

Document downloaded from:

<http://hdl.handle.net/10251/35132>

This paper must be cited as:

Young, TD.; Romero Alcalde, E.; Román Moltó, JE. (2013). Parallel finite element density functional computations exploiting grid refinement and subspace recycling. *Computer Physics Communications*. 184(1):66-72. doi:10.1016/j.cpc.2012.08.011.



The final publication is available at

<http://dx.doi.org/10.1016/j.cpc.2012.08.011>

Copyright Elsevier

# Parallel Finite Element Density Functional Computations Exploiting Grid Refinement and Subspace Recycling

T. D. Young\*

*Institut Podstawowych Problemów Techniki Polskiej Academia Nauk, ul. Pawińskiego 5b,  
02–106 Warsaw, Poland*

E. Romero

*Institut I3M, Universitat Politècnica de València, Camí de Vera s/n,  
46022 València, Spain*

J. E. Roman

*D. Sistemes Informàtics i Computació, Universitat Politècnica de València, Camí de Vera s/n,  
46022 València, Spain*

---

## Abstract

In this communication computational methods that facilitate finite element analysis of density functional computations are developed. They are: (i)  $h$ -adaptive grid refinement techniques that reduce the total number of degrees of freedom in the real space grid while improving on the approximate resolution of the wanted solution; and (ii) subspace recycling of the approximate solution in self-consistent cycles with the aim of improving the performance of the generalized eigenproblem solver. These techniques are shown to give a convincing speed-up in the computation process by alleviating the overhead normally associated with computing systems with many degrees-of-freedom.

### *Keywords:*

Density functional theory, Finite element discretization, Grid refinement, Large-scale eigenvalue problem, Message-passing parallelization

---

\*Corresponding author

*Email addresses:* tyoung@ippt.pan.pl (T. D. Young), elroal@upvnet.upv.es (E. Romero), jroman@dsic.upv.es (J. E. Roman)

---

## 1. Introduction

Density Functional Theory (DFT) [1, 2] is a first-principles framework used in theoretical and computational physics for determination of the ground-state properties of multi-reference systems, and has far-reaching applicability in nuclear, atomic, and solid-state physics. DFT-based simulations of electronic structure are usually facilitated by means of self-consistent iterations between a pair of coupled equations: (i) Schrödinger’s wave equation that describes the wave-like properties of quantum objects; and (ii) Poisson’s equation that describes the charge density as a functional of the relative charges of the  $N$ -electron system. In  $\mathbb{R}^d$ , a  $d$ -dimensional real space, the coupled Schrödinger-Poisson equation set can be solved by the use of a variety of real space grid techniques [3]. One technique that is under expansive research as a basis for quantum mechanical problems, is the adaptive finite element method.

The cost of solving DFT-based problems in the finite element basis is directly related to the choice of finite element basis functions (order and type), the choice of representation of the grid (geometry, refinement), and the number of iterations required to obtain self-consistency (tolerance). In this paper, computational methods that deal with improving the representation of the underlying real space grid and methods that improve on finding eigenvectors that solve the generalized Schrödinger-like eigenproblem are developed. In that context, it will be shown that our line of research is along similar lines to that of Napoli *et al.* [4].

To date, much of the work whose goal is to solve electronic structure problems on an adaptive finite element grid relies on improving the underlying polynomial interpolation between grid points on which the computation is performed (so-called  $p$ -adaptivity). See for example Ref. [5, 6, 7]. An alternative to improving on the polynomial interpolation of the finite element approximation is to focus on the grid on which computations are performed, *i.e.*,  $h$ -adaptivity. This is achievable by making use of non-regular grids based on geometric (logarithmic) considerations to locally control the resolution of calculations of electronic structure [8, 9, 10]. The underlying motivation there, was that the space grid needs refinement near to the locality of the nucleus, where the potential is steeply changing and where the solution vectors are varying most rapidly. Based on similar considerations, a strategy of  $h$ -adaptivity is developed here with the aim of reducing the number of degrees-of-freedom to consider while improving on the accuracy of the computed result.

