}|\tilde{\Psi}_{\mathrm{nk}}\rangle$$

$$\mathcal{O} = \mathcal{I} + \sum_{\mathrm{I}} \sum_{\mathrm{ij}\in\mathrm{I}} |\tilde{\mathcal{P}}_{\mathrm{i}}^{\mathrm{I}}\rangle (\langle\Phi_{\mathrm{i}}^{\mathrm{I}}|\Phi_{\mathrm{j}}^{\mathrm{I}}\rangle - \langle\tilde{\Phi}_{\mathrm{i}}^{\mathrm{I}}|\tilde{\Phi}_{\mathrm{j}}^{\mathrm{I}}\rangle) \langle\tilde{\mathcal{P}}_{\mathrm{j}}^{\mathrm{I}}|$$

$$\mathrm{n}(\mathbf{r}) = \tilde{\mathrm{n}}(\mathbf{r}) + \sum_{\mathrm{I}} (\mathrm{n}^{1,\mathrm{I}}(\mathbf{r}) - \tilde{\mathrm{n}}^{1,\mathrm{I}}(\mathbf{r}))$$

SCF loop

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

SCF loop

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

Time consuming parts

- The non-local like terms.

SCF loop

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

Time consuming parts

- The non-local like terms.
- The resolution of the KS equations (LOBPCG).

SCF loop

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \tilde{\mathcal{H}} | \tilde{\Psi}_n \rangle = \epsilon_n \langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \mathcal{O} | \tilde{\Psi}_n \rangle$$

Time consuming parts

- The non-local like terms.
- The resolution of the KS equations (LOBPCG).
- The diagonalisation within the sub-space.

SCF loop

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \tilde{\mathcal{H}} | \tilde{\Psi}_n \rangle = \epsilon_n \langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \mathcal{O} | \tilde{\Psi}_n \rangle$$

Time consuming parts

- The non-local like terms.
- The resolution of the KS equations (LOBPCG).
- The diagonalisation within the sub-space.
- The calculation of the density and local potential (FFT).

SCF loop

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

Possible Parallelizations

- over the **k**-points.
- over the plane-waves → require a parallel 3dim-FFT [a].
- over the bands → require a block eigensolver.

[a] S. Goedecker *et al.*, Comput. Phys. Comm. **154**, 105 (2003).

Algorithm 1 LOBPCG

Require: $\mathbf{\Psi}^0 = \{\Psi_1^0, \ldots, \Psi_m^0\}$ close to the minimum and K a preconditioner; $\mathbf{P} = \{P_1^{(0)}, \ldots, P_m^0\}$ is initialized to 0.

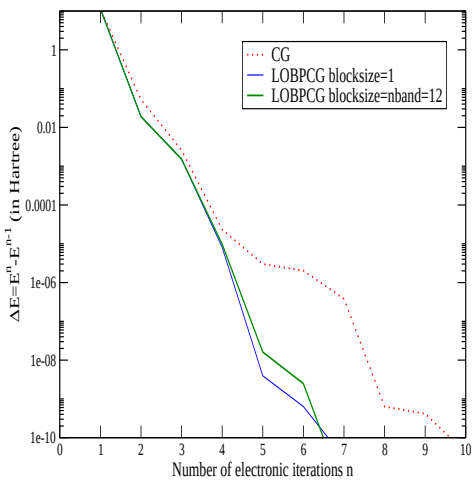
1: **for** i=0,1,...,κ **do**
2: $\quad \Upsilon^{(i)} = \Upsilon(\mathbf{\Psi}^{(i)})$
3: $\quad \mathbf{R}^{(i)} = \mathcal{H}\mathbf{\Psi}^{(i)} - \Upsilon^{(i)}\mathcal{O}\mathbf{\Psi}^{(i)}$
4: $\quad \mathbf{W}^{(i)} = K\mathbf{R}^{(i)}$
5: $\quad$ The Rayleigh-Ritz method is applied within the subspace $\Xi = \left\{P_1^{(i)}, \ldots, P_m^{(i)}, \Psi_1^{(i)}, \ldots, \Psi_m^{(i)}, W_1^{(i)}, \ldots, W_m^{(i)}\right\}$
6: $\quad \mathbf{\Psi}^{(i+1)} = \Delta^{(i)}\mathbf{\Psi}^{(i)} + \Lambda^{(i)}\mathbf{W}^{(i)} + \Gamma^{(i)}\mathbf{P}^{(i)}$
7: $\quad \mathbf{P}^{(i+1)} = \Lambda^{(i)}\mathbf{W}^{(i)} + \Gamma^{(i)}\mathbf{P}^{(i)}$
8: **end for**

