



Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

# Large scale electronic structure calculations with the LOBPCG method

François Bottin<sup>1</sup>, Stéphane Leroux<sup>1</sup>, Andrew Knyazev<sup>2</sup>, Gilles Zérah<sup>1</sup>

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.**



Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs  
CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs  
CG

Conclusion

Conclusion



Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

- 1 Theoretical background**
  - *The Norm-Conserving method (NC)*
  - *The Projector Augmented-Wave method (PAW)*
  - *The self-consistent loop (SCF)*
  - *Locally Optimal Block Preconditioned Conjugate Gradient method (LOBPCG)*
  - *LOBPCG method versus CG*
- 2 BandFFT parallelization**
  - *Implementation*
- 3 BandFFT Results**
  - *Benchmarks*
  - *Norm-conserving Results*
  - *PAW results*
  - *Adding k-points parallelization*
- 4 Tests with LOBPCGII**
  - *Avoiding the 3mX3m RR*
  - *LobpcgII: discussion*
  - *LOBPCGI method versus LOBPCGII*
- 5 Conclusion – Prospects**
  - *Conclusion*



Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

- 1 Theoretical background**
  - *The Norm-Conserving method (NC)*
  - *The Projector Augmented-Wave method (PAW)*
  - *The self-consistent loop (SCF)*
  - *Locally Optimal Block Preconditioned Conjugate Gradient method (LOBPCG)*
  - *LOBPCG method versus CG*
- 2 BandFFT parallelization**
  - *Implementation*
- 3 BandFFT Results**
  - *Benchmarks*
  - *Norm-conserving Results*
  - *PAW results*
  - *Adding k-points parallelization*
- 4 Tests with LOBPCGII**
  - *Avoiding the 3mX3m RR*
  - *LobpcgII: discussion*
  - *LOBPCGI method versus LOBPCGII*
- 5 Conclusion – Prospects**
  - *Conclusion*

## The Norm-Conserving method (NC)



Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs  
CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs  
CG

Conclusion

Conclusion

### Hamiltonian

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

### Density

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

## The Projector Augmented-Wave method (PAW)



Theory

NC

**PAW**

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

### Hamiltonian

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

### Densities – Overlap operator

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

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

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



## SCF loop

Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

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

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

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

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



## SCF loop

Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

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

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

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

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

## Time consuming parts

- The non-local like terms.



## SCF loop

Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

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

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

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

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

## Time consuming parts

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



## SCF loop

Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

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

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

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

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

## Time consuming parts

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



## SCF loop

Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

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

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

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

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

## 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

Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

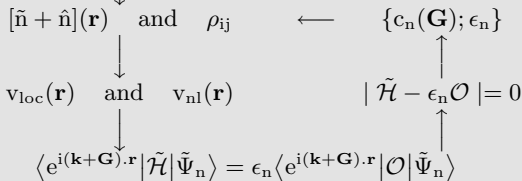
LOBPCG vs

CG

Conclusion

Conclusion

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



## Possible Parallelizations

- over the  $\mathbf{k}$ -points.
- over the plane-waves  $\rightarrow$  require a parallel 3dim-FFT <sup>a</sup>.
- over the bands  $\rightarrow$  require a block eigensolver.

<sup>a</sup>S. Goedecker *et al.*, Comput. Phys. Comm. **154**, 105 (2003).

---

## Algorithm 1 LOBPCG

---

**Require:**  $\Psi^0 = \{\Psi_1^0, \dots, \Psi_m^0\}$  close to the minimum and  $K$  a preconditioner;  $\mathbf{P} = \{\mathbf{P}_1^{(0)}, \dots, \mathbf{P}_m^{(0)}\}$  is initialized to 0.