A computationally inhibiting factor that affects the cost of iterative solutions toward self-consistency, is the number of eigenvalue problems that need to be solved in the process. The solver that addresses the computation of some eigenpairs of the Schrödinger equation should converge fast and deal with degeneracies and eigenvalue clustering. This makes the problem challenging. Preconditioned conjugate gradient methods have been usually proposed to handle this task, however Davidson methods have been recently demonstrated as being faster and more robust in many cases, *e.g.*, [11]. An efficient solver for the associated algebraic eigenproblem is therefore employed in this work. Here, efficiency is understood in various ways: (i) the ability to exploit the sparsity of matrices; (ii) expedient convergence to a small number of wanted eigenvalues; and (iii) parallel capabilities to address large-scale problems associated with real space grid computations.

This paper is divided into four further sections. In the next section the theoretical framework of DFT in the finite element basis is laid out and the computational toolkits that were extended and adapted for density functional computations are acknowledged. In Section 3 methods for handling error estimates are discussed, in particular, a method that derives an error estimate based on *a posteriori* knowledge of the physical system at hand is developed. Numerical experiments are performed that demonstrate their effectiveness to atomic systems. In Section 4 a computational approach for large-scale eigenvalue problems is described in some detail. Numerical experiments are performed, hand-in-hand with the adaptive grid methods of the previous section, that exploit subspace recycling and parallelization to obtain the eigenfunctions of light atoms in an efficient way. Finally, in Section 5, this contribution is summarised and future developments are outlined.

## 2. Theoretical background

### 2.1. Physical framework in the finite element approach

Density functional methods are based on the solution of Schrödinger's wave equation  $\hat{H}\psi_\mu = \epsilon_\mu\psi_\mu$  determined by a Hamiltonian operator  $\hat{H}$  that contains a kinetic energy term and an effective potential energy term  $V[n]$  that functionally depends on the single-particle density  $n(r)$ . The so-called Kohn-Sham equation

set in atomic units ( $e = m_e = \hbar = 1$ ) is written:

$$\hat{H}\psi_\mu(r) = \left(-\frac{1}{2}\Delta + V(r)\right)\psi_\mu(r) = \epsilon_\mu\psi_\mu, \quad (1a)$$

$$n(r) = \sum_\mu |\psi_\mu(r)|^2 \quad 1 \leq \mu \leq N, \quad (1b)$$

$$V(r) = V_C(r) + \int_{r'} V_H(r, r') dr' + V_{XC}[n](r), \quad (1c)$$

where the eigenmodes (wave-functions) are normalised such that the condition

$$\int_r \psi_\mu \psi_\nu dr = \delta_{\mu\nu}, \quad (1d)$$

holds and thus defines the inner product  $\langle \psi_\mu, \psi_\nu \rangle = \delta_{\mu\nu}$ . The wave-functions  $\psi_\mu$  that solve Schrödinger's wave equation (1a) for an  $N$ -particle system together compose a single-particle density (1b), that is used to determine the effective potential  $V$  in (1a). The effective potential *cf.* Eqn. (1c) is itself composed of three parts: the classical nuclear-atomic Coulomb potential  $V_C$ , a Hartree term  $V_H$  that describes interactions between  $N$  electrons, and the exchange-correlation potential energy  $V_{XC}$ . The Hartree potential term is determined by evaluating Poisson's equation:  $\Delta V_H = -4\pi n(r)$ . The exchange-correlation term  $V_{XC}$  is ignored, reducing the complexity of the ground state for the computational scheme. The finite element method is employed to find a numerical solution to Schrödinger's equation (1a), where piecewise trilinear functions are used to represent the wave-functions  $\psi_\mu$ . After discretization Eqn. (1a) becomes

$$\sum_j \left( \left\langle \frac{1}{2} \nabla \varphi_i, \nabla \varphi_j \right\rangle + \langle V(r) \varphi_i, \varphi_j \rangle \right) v_j = \epsilon \sum_j \langle \varphi_i, \varphi_j \rangle v_j, \quad (2a)$$

where  $\varphi$  are a set of functions and  $v$  are a set of expansion coefficients; and is supplemented with the boundary condition

$$\varphi_i = 0, \quad \forall i \text{ on } \Gamma. \quad (2b)$$

In vector-matrix notation Eqn. (2a) is written as a generalised eigenvalue problem,  $Av = \epsilon Bv$ , the numerical solution to which is discussed in detail in Section 4.