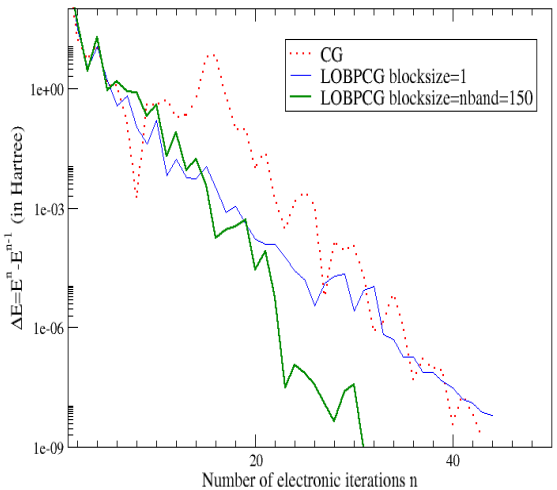
More efficient than CG in many cases [1]

[1] A. Knyazev *Toward the Optimal Preconditioned Eigensolver: Locally Optimal Block Preconditioned Conjugate Gradient Method*, SIAM Journal on Scientific Computing **23**, 517 (2001).

Carbon in its Diamond phase, Plutonium in its alpha phase

We compare also the use of different blocksizes: blocksize=1 and blocksize=nband





- The convergence is linear up to a higher precision for lobpcg, but blocksize=nband yields better results

Implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

In LOBPCG: blocks of size m

$$\begin{matrix} c_{1:m}(g_{11}) & c_{1:m}(g_{12}) & \dots & c_{1:m}(g_{1p}) \\ c_{1:m}(g_{21}) & c_{1:m}(g_{22}) & \dots & c_{1:m}(g_{2p}) \\ \vdots & \vdots & \ddots & \vdots \\ c_{1:m}(g_{m1}) & \dots & \dots & c_{1:m}(g_{mp}) \end{matrix}$$

Implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

In LOBPCG: blocks of size m

$$\begin{matrix} c_{1:m}(g_{11}) & c_{1:m}(g_{12}) & \dots & c_{1:m}(g_{1p}) \\ c_{1:m}(g_{21}) & c_{1:m}(g_{22}) & \dots & c_{1:m}(g_{2p}) \\ \vdots & \vdots & \ddots & \vdots \\ c_{1:m}(g_{m1}) & \dots & \dots & c_{1:m}(g_{mp}) \end{matrix}$$

Implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

In LOBPCG: blocks of size m

$$\begin{matrix} c_{1:m}(g_{11}) & c_{1:m}(g_{12}) & \dots & c_{1:m}(g_{1p}) \\ c_{1:m}(g_{21}) & c_{1:m}(g_{22}) & \dots & c_{1:m}(g_{2p}) \\ \vdots & \vdots & \ddots & \vdots \\ c_{1:m}(g_{m1}) & \dots & \dots & c_{1:m}(g_{mp}) \end{matrix}$$

Implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

In LOBPCG: blocks of size m

$$\begin{matrix} c_{1:m}(g_{11}) & c_{1:m}(g_{12}) & \cdots & c_{1:m}(g_{1p}) \\ c_{1:m}(g_{21}) & c_{1:m}(g_{22}) & \cdots & c_{1:m}(g_{2p}) \\ \vdots & \vdots & \ddots & \vdots \\ c_{1:m}(g_{m1}) & \cdots & \cdots & c_{1:m}(g_{mp}) \end{matrix}$$

Implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

... which can be transposed...

$$\begin{vmatrix} c_1(g_{11}) & c_1(g_{21}) & \dots & c_1(g_{m1}) \\ c_2(g_{11}) & c_2(g_{21}) & \dots & c_2(g_{m1}) \\ \vdots & \vdots & \ddots & \vdots \\ c_m(g_{11}) & \dots & \dots & c_m(g_{m1}) \end{vmatrix}$$

Implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

... which can be transposed...

$$\begin{vmatrix} c_1(g_{11}) & c_1(g_{21}) & \dots & c_1(g_{m1}) \\ c_2(g_{11}) & c_2(g_{21}) & \dots & c_2(g_{m1}) \\ \vdots & \vdots & \ddots & \vdots \\ c_m(g_{11}) & \dots & \dots & c_m(g_{m1}) \end{vmatrix}$$

Implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

... which can be transposed...

$$\begin{vmatrix} c_1(g_{11}) & c_1(g_{21}) & \dots & c_1(g_{m1}) \\ c_2(g_{11}) & c_2(g_{21}) & \dots & c_2(g_{m1}) \\ \vdots & \vdots & \ddots & \vdots \\ c_m(g_{11}) & \dots & \dots & c_m(g_{m1}) \end{vmatrix}$$

Implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

... which can be transposed...

$$\begin{vmatrix} c_1(g_{11}) & c_1(g_{21}) & \dots & c_1(g_{m1}) \\ c_2(g_{11}) & c_2(g_{21}) & \dots & c_2(g_{m1}) \\ \vdots & \vdots & \ddots & \vdots \\ c_m(g_{11}) & \dots & \dots & c_m(g_{m1}) \end{vmatrix}$$

Implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

... which can be transposed...

$$\begin{vmatrix} c_1(g_{11}) & c_1(g_{21}) & \dots & c_1(g_{m1}) \\ c_2(g_{11}) & c_2(g_{21}) & \dots & c_2(g_{m1}) \\ \vdots & \vdots & \ddots & \vdots \\ c_m(g_{11}) & \dots & \dots & c_m(g_{m1}) \end{vmatrix}$$

Implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

... which can be transposed...

$$\begin{vmatrix} c_1(g_{11}) & c_1(g_{21}) & \dots & c_1(g_{m1}) \\ c_2(g_{11}) & c_2(g_{21}) & \dots & c_2(g_{m1}) \\ \vdots & \vdots & \ddots & \vdots \\ c_m(g_{11}) & \dots & \dots & c_m(g_{m1}) \end{vmatrix}$$

