# ESLW_Drivers

10-21 July 2017

Volker Blum - ELSI

Viktor Yu - ELSI

William Huhn - ELSI

David Lopez - Siesta

Yann Pouillon - Abinit

Micael Oliveira – Octopus & Abinit

Fabiano Corsetti – Siesta & Onetep

Paolo Giannozzi – QE

Anoop Chandran - QE

Pietro Delugas - QE

Ivan Carnimeo - QE

Emine Kucukbenli - QE

Layla Martin-Samos - QE

Stefano de Gironcoli - QE

Complementary to the previous ESL workshops, which addressed solvers for a variety of specific representations of the KS problem, "drivers" targeted the problem of extracting existing KS solvers from major Electronic Structure codes and refactor them as libraries in a representation-agnostic fashion, resulting in stronger code interoperability.

The two main iterative eigensolvers employed in the *pw.x* code of the *Quantum ESPRESSO* distribution have been completely disentangled from the rest of the code and converted to libraries. The solvers make use of a Linear Algebra domain-specific library LAXlib, developed within the MaX CoE, which is interfaced with ELPA and ScalaPack.

Solvers exploit MPI parallelization and in addition to basis-set component distribution, a parallelization over target states is possible, as well as a specific parallelization for the dense linear algebra.

Generic k-point as well as Gamma specific versions of the solvers are included. The Reverse Communication Interface (RCI) paradigm, allowing for a complete abstraction from the basis type and the interface used to perform the matrix-vector operations, has also been implemented for one of the solvers.

Diagonalization of $H_{KS}$ is a major step in the scf solution of any system.

In pw.x two methods are implemented:

- Davidson diagonalization
  - efficient in terms of number of Hpsi required
  - memory intensive: requires a work space up to
    (1+3**david*) * nbnd * npwx
    and diagonalization of matrices up to
    *david**nbnd x *david**nbnd
    where *david* is by default 4, but can be reduced to 2

- Conjugate Gradient
  - memory friendly: bands are dealt with one at a time.
  - the need to orthogonalize to lower states makes it intrinsically sequential and not efficient for large systems.

# Davidson Diagoalization

- Given trial eigenpairs: $\{|\phi_i^{(n)}\rangle, \varepsilon_i^{(n)}\}$
- Eigenpairs of the reduced Hamiltonian

$$\tilde{H}_{ij} = \langle\phi_i^{(n)}|H_{KS}|\phi_j^{(n)}\rangle, \qquad \tilde{S}_{ij} = \langle\phi_i^{(n)}|S|\phi_j^{(n)}\rangle$$

- Build the correction vectors $|\tilde{\phi}_i^{(n)}\rangle$

$$|\tilde{\phi}_i^{(n)}\rangle = (H_{diag} - \varepsilon_i S_{diag})^{-1}(H_{KS} - \varepsilon_i S)|\phi_i^{(n)}\rangle$$

- Build an extended reduced Hamiltonian

$$\tilde{H}_{ij} = \langle\phi_i^{(n)}/\tilde{\phi}^{(n)}|H_{KS}|\phi_j^{(n)}/\tilde{\phi}_j^{(n)}\rangle, \quad \tilde{S}_{ij} = \langle\phi_i^{(n)}/\tilde{\phi}_i^{(n)}|S|\phi_j^{(n)}/\tilde{\phi}_j^{(n)}\rangle$$

- Diagonalize the <u>small</u> $2nbnd \times 2nbnd$ reduced Hamiltonian to get the new estimate for the eigenpairs

$$(\tilde{H} - \varepsilon\tilde{S})v = 0 \quad \longrightarrow \quad \{|\phi_i^{(n+1)}\rangle, \varepsilon_i^{(n+1)}\}$$

- Repeat if needed in order to improve the solution

$\rightarrow 3nbnd \times 3nbnd \rightarrow 4nbnd \times 4nbnd \ldots \rightarrow$ <u>$nbnd \times nbnd$</u>

•Davidson diagonalization

-efficient in terms of number of Hpsi required

-memory intensive: requires a work space up to

(1+3*david) * nbnd * npwx

and diagonalization of matrices up to

*david**nbnd x *david**nbnd

where *david* is by default 4, but can be reduced to 2

•routines

- regterg , cegterg *real/cmplx eigen iterative generalized*

- h_psi, s_psi, g_psi

- rdiaghg, cdiaghg *real/cmplx diagonalization H generalized*

# Conjugate Gradient

- For each band, given a trial eigenpair: $\{|\phi_i^{(n)}\rangle, \varepsilon_i\}$

- Minimize the single particle energy

$$E(|\phi_i\rangle) = \langle\phi_i|H_{KS}|\phi_j\rangle$$

by (pre-conditioned) CG method

subject to the constraints

$$\langle\phi_i|S|\phi_j\rangle = \delta_{ij}, \quad \forall j \leq i$$

- Repeat for next band until completed

•Conjugate gradient

-memory friendly: bands are dealt with one at a time.