Using the same finite element approach, Poisson’s equation with  $n(r)$  calculated from Eqn. (1b), is expressed as

$$\sum_j \langle \nabla \varphi_i, \nabla \varphi_j \rangle v_j = -4\pi \sum_j \langle \varphi_j, n(r) \rangle, \quad (3)$$

which can be written in vector-matrix notation as  $Av = f$ , and has the same Dirichlet boundary conditions (2b).

In this work, scalar Lagrange finite elements of polynomial order one are employed. This results in eight-node finite elements where in each Cartesian direction two quadrature points are chosen for numerical evaluation of Eqn. (2,3).

## 2.2. Computational toolkit

The computational toolkit sketched out below has been used in some of our previous work [12]. Here, it has been adapted for parallel finite element DFT computations exploiting grid refinement and subspace recycling. (i) `deal.II`, an acronym for the Differential Equations Analysis Library [13] is written in templated C++, provides an abstract set of base tools for computations with adaptive finite elements. Additionally grid management over distributed memory is handled through an interface to the `p4est` library, which facilitates the dynamic management of a collection of “adaptive octrees” [14]. (ii) `SLEPc`, the Scalable Library for Eigenvalue Problem computations [15], is a software package for the solution of large-scale eigenvalue problems on parallel computers. `SLEPc` is built on top of `PETSc` (Portable, Extensible Toolkit for Scientific computation, [16]), a parallel framework for the numerical solution of partial differential equations, whose approach is to encapsulate mathematical algorithms using object-oriented programming techniques to manage the complexity of numerical message-passing codes. `SLEPc` provides a collection of eigensolvers, including several Davidson-type methods, see Section 4. (iii) “namespace `ewalena`” is a library package [17] developed with this work. The library is written in templated C++ and provides an extensible collection of tools for computational physics; in particular, atomic potentials (and their transforms) and provision of a database of physical constants.

## 3. Management of the computational grid

The most common application of error estimates is based on examination of the quality of the computed solution that is then used to flag finite element cells for refinement in regions where the quality is expected to be low. The equations are assembled on a locally refined grid and the refined solution is recomputed.

### 3.1. Kelly error estimate

A tool for estimating the errors in sets of linear partial differential equations is the *a priori* error estimate of Kelly *et al.* [18]. It is based on estimating errors to the Laplace equation,  $\Delta\phi = 0$ , but has been extended to encompass more involved systems [19]. The basic idea is to examine the solution vector(s) of a system of equations and determine local regions on the grid where the solution is inaccurate. The algorithm for determining the error estimate per finite element cell is: (i) compute the second-order derivative (Hessian) of the solution vector at each point on the grid; (ii) integrate the discontinuity between cell faces; and (iii) assign an error estimate based on the measure of the discontinuity to that finite element cell. The greater the discontinuity between cell faces, the greater the error estimate. In other words, it is anticipated that an improvement on the continuity of the gradient of the solution, corresponds to an improvement on the quality of the solution.

In our application, the implementation of the Kelly error estimate is directly taken from the “KellyErrorEstimate” class in the `deal.II` library, where the solution vector corresponds to the electrostatic potential determined from Eqn. (3). Further details are found in the class documentation [13].

### 3.2. Potential-based error estimate

Recall that the solution to the Kohn-Sham equation set is composed of self-consistent iterations of Eqns. (1). The self-consistent procedure can be initiated by making an initial guess of the effective potential, say  $V = V_C$  in Eqn. (1a), after which corrections to the effective potential  $V = V_C + \delta V$  are computed via Eqns. (1b) and (1c). It is given, that if deviations of the effective potential from the Coulomb potential are small, then  $\delta V$  is small and modifications of the wave-functions  $\delta\psi$  are also small. This relation between the solution vectors  $\psi_i$  and the underlying potential  $V$ , leads the thought that the effective potential expressed in the Schrödinger part (1a) of the Kohn-Sham equation set can be effectively used as a basis on which to form an error estimate. This is in contrast to basing an error estimate on a computed solution. The main considerations that make this scheme of work are: (i) the corrections to the effective potential  $\delta V$  are specified by corrections to the wave-functions  $\delta\psi_i$  through Eqn. (1b); (ii) the wave-functions  $\psi_i$  drop-off exponentially as they extend far from the potential barrier (core), and; (iii) a linear shift in the potential function  $V$  does not affect the form of the wave-functions, but the eigenvalues  $\epsilon_i$  only.