- 1: **for**  $i=0, 1, \dots, \kappa$  **do**
  - 2:    $\Upsilon^{(i)} = \Upsilon(\Psi^{(i)})$
  - 3:    $\mathbf{R}^{(i)} = \mathcal{H}\Psi^{(i)} - \Upsilon^{(i)}\mathcal{O}\Psi^{(i)}$
  - 4:    $\mathbf{W}^{(i)} = K\mathbf{R}^{(i)}$
  - 5:   The Rayleigh-Ritz method is applied within the subspace  $\Xi = \{\mathbf{P}_1^{(i)}, \dots, \mathbf{P}_m^{(i)}, \Psi_1^{(i)}, \dots, \Psi_m^{(i)}, \mathbf{W}_1^{(i)}, \dots, \mathbf{W}_m^{(i)}\}$
  - 6:    $\Psi^{(i+1)} = \Delta^{(i)}\Psi^{(i)} + \Lambda^{(i)}\mathbf{W}^{(i)} + \Gamma^{(i)}\mathbf{P}^{(i)}$
  - 7:    $\mathbf{P}^{(i+1)} = \Lambda^{(i)}\mathbf{W}^{(i)} + \Gamma^{(i)}\mathbf{P}^{(i)}$
  - 8: **end for**
- 

More efficient than CG in many cases <sup>1</sup>

---

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



Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

## Carbon in its Diamond phase, Plutonium in its alpha phase

We compare also the use of different block sizes:  $\text{blocksize}=1$  and  $\text{blocksize}=\text{nband}$



Theory

NC

PAW

SCF

LOBPCG

LOBPCG VS

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

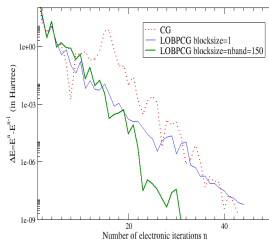
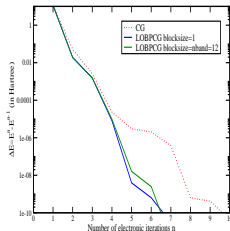
Hypothesis

LOBPCG VS

CG

Conclusion

Conclusion



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



Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

- 1 Theoretical background**
  - *The Norm-Conserving method (NC)*
  - *The Projector Augmented-Wave method (PAW)*
  - *The self-consistent loop (SCF)*
  - *Locally Optimal Block Preconditioned Conjugate Gradient method (LOBPCG)*
  - *LOBPCG method versus CG*
- 2 BandFFT parallelization**
  - *Implementation*
- 3 BandFFT Results**
  - *Benchmarks*
  - *Norm-conserving Results*
  - *PAW results*
  - *Adding k-points parallelization*
- 4 Tests with LOBPCGII**
  - *Avoiding the 3mX3m RR*
  - *LobpcgII: discussion*
  - *LOBPCGI method versus LOBPCGII*
- 5 Conclusion – Prospects**
  - *Conclusion*



## Implementation

Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

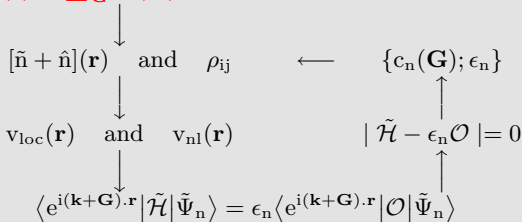
LOBPCG vs

CG

Conclusion

Conclusion

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



## In LOBPCG: blocks of size m

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



## Implementation

Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

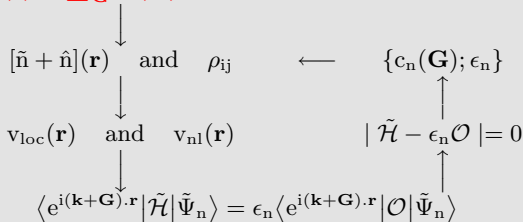
LOBPCG vs

CG

Conclusion

Conclusion

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



## In LOBPCG: blocks of size m

$$\begin{array}{cccc}
 c_{1:m}(\mathbf{g}_{11}) & c_{1:m}(\mathbf{g}_{12}) & \dots & c_{1:m}(\mathbf{g}_{1p}) \\
 c_{1:m}(\mathbf{g}_{21}) & c_{1:m}(\mathbf{g}_{22}) & \dots & c_{1:m}(\mathbf{g}_{2p}) \\
 \vdots & \vdots & \ddots & \vdots \\
 c_{1:m}(\mathbf{g}_{m1}) & \dots & \dots & c_{1:m}(\mathbf{g}_{mp})
 \end{array}$$