-the need to orthogonalize to lower states makes it intrinsically sequential and not efficient for large systems.

•routines

- rcgdiagg , ccgdiagg *real/cmplx CG diagonalization generalize*

- rotate_wfc_gamma, rotate_wfc_k *real/cmplx initial diag*

- h_1psi, s_1psi

  * preconditioning

The software developed during the Workshop is hosted by the e-cam gitlab server as a public sub-project of the ESL initiative (gitlab.e-cam2020/esl/ESLW_Drivers). It will be released under LGPL 2.1+

A toy code implementing the Cohen-Bergstresser empirical pseudopotential method is included to exemplify the use of the solvers and allow a quick test of their functionalities.

A git branch to contribute additional toy codes based on different basis set representations, tight-binding or real-space, to explicitly demonstrate the representation-agnostic nature of the solvers has been created and is under development.

A version of the PPCG algorithm is being ported.

Contribution to the KS_Solvers library of additional solvers from other codes is also foreseen.

# https://gitlab.e-cam2020.eu/esl/ESLW_Drivers

| | |
|---|---|
| CB_toy_code/Doc | *empty* |
| /examples | *contains inputs and ref. outputs* |
| /src | *contains simple code mains* |
| FFTXlib | *fft library used by CB_toy_code* |
| KS_Solvers/CG | *band-by-band CG* |
| /Davidson | *Davidson iterative diagonalization* |
| /Davidson_RCI | *Reverse Comm Interf version* |
| /PPCG | *under construction* |
| LAXlib | *linear algebra library (int w ELPA)* |
| UtilXlib | *basic utilities (error,timinig,para)* |
| archive | *library archive (lapack source)* |
| clib | *c timing routine* |
| include | |
| install | *configure, makedeps* |
| Makefile | |
| configure | |

# PPCG – Projected Preconditioned Conjugate Gradient

E. Vecharynski, C. Yang, J.E. Pask, *J. Comp.Phys.* ***290***,73 (2015)

**Algorithm 2:** The projected preconditioned conjugate gradient (PPCG) algorithm.

**Input**: The matrix $A$, a preconditioner $T$, and a starting guess of the invariant subspace $X^{(0)} \in \mathbb{C}^{n\times k}$ associated with the $k$ smallest eigenvalues of $A$;

**Output**: An approximate invariant subspace $X \in \mathbb{C}^{n\times k}$ associated with the $k$ smallest eigenvalues of $A$;

1: $X \leftarrow \mathtt{orth}(X^{(0)})$; $P \leftarrow [\ ]$;
2: **while** convergence not reached **do**
3: $W \leftarrow T(AX - X(X^*AX))$;
4: $W \leftarrow (I - XX^*)W$;
5: $P \leftarrow (I - XX^*)P$;
6: **for** $j = 1, \ldots, k$ **do**
7: $S \leftarrow [x_j, w_j, p_j]$;
8: Find the smallest eigenpair $(\theta_{\min}, c_{\min})$ of $S^*ASc = \theta S^*Sc$, where $c^*S^*Sc = 1$;
9: $\alpha_j \leftarrow c_{\min}(1)$, $\beta_j \leftarrow c_{\min}(2)$; and $\gamma_j \leftarrow c_{\min}(3)$ ($\gamma_j = 0$ at the initial step);
10: $p_j \leftarrow \beta_j w_j + \gamma_j p_j$;
11: $x_j \leftarrow \alpha_j x_j + p_j$.
12: **end for**
13: $X \leftarrow \mathtt{orth}(X)$;
14: If needed, perform the Rayleigh–Ritz procedure within span($X$);
15: **end while**

each band (or small group of bands) is updated by diagonalizing a small 3*blksize x 3*blksize matrix built from the current X, the orthogonal residual and the orthogonal conjugate direction