Let the function from which an error estimate will be determined be written as a function of the Coulomb potential

$$F(r) \propto f(V_C(r)), \quad (4)$$

where  $r$  denotes the radial distance from the nucleus located at the origin. A generalisation of the following discussion to one in which the nucleus is located elsewhere and/or to multi-atom systems is straightforward.

The energy of the vacuum  $E_0$  is subtracted from the function  $F$  of Eqn. (4), as only changes in energy are important, and the proportionality symbol is replaced by a constant:  $F \rightarrow \eta \cdot |f(V_C - E_0)|$ . This is rewritten as an exponential in  $f$  giving:  $F(r) \rightarrow \exp[-\eta \cdot |f|] - 1$ , where a shift (minus one) normalises the values of  $F(r)$ , as discussed below. The result is the function

$$F(r) = 1 - \exp\left[-\eta \left|\frac{Z}{r} - E_0\right|\right], \quad (5)$$

where  $\eta$  is a proportionality value (set to unity  $\eta = 1$ ), and  $Z$  is the effective charge of the nucleus. Eqn. (5) has some satisfying properties: (i) the function  $F$  is differentiable where  $\lim_{r \rightarrow \pm 0} F(r) = 1$ , and convergent as  $r \rightarrow \pm \infty$ ; *i.e.*, has values in the normalised range  $F \in [0, 1]$ ; and (ii) it is independent of the reference to the vacuum energy. Furthermore, the exponential function (5) can easily be expanded in a power series for the computation of errors to any desired numerical accuracy; though that facet of the method is not considered here.

Following the discussion above in Section 3.1, the discontinuity of the potential function (5) is evaluated by the normal derivative  $\partial/\partial\hat{n}$  on the boundary of each finite element cell  $K$ , and the error estimate

$$\mathcal{E}_K = \left(D \int_{\Gamma_K} \left|\frac{\partial F}{\partial \hat{n}}\right|^2 d\Gamma_K\right)^{\frac{1}{2}}, \quad (6)$$

is thus determined, where  $D$  is the diameter of the cubic finite element cell. In other words, it is anticipated that an improvement on the continuity of the projected representation of the underlying potential (5), corresponds to an improvement on the quality of the solution vectors to Schrödinger's equation (1a).

### 3.3. Numerical experiments

The function  $F$  and the estimated error of Eqn. (6) are plotted together in Fig. 1 where the error estimate has been determined per finite element cell. The initial coarse grid gives the expected poor approximation of the solution: where  $\max[\mathcal{E}_K^{(0)}] \sim 0.25$ . Subsequent application of local refinement with a fixed refinement fraction  $k = 1/d^2 = 1/4$ , leads to a reduction of the overall error to  $\max[\mathcal{E}_K^{(2)}] \sim 0.06$  after two levels of refinement and  $\max[\mathcal{E}_K^{(4)}] \sim 0.006$  after four levels of refinement – a 76% and 98% reduction in the maximal error estimate respectively.