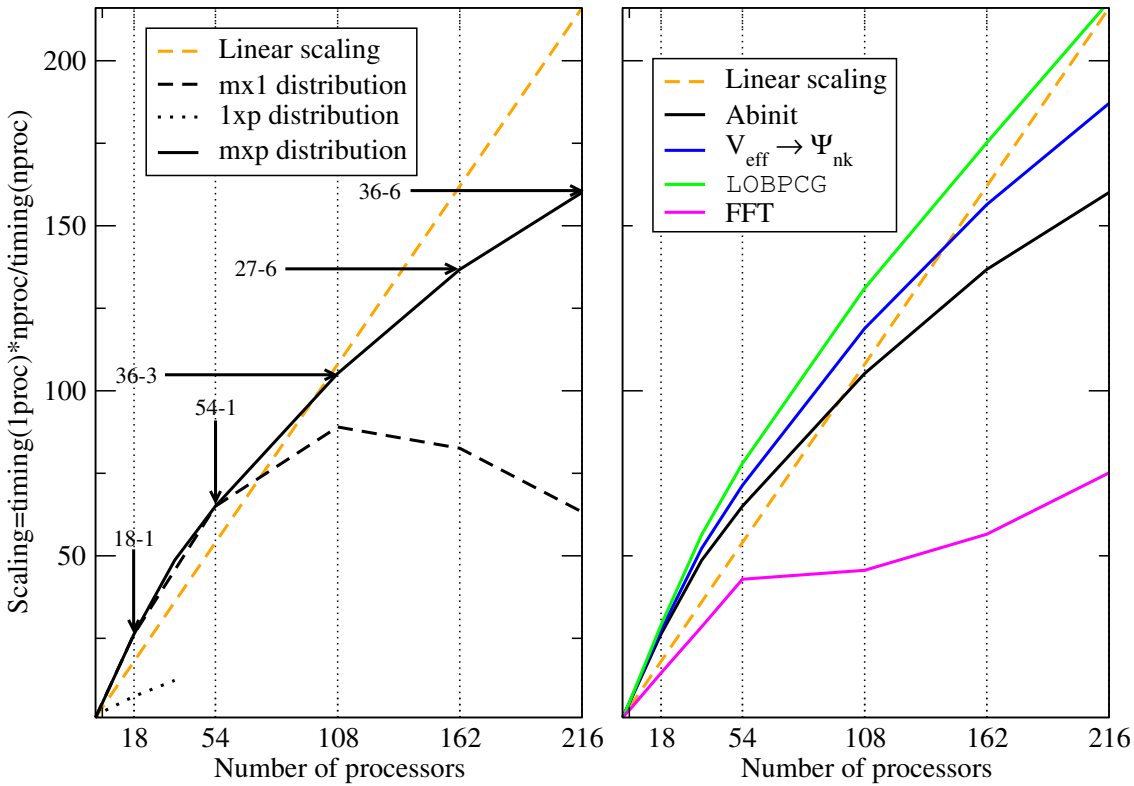
Benchmarks – Test cases

- Gold system with 108 atoms, 648 bands, 108^3 FFT grid, 1 **k**-point and E_{cut}=24 Ha.
- We stop the SCF for nstep=15
- A two-dimensional grid of processors with nproc=1, 4, 18, 36, 54, 108, 162 and 216.
- Example: for nproc=108, we can choose m×p=108×1, 54×2, 36×3, 27×4, 18×6, 12×9, 9×12, 6×18, 4×27 and 3×36
- Tests are performed on 2 supercomputers:

Supercomputer	Node	Interconnection
Tantale (CCRT)	4-procs AMD OPTERON 2.4 GHz	Infiniband
TERA-10 (CEA/DIF)	Novascale 16-procs Intel Itanium	Quadrics

Norm-conserving Results

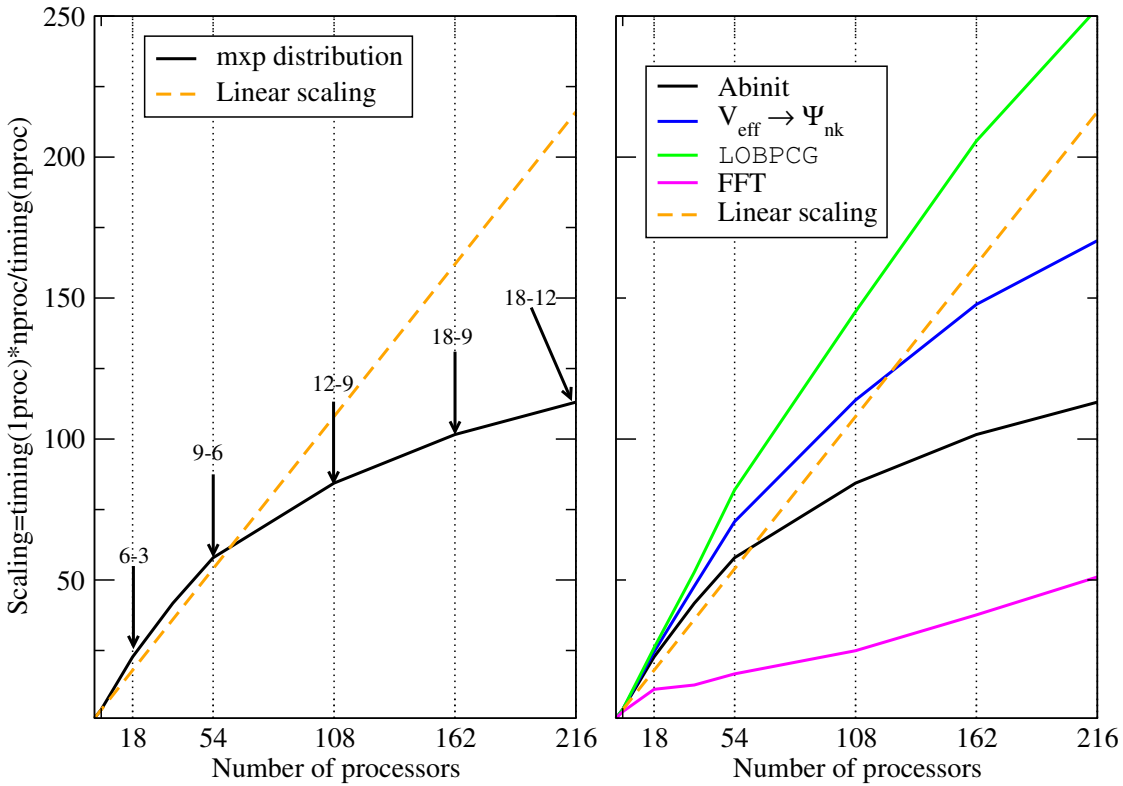
108 atoms, 648 bands, 108^3 FFT grid, 1 **k**-point and E_{cut}=24 Ha.