## Implementation

Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

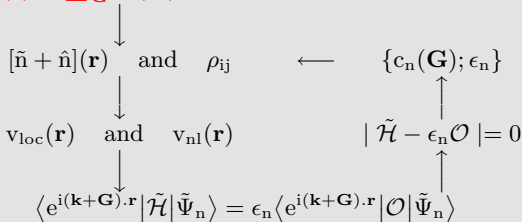
LOBPCG vs

CG

Conclusion

Conclusion

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



## In LOBPCG: blocks of size m

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



## Implementation

Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

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

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

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

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

## In LOBPCG: blocks of size m

$$\begin{array}{cccc} c_{1:m}(\mathbf{g}_{11}) & c_{1:m}(\mathbf{g}_{12}) & \dots & c_{1:m}(\mathbf{g}_{1p}) \\ c_{1:m}(\mathbf{g}_{21}) & c_{1:m}(\mathbf{g}_{22}) & \dots & c_{1:m}(\mathbf{g}_{2p}) \\ \vdots & \vdots & \ddots & \vdots \\ c_{1:m}(\mathbf{g}_{m1}) & \dots & \dots & c_{1:m}(\mathbf{g}_{mp}) \end{array}$$



## Implementation

Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

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

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

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

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

... which can be transposed...

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



## Implementation

Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

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

$$\begin{array}{ccc}
 [\tilde{n} + \hat{n}](\mathbf{r}) & \text{and} & \rho_{ij} & \longleftarrow & \{c_n(\mathbf{G}); \epsilon_n\} \\
 \downarrow & & & & \uparrow \\
 v_{\text{loc}}(\mathbf{r}) & \text{and} & v_{\text{nl}}(\mathbf{r}) & & |\tilde{\mathcal{H}} - \epsilon_n \mathcal{O}| = 0 \\
 & & & & \uparrow
 \end{array}$$

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

... which can be transposed...

$$\begin{array}{cccc|c}
 c_1(\mathbf{g}_{11}) & c_1(\mathbf{g}_{21}) & \dots & c_1(\mathbf{g}_{m1}) & \\
 c_2(\mathbf{g}_{11}) & c_2(\mathbf{g}_{21}) & \dots & c_2(\mathbf{g}_{m1}) & \\
 \vdots & \vdots & \ddots & \vdots & \\
 c_m(\mathbf{g}_{11}) & \dots & \dots & c_m(\mathbf{g}_{m1}) & 
 \end{array}$$



## Implementation

Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

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

$$\begin{array}{ccc}
 [\tilde{n} + \hat{n}](\mathbf{r}) & \text{and} & \rho_{ij} & \longleftarrow & \{c_n(\mathbf{G}); \epsilon_n\} \\
 \downarrow & & & & \uparrow \\
 v_{\text{loc}}(\mathbf{r}) & \text{and} & v_{\text{nl}}(\mathbf{r}) & & |\tilde{\mathcal{H}} - \epsilon_n \mathcal{O}| = 0 \\
 & & & & \uparrow
 \end{array}$$

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

... which can be transposed...

$$\begin{array}{cccc|c}
 c_1(\mathbf{g}_{11}) & c_1(\mathbf{g}_{21}) & \dots & c_1(\mathbf{g}_{m1}) & \\
 c_2(\mathbf{g}_{11}) & c_2(\mathbf{g}_{21}) & \dots & c_2(\mathbf{g}_{m1}) & \\
 \vdots & \vdots & \ddots & \vdots & \\
 c_m(\mathbf{g}_{11}) & \dots & \dots & c_m(\mathbf{g}_{m1}) & 
 \end{array}$$



## Implementation

Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

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

$$\begin{array}{ccc}
 [\tilde{n} + \hat{n}](\mathbf{r}) & \text{and} & \rho_{ij} & \longleftarrow & \{c_n(\mathbf{G}); \epsilon_n\} \\
 \downarrow & & & & \uparrow \\
 v_{\text{loc}}(\mathbf{r}) & \text{and} & v_{\text{nl}}(\mathbf{r}) & & |\tilde{\mathcal{H}} - \epsilon_n \mathcal{O}| = 0 \\
 & & & & \uparrow
 \end{array}$$