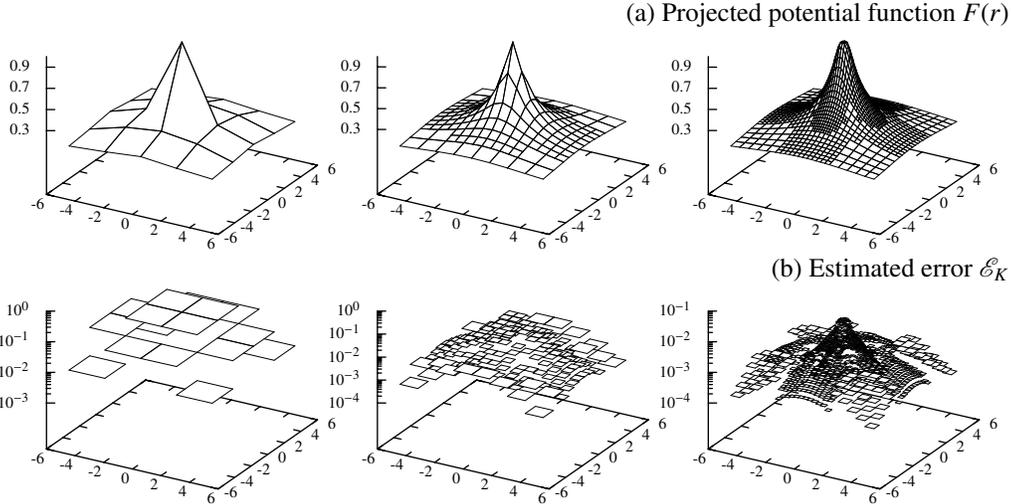


Figure 1: Results for (a) projected potential function (5); and (b) the resultant error estimate  $\mathcal{E}_K$ . From left to right: the globally-refined coarse grid starting from a single cell,  $G = 2$  cycles; the same grid adaptively-refined by  $A = 2$  cycles; and again after  $A = 4$  cycles of adaptive refinement. Note the use of a log-scale to visualize the estimated error.

A stronger test of the effectiveness of the error estimate is to examine the convergence rate of the total energy of the system with respect to the number of active cells in the grid and the number of degrees-of-freedom. Starting from a single three-dimensional finite element cell refined globally,  $G \in \{2, 4, 6\}$  times, three coarse grids were obtained with 64, 4096, and 262144 active cells respectively. From one of the coarse grids, iterative cycle of error ( $A \in \mathbb{N}^+$ ) estimation begins. *For each test, the single-particle energy is computed for benchmarking purposes only – this is strictly not necessary in order for the refinement process to work.* The lowest energy estimated in the tests is  $E' = -0.4987$ , and Fig. 2 plots the relative distance of the estimated energy  $E$  to  $E'$  after successive refinements. We observe that the estimated energy  $E$  of the sequence of grids for any value of  $G$  converges to some value close to  $E'$ . However greater values of  $G$  produce grids with up to 10 times more degrees-of-freedom for similar values of  $E$ .

Finally, the grid refinement patterns obtained using the scheme investigated above and the Kelly error estimate based on computation of the single-particle density Eqn. (1b) are given in Fig. 3 as a projection through the  $xy$ -plane of a three-dimensional grid centered on the point,  $(x, y, z) = 0$ . At first sight the two error estimates seem to produce very similar results. The refinement is clustered around the core of the atom and extends outward. A closer inspection of the grids

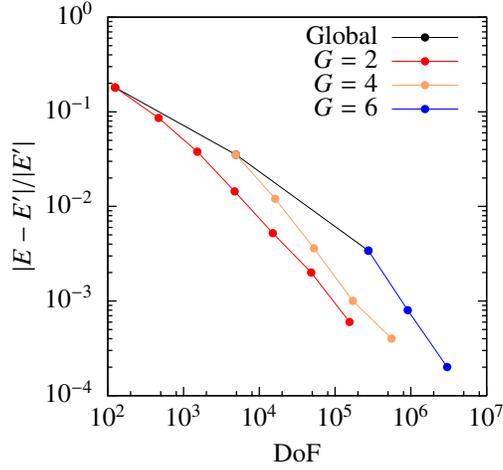


Figure 2: Evolution of both the estimated energy with respect to the best value obtained  $E' = -0.4987$  and the number of degrees-of-freedom of grids after 2, 4, and 6 global refinements ( $G$ ) of a single finite element cell and followed by projected potential refinements with a refinement fraction  $k = 1/4$ . In the case of global refinement, the calculations stop when the memory allocation for the next cycle of adaptivity exceeds that of the machine (12 GB).