- In sequential: 90 000 sec. (with 90% in LOBPCG).
- Linear scaling up to 100 for ABINIT and 200 for LOBPCG.

PAW results (with 2 projectors per angular momentum)

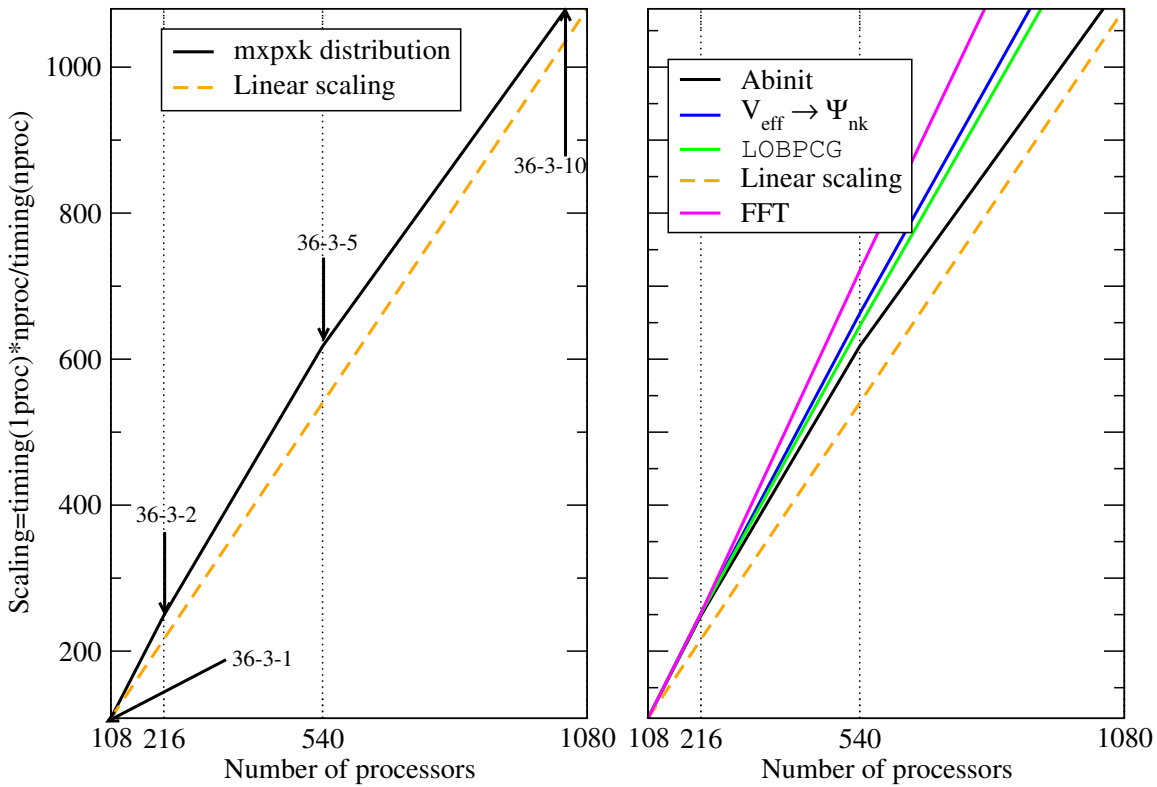
108 atoms, 648 bands, 72^3 FFT grid, 1 **k**-point and E_{cut}=12 Ha.



- In sequential: 75 000 sec. (with 90% in LOBPCG).
- Due to non-local like and spherical terms.

The triple nkG parallelization (in ABINIT 5.5.x)

108 atoms, 648 bands, 108^3 FFT grid, 10 **k**-point and E_{cut}=24 Ha.