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

... which can be transposed...

$$\begin{array}{cccc|c}
 c_1(\mathbf{g}_{11}) & c_1(\mathbf{g}_{21}) & \dots & c_1(\mathbf{g}_{m1}) & \\
 c_2(\mathbf{g}_{11}) & c_2(\mathbf{g}_{21}) & \dots & c_2(\mathbf{g}_{m1}) & \\
 \vdots & \vdots & \ddots & \vdots & \\
 c_m(\mathbf{g}_{11}) & \dots & \dots & c_m(\mathbf{g}_{m1}) &
 \end{array}$$



## Implementation

Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

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

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

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

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

... which can be transposed...

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



## Implementation

Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

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

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

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

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

... which can be transposed...

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



Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

- 1 Theoretical background**
  - *The Norm-Conserving method (NC)*
  - *The Projector Augmented-Wave method (PAW)*
  - *The self-consistent loop (SCF)*
  - *Locally Optimal Block Preconditioned Conjugate Gradient method (LOBPCG)*
  - *LOBPCG method versus CG*
- 2 BandFFT parallelization**
  - *Implementation*
- 3 BandFFT Results**
  - **Benchmarks**
  - **Norm-conserving Results**
  - **PAW results**
  - **Adding k-points parallelization**
- 4 Tests with LOBPCGII**
  - *Avoiding the 3mX3m RR*
  - *LobpcgII: discussion*
  - *LOBPCGI method versus LOBPCGII*
- 5 Conclusion – Prospects**
  - *Conclusion*

## Benchmarks – Test cases



Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

- Gold system with 108 atoms, 648 bands,  $108^3$  FFT grid, 1  $\mathbf{k}$ -point and  $E_{\text{cut}}=24$  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  $m \times p = 108 \times 1$ ,  $54 \times 2$ ,  $36 \times 3$ ,  $27 \times 4$ ,  $18 \times 6$ ,  $12 \times 9$ ,  $9 \times 12$ ,  $6 \times 18$ ,  $4 \times 27$  and  $3 \times 36$
- 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  $\mathbf{k}$ -point and  $E_{\text{cut}}=24$  Ha.



Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

$k$ -points

Implementation

LOBPCG

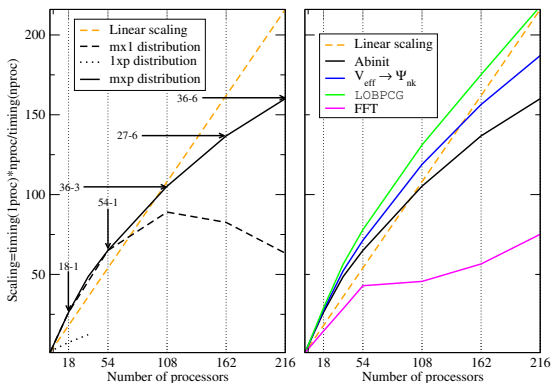
Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion



- 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  $\mathbf{k}$ -point and  $E_{\text{cut}}=12$  Ha.



### Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

### Implementation

Principles

Results

Benchmarks

NC

PAW

$k$ -points

### Implementation

LOBPCG

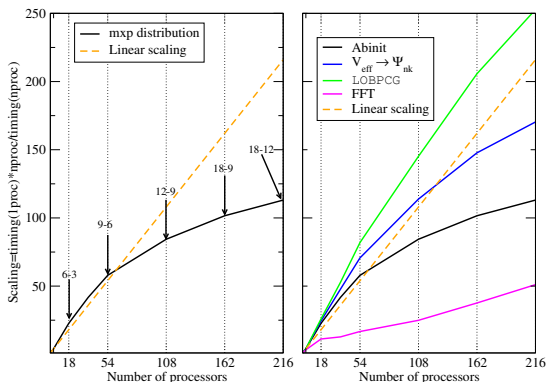
Hypothesis

LOBPCG vs

CG

### Conclusion

Conclusion



- 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  $E_{\text{cut}}=24$  Ha.



Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

**k**-points

Implementation

LOBPCG

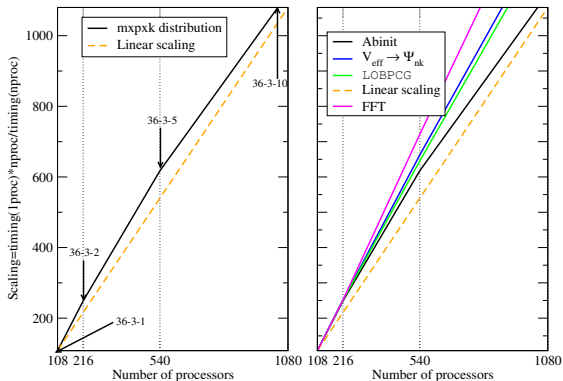
Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion





Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

- 1 Theoretical background**
  - *The Norm-Conserving method (NC)*
  - *The Projector Augmented-Wave method (PAW)*
  - *The self-consistent loop (SCF)*
  - *Locally Optimal Block Preconditioned Conjugate Gradient method (LOBPCG)*
  - *LOBPCG method versus CG*
- 2 BandFFT parallelization**
  - *Implementation*
- 3 BandFFT Results**
  - *Benchmarks*
  - *Norm-conserving Results*
  - *PAW results*
  - *Adding k-points parallelization*
- 4 Tests with LOBPCGII**
  - *Avoiding the 3mX3m RR*
  - *LobpcgII: discussion*
  - *LOBPCGI method versus LOBPCGII*
- 5 Conclusion – Prospects**
  - *Conclusion*

**Algorithm 2** LOBPCGII

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

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

Computes a smaller Ritz matrix. Now the  $P$  are somewhat inconsistent with the  $\Psi$



Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

## LobpcgII: discussion



### Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

### Implementation

Principles

### Results

Benchmarks

NC

PAW

k-points

### Implementation

LOBPCG

**Hypothesis**

LOBPCG vs

CG

### Conclusion

Conclusion

- Eigenvalues are sometimes running 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  $\text{span}(\Psi_j^{i+1}, \Psi_j^i) = \text{span}(\Psi_j^{i+1}, P_j^{i+1})$

## LOBPCGI vs LOBPCGII

We compare also the use of different blocksizes:  $\text{blocksize}=1$  and  $\text{blocksize}=\text{nband}$



Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

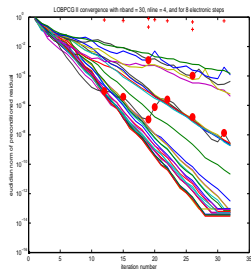
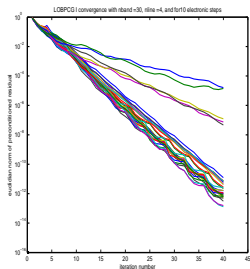
Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion





Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion

- 1 Theoretical background**
  - *The Norm-Conserving method (NC)*
  - *The Projector Augmented-Wave method (PAW)*
  - *The self-consistent loop (SCF)*
  - *Locally Optimal Block Preconditioned Conjugate Gradient method (LOBPCG)*
  - *LOBPCG method versus CG*
- 2 BandFFT parallelization**
  - *Implementation*
- 3 BandFFT Results**
  - *Benchmarks*
  - *Norm-conserving Results*
  - *PAW results*
  - *Adding k-points parallelization*
- 4 Tests with LOBPCGII**
  - *Avoiding the 3mX3m RR*
  - *LobpcgII: discussion*
  - *LOBPCGI method versus LOBPCGII*
- 5 Conclusion – Prospects**
  - *Conclusion*



## Conclusion

- LOBPCG is responsible for the superlinear behaviour and scales perfectly up to 200 processors.
- ABINIT scales linearly 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

Theory

NC

PAW

SCF

LOBPCG

LOBPCG vs

CG

Implementation

Principles

Results

Benchmarks

NC

PAW

k-points

Implementation

LOBPCG

Hypothesis

LOBPCG vs

CG

Conclusion

Conclusion