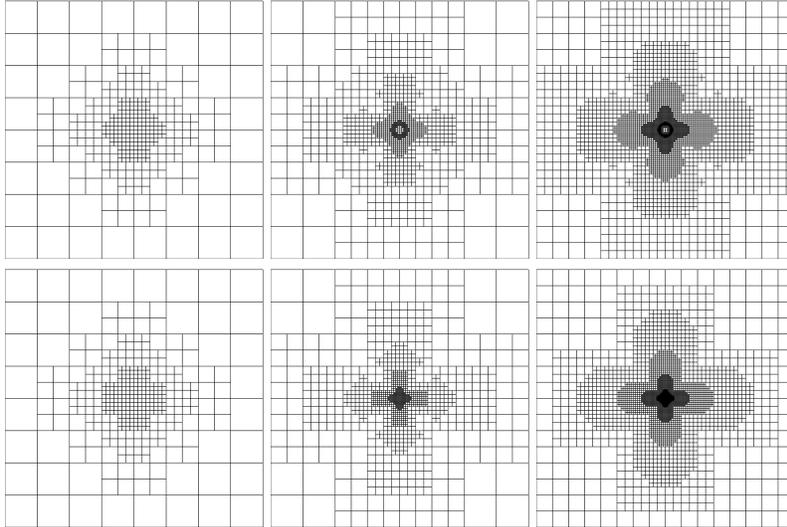


Figure 3: Images of the three-dimensional finite element grid plotted through  $z = 0$  showing three cycles of grid refinement (left-to-right) starting from a coarse grid ( $G = 3$ ) and using a fixed fraction:  $k = 1/d = 1/9$ . Error estimates are based on the Coulomb potential  $V_C$  (top) and single-particle density  $n(r)$  (bottom).

reveal one peculiar difference between refinement schemes, and that is, that the potential based error estimate partially misses the core region. The explanation for this, is that the function constructed in Eqn. (5) is flatter in the core region than the single-particle density function (1b), leading to a spuriously low error estimate in that region. That can be remedied by choosing a higher refinement fraction  $k$  (not shown), or by choosing  $\eta \neq 1$  in Eqn. (5). The latter option is left open to future experimentation.

#### 4. Large-scale eigenvalue problems

In this paper, we are primarily concerned with eigensolutions to the real generalized symmetric-definite eigenvalue problem,

$$Ax = \lambda Bx \quad , \quad (7)$$

where  $A, B \in \mathbb{R}^{n \times n}$ ,  $\lambda \in \mathbb{R}$  (eigenvalue) and  $x \in \mathbb{R}^n$  (eigenvector).  $A$  is symmetric and  $B$  is symmetric positive-definite. In the methodology described in previous sections, the matrices corresponding to the eigenvalue problem derived by equations (2a) and (2b) fit the above formulation.

Let us focus on projection methods for the above eigenproblem that are appropriate when the matrices  $A$  and  $B$  are large but their action on a vector is relatively cheap (*e.g.*, they are sparse) and only part of the spectrum is required. Projection methods rely on two main stages: (i) building a subspace basis; and (ii) extracting approximations from that subspace. These computations are carried out iteratively, where the extracted approximations are used to improve the subspace and extract better approximations, until the computed approximations are sufficiently accurate. For background material on projection methods, the reader is referred to [20, 21] and references therein.

##### 4.1. Subspace extraction and expansion

Consider the computation of  $k$  eigenpairs,  $(\lambda_i, x_i)$ ,  $i = 1, \dots, k$ , usually with  $k \ll n$ . The basic principle of projection methods is to find the best approximations to the eigenvectors in a given subspace of small dimension. Let  $V$  be an  $n \times m$  matrix, with  $k \leq m \ll n$ , whose columns  $v_i$  constitute an  $B$ -orthonormal basis of a given subspace  $\mathcal{V}$ , *i.e.*,  $V^T B V = I_m$ , where  $I$  is the identity matrix and  $\text{span}\{v_1, v_2, \dots, v_m\} = \mathcal{V}$ . Then the eigenvalues of the so-called Rayleigh quotient matrix  $H = V^T A V$  approximate eigenvalues of (7). More precisely, if  $H y_i = \theta_i y_i$  then the eigenpair approximations are  $\tilde{\lambda}_i = \theta_i$  and  $\tilde{x}_i = V y_i$ . These approximate

eigenvectors belong to subspace  $\mathcal{V}$  and are the best possible approximations in that subspace.

The method outlined above is referred to as the Rayleigh-Ritz procedure, and provides  $m$  Ritz approximations  $(\tilde{\lambda}_i, \tilde{x}_i)$ . This routine can be viewed as an orthogonal projection satisfying the Galerkin condition that the residuals are orthogonal to the subspace,

$$r_i = A\tilde{x}_i - \tilde{\lambda}_i B\tilde{x}_i \perp \mathcal{V} \quad . \quad (8)$$

The quality of the eigenpair approximation  $(\tilde{\lambda}_i, \tilde{x}_i)$  depends on how the subspace  $\mathcal{V}$  is built. A popular choice is to use Krylov subspaces of increasing dimension, since they contain increasingly good approximations of eigenvectors of extreme eigenvalues. Computing a  $B$ -orthogonal basis of the Krylov subspace associated with matrix  $C = B^{-1}A$  and a given initial vector  $v_1$ ,  $\mathcal{K}_m(Cv_1) = \text{span}\{v_1, Cv_1, C^2v_1, \dots, C^{m-1}v_1\}$ , can be done with the Lanczos or Arnoldi algorithms, which basically consist in  $B$ -orthogonalizing the vector  $Cv_k$  with respect to the previous basis vectors.