• PPCG work in progress

•-memory friendly: bands are dealt with a small block at a time.
•-global calls to h_psi give opportunity for band parallelization
(not working properly yet)
•-each block can be dealt with independently (parallelization)
•-most operations on arrays use efficient BLAS3 calls (DGEMM)

•routines

- ppcg, *real PPCG, cmplx version presently not available*

- rotate_wfc_gamma, *real initial diag (the same as CG)*

- h_psi, (s_psi) *generalized algorithm not available yet*

* preconditioning

# Some recent work on an alternative iterative methods

## A PARALLEL ORBITAL-UPDATING APPROACH FOR ELECTRONIC STRUCTURE CALCULATIONS *

XIAOYING DAI†, XINGAO GONG‡, AIHUI ZHOU† , AND JINWEI ZHU†

**Abstract.** In this paper, we propose an orbital iteration based parallel approach for electronic structure calculations. This approach is based on our understanding of the single-particle equations of independent particles that move in an effective potential. With this new approach, the solution of the single-particle equation is reduced to some solutions of independent linear algebraic systems and a small scale algebraic problem. It is demonstrated by our numerical experiments that this new approach is quite efficient for full-potential calculations for a class of molecular systems.

arXiv:1405.0260v2 [math.NA] 20/11/2014

## A PARALLEL ORBITAL-UPDATING BASED OPTIMIZATION METHOD FOR ELECTRONIC STRUCTURE CALCULATIONS *

XIAOYING DAI†, ZHUANG LIU‡, XIN ZHANG§, AND AIHUI ZHOU¶

**Abstract.** In this paper, we propose a parallel optimization method for electronic structure calculations based on a single orbital-updating approximation. It is shown by our numerical experiments that the method is efficient and reliable for atomic and molecular systems of large scale over supercomputers.

arXiv:1510.07230v1 [math.NA] 25/10/2015

# ParO in a nutshell

ALGORITHM 1.1.

1. *Given initial data* $(\lambda_i^{(0)}, u_i^{(0)}) \in \mathbb{R} \times H_0^1(\Omega)$ *with* $(u_i^{(0)}, u_j^{(0)})_\Omega = \delta_{ij}, (i, j = 1, 2, \cdots, N)$, *define* $\mathcal{T}_0$ *and* $V_0$, *and let* $n = 0$
2. *Construct* $\mathcal{T}_{n+1}$ *and* $V_{n+1}$ *based on an adaptive procedure to* $(\lambda_i^{(n)}, u_i^{(n)})$.
3. *For* $i = 1, 2, \cdots, N$, *find* $u_i^{(n+1/2)} \in V_{n+1}$ *satisfying*

$$a(U^{(n)}; u_i^{(n+1/2)}, v) = \lambda_i^{(n)}(u_i^{(n)}, v) \ \forall v \in V_{n+1}$$

   *in parallel.*
4. *Project to eigenspace: find* $(\lambda^{(n+1)}, u^{(n+1)}) \in \mathbb{R} \times \tilde{V}_{n+1}$ *satisfying* $\|u^{(n+1)}\|_{0,\Omega} = 1$ *and*

$$a(U^{(n+1/2)}; u^{(n+1)}, v) = \lambda^{(n+1)}(u^{(n+1)}, v) \quad \forall v \in \tilde{V}_{n+1}$$

   *to obtain eigenpairs* $(\lambda_i^{(n+1)}, u_i^{(n+1)})(i = 1, 2, \cdots, N)$.
5. *Let* $n = n + 1$ *and go to Step 2.*

*Here* $\tilde{V}_{n+1} = \text{span}\{u_1^{(n+1/2)}, u_2^{(n+1/2)}, \cdots, u_N^{(n+1/2)}\}$, $U^{(n)} = (u_1^{(n)}, u_2^{(n)}, \cdots, u_N^{(n)})$, $U^{(n+1/2)} = (u_1^{(n+1/2)}, u_2^{(n+1/2)}, \cdots, u_N^{(n+1/2)})$, *and* $a(\cdot; \cdot, \cdot)$ *is the nonlinear variational form associated the Kohn-Sham equation defined in Section 2.2.*

arXiv:1405.0260v2 [math.NA] 20/11/2014

# ParO as I understand it

- Given trial eigenpairs: $\{|\phi_i^{(n)}\rangle, \varepsilon_i^{(n)}\}$
- Solve <u>in parallel</u> the *nbnd* linear systems

$$(H_{KS} + \lambda S)|\tilde{\phi}_i^{(n)}\rangle = (\varepsilon_i^{(n)} + \lambda) S|\phi_i^{(n)}\rangle$$