Algorithm 2 LOBPCGII

Require: $\mathbf{\Psi}^0 = \{\Psi_1^0, \ldots, \Psi_m^0\}$ close to the minimum and K a preconditioner; $\mathbf{P} = \{P_1^{(0)}, \ldots, P_m^0\}$ is initialized to 0.

1: **for** i=0,1,...,κ **do**
2: $\quad \Upsilon^{(i)} = \Upsilon(\mathbf{\Psi}^{(i)})$
3: $\quad \mathbf{R}^{(i)} = \mathcal{H}\mathbf{\Psi}^{(i)} - \Upsilon^{(i)}\mathcal{O}\mathbf{\Psi}^{(i)}$
4: $\quad \mathbf{W}^{(i)} = K\mathbf{R}^{(i)}$
5: $\quad$ The Rayleigh-Ritz method is applied within each subspace $\Xi_j = \left\{P_1^{(i)}, \Psi_1^{(i)}, W_1^{(i)}, \right\}$
6: $\quad \hat{\Psi}_j^{(i+1)} = \Delta_j^{(i)}\Psi_j^{(i)} + \Lambda_j^{(i)}W_j^{(i)} + \Gamma_j^{(i)}P_j^{(i)}$
7: $\quad P_j^{(i+1)} = \Lambda_j^{(i)}W_j^{(i)} + \Gamma_j^{(i)}P_j^{(i)}$
8: $\quad$ Apply RR on the subspace $\{\hat{\Psi}_1^{i+1}, \ldots, \hat{\Psi}_m^{i+1}\}$
9: **end for**

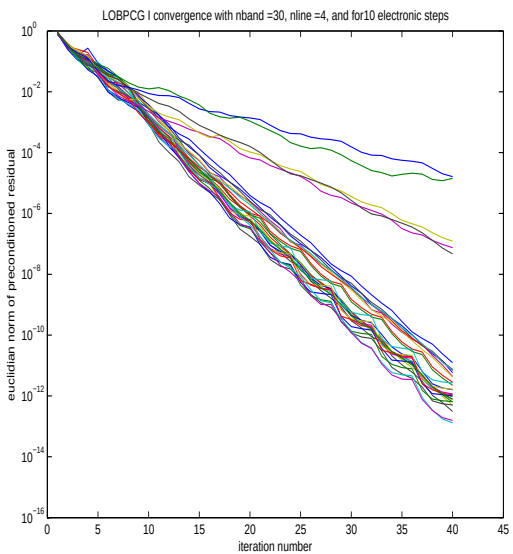
Computes a smaller Ritz matrice. Now the P are somewhat inconsistent with the Ψ

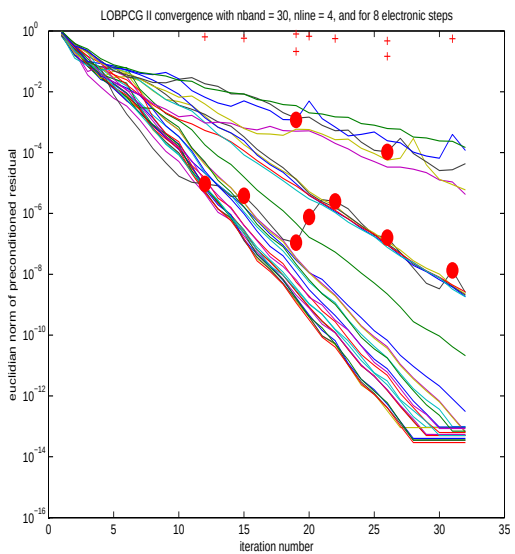
Lobpcgll: discussion

- Eigenvalues are sometimes runnig away
- This is due to the lack of orthogonalization step
- To remain in the spirit of CG eigensolvers the P must be modified
- A good approximation is to impose $\mathrm{span}(\Psi_j^{i+1}, \Psi_j^i) = \mathrm{span}(\Psi_j^{i+1}, P_j^{i+1})$

LOBPCGI vs LOBPCGII

We compare also the use of different blocksizes: blocksize=1 and blocksize=nband





Conclusion

- LOBPCG is responsible for the superlinear behaviour and scales perfectly up to 200 processors.
- ABINIT scales lineary up to 100 processors in NC and slightly underneath in PAW.
- LOBPCGII could yield an even greater gain in scalability, by avoiding the orthogonalization step
- This is the object of our current studies