An alternative to Krylov methods, which require the accurate factorisation of  $B$ , are Davidson-type methods such as Generalized Davidson (GD) or Jacobi-Davidson (JD). The main characteristic of this class of methods is that they expand the subspace with a so-called correction vector  $t$ , which is computed from the residual vector  $r$  associated with the most wanted eigenpair with the aim of improving it further. This new vector can be computed by simply preconditioning the residual,

$$t = K^{-1}r \quad , \quad (9)$$

as in the GD method [22], or by solving the correction equation,

$$(I - B\tilde{x}\tilde{x}^*)(A - \tilde{\lambda}B)(I - \tilde{x}\tilde{x}^*B)t = -r \quad , \quad (10)$$

as in the JD method [23]. As in (9), a preconditioner  $K$  can also be introduced in (10).  $K$  is usually a rough approximation of  $A - \tau B$  in case of seeking the closest eigenvalues to  $\tau$ , or of  $B$  in case of seeking eigenvalues in the periphery of the spectrum.

In many contexts, especially when computing interior eigenvalues in difficult non-symmetric problems, JD can be the most competitive method, and even more so if a good, cheap preconditioner is available. However, for symmetric problems where only extreme eigenvalues are to be sought, the high cost of the correction equation does not usually compensate. In this paper, we focus on the Olsen expansion [24], a variant of GD that constrains the correction  $t$  in order to be

$B$ -orthogonal against the approximate eigenvector  $\tilde{x}$ ,

$$\left( I - \frac{K^{-1}B\tilde{x}\tilde{x}^*}{\tilde{x}^*K^{-1}B\tilde{x}} \right) K^{-1}r \quad . \quad (11)$$

As a result, Olsen's variant avoids the stagnation occurring when the computed corrections are almost collinear to  $\tilde{x}$ , that can be produced by the standard GD expansion when  $K$  is too close to  $A - \tilde{\lambda}_i B$ .

#### 4.2. Subspace Recycling

A drawback of Krylov methods is that they start building the subspace from a single vector  $v_1$ . If one has an *a priori* knowledge of a rough approximation of the solution, *e.g.*, from a previous iteration of the self-consistent loop, then this knowledge cannot be exploited. In contrast, Davidson methods can benefit from using a rough approximation of the solution as initial guess, and thus, improve convergence considerably with the corresponding reduction of the overall cost.

In our application, the eigenvalue computation at a given self-consistent iteration is started with an initial guess  $V^0$  coming from the solution computed in the previous iteration. In order to further improve the convergence, this initial subspace can be enriched with a block Krylov subspace generated by the operator  $K^{-1}(A - \tau B)$  acting on the initial guess.

#### 4.3. Numerical experiments with the combined scheme

The following discussion evaluates the performance of our implementation in terms of energy convergence and scalability to a large number of processes. The application code was executed on Tirant, a machine consisting of 256 JS20 blade computing nodes, each with two 64-bit 2.2 GHz PowerPC g 970+ processors. The Poisson problem is solved iteratively with PCG accelerated with a Block Jacobi preconditioner with an incomplete Cholesky factorisation (and a fill level equal to 5). From the Schrödinger problem as many eigenpairs as electrons are in the atom are computed approximately, using the Olsen variant described previously and configured as follows. The search subspace is initialised with 10 vectors and bounded to 18 vectors. When the subspace is full, the solver restarts with 8 vectors. The preconditioner used is the Block Jacobi approximation of  $A - \tau B$  also with an incomplete Cholesky factorisation (level of fill equal to 5). The initial target value is  $\tau = -1$ , and subsequent targets are obtained from the previously converged eigenvalues.