- Build the reduced Hamiltonian

$$\tilde{H}_{ij} = \langle\tilde{\phi}_i^{(n)}|H_{KS}|\tilde{\phi}_j^{(n)}\rangle, \quad \tilde{S}_{ij} = \langle\tilde{\phi}_i^{(n)}|S|\tilde{\phi}_j^{(n)}\rangle$$

- Diagonalize the <u>small</u> *nbnd* x *nbnd* reduced Hamiltonian to get the new estimate for the eigenpairs

$$(\tilde{H} - \varepsilon\tilde{S})v = 0 \quad \longrightarrow \quad \{|\phi_i^{(n+1)}\rangle, \varepsilon_i^{(n+1)}\}$$

- Repeat if needed in order to improve solution at fixed Hamiltonian

# A variant of ParO method

- Given trial eigenpairs: $\{|\phi_i^{(n)}\rangle, \varepsilon_i^{(n)}\}$
- Solve <u>in parallel</u> the *nbnd* linear systems

$$(H_{KS} + \lambda S)|\tilde{\phi}_i^{(n)}\rangle = (\varepsilon_i^{(n)} + \lambda) S|\phi_i^{(n)}\rangle$$

- Build the reduced Hamiltonian from both $|\tilde{\phi}_i^{(n)}\rangle$ & $|\phi_i^{(n)}\rangle$

$$\tilde{H}_{ij} = \langle \tilde{\phi}_i^{(n)}/\phi_i^{(n)}|H_{KS}|\tilde{\phi}_j^{(n)}/\phi_j^{(n)}\rangle, \quad \tilde{S}_{ij} = \langle \tilde{\phi}_i^{(n)}/\phi_i^{(n)}|S|\tilde{\phi}_j^{(n)}/\phi_j^{(n)}\rangle$$

- Diagonalize the <u>small</u> $2nbnd \times 2nbnd$ reduced Hamiltonian to get the new estimate for the eigenpairs

$$(\tilde{H} - \varepsilon\tilde{S})v = 0 \quad \longrightarrow \quad \{|\phi_i^{(n+1)}\rangle, \varepsilon_i^{(n+1)}\}$$

- Repeat if needed in order to improve solution at fixed Hamiltonian

# A variant of ParO method (2)

- Given trial eigenpairs: $\{|\phi_i^{(n)}\rangle, \varepsilon_i^{(n)}\}$
- Solve <u>in parallel</u> the *nbnd* linear systems

$$\left(H_{KS} - \varepsilon_i^{(n)} S + \alpha S|\phi_i^{(n)}\rangle\langle\phi_i^{(n)}|S\right)|\tilde{\phi}_i^{(n)}\rangle = -(H_{KS} - \varepsilon_i^{(n)} S)|\phi_i^{(n)}\rangle$$

- Build the reduced Hamiltonian from both $|\tilde{\phi}_i^{(n)}\rangle$ & $|\phi_i^{(n)}\rangle$

$$\tilde{H}_{ij} = \langle\tilde{\phi}_i^{(n)}/\phi_i^{(n)}|H_{KS}|\tilde{\phi}_j^{(n)}/\phi_j^{(n)}\rangle, \quad \tilde{S}_{ij} = \langle\tilde{\phi}_i^{(n)}/\phi_i^{(n)}|S|\tilde{\phi}_j^{(n)}/\phi_j^{(n)}\rangle$$

- Diagonalize the <u>small</u> 2*nbnd* x 2*nbnd* reduced Hamiltonian to get the new estimate for the eigenpairs

$$(\tilde{H} - \varepsilon\tilde{S})v = 0 \quad \longrightarrow \quad \{|\phi_i^{(n+1)}\rangle, \varepsilon_i^{(n+1)}\}$$

- Repeat if needed in order to improve solution at fixed Hamiltonian

- Domain decomposition parallelization
- Basis set components are distributed
- Memory is distributed

- Band group parallelization
- Operations on dynamically defined band groups are distributed
- Memory is NOT distributed

- Parallel dense diagonalization
- A dedicated communicator is present (interface with ScaLapack and ELPA)