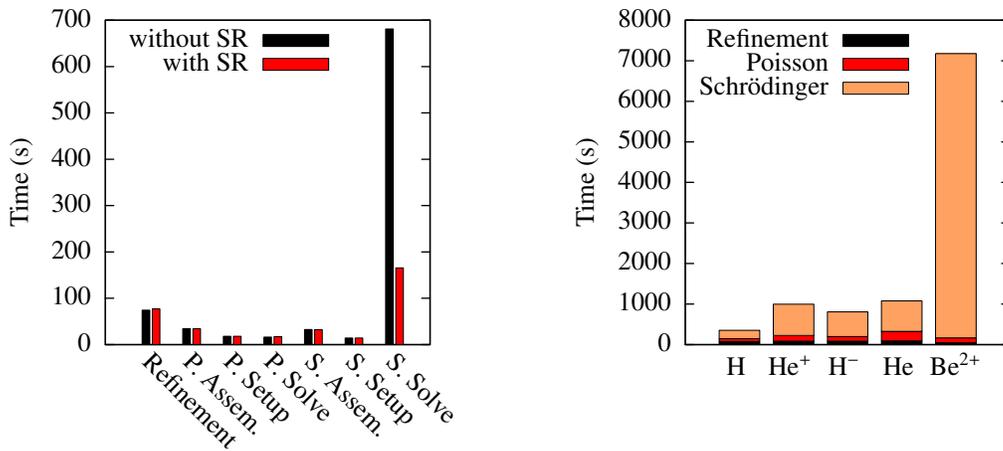


Figure 4: Wall-clock times with 16 processes computing the electronic configuration of H without and with Subspace Recycling (SR) (left), and for various atoms with SR (right). The time is split between the refinement process and the solution of the Poisson (P) and the Schrödinger (S) problems.

The methods of grid adaptivity introduced in Section 3 and eigenvector recycling discussed above are used to accelerate the computation. First, the potential-based error estimate generates an initial grid through a series of  $A$ -cycles. Second, the Kelly error estimate is applied in  $K$ -cycles to the solution of the Poisson equation, where, for each  $K$  the Kohn-Sham equation set is solved to the desired tolerance. A sample computation is given in Table 1, where the number of degrees-of-freedom and number of iterations spent by the linear solver and the eigensolver are shown along with the respective convergence of the Hartree energy. It is clearly seen that the Hartree energy improves with use of the Kelly error estimate, and in any given fixed Kelly phase, the energy converges with few self-consistent iterations due to recycling the eigenvectors computed in the previous cycle. The observed increase in the number of iterations between Kelly cycles is with respect to an increase in the number of degrees-of-freedom.

Fig. 4 compares the time spent by the different parts of the solution process for the computation laid out in Table 1. This highlights the effectiveness of the time spent by the GD solver (left) and an evaluation of the time spent in the various solvers when applied to a sample of light atoms (right). Initialising the solver subspace with the eigenvectors computed in the previous iteration (subspace recycling) is found to have a positive impact on the solution time for the Schrödinger part; that is, more than four times faster with eigenvector recycling. The time

Table 1: A sample computation making use of grid refinement  $A$  and  $K$ . The number of active cells (Act. cells), the number of degrees-of-freedom (Dofs), the number of iterations spent by the linear system solver (Poi.) and the eigenproblem solver (Schr.), the Hartree energy ( $E^i$ ), and the energy difference ( $\Delta E = |E^{i-1} - E^i|$ ). Note, that no convergence data is given for the adaptive cycles ( $A$ ) based on the projected potential, as it was not required to solve the generalized eigenvalue problem in those stages of the computation.

Phase	Act. cells	Dofs	Poi.	Schr.	$E$	$\Delta E$
$A = 4$	116012	148749	56	36	-0.497505	
$K = 1$	207152	241837	53	54		$8.27 \cdot 10^{-4}$
			53	1		$1.34 \cdot 10^{-10}$
			53	1		$1.00 \cdot 10^{-12}$
			53	1	-0.498332	$< 10^{-13}$
$K = 2$	370175	415072	43	69		$1.57 \cdot 10^{-4}$
			43	1		$3.20 \cdot 10^{-11}$
			43	1		$6.00 \cdot 10^{-12}$
			43	1	-0.498489	$1.00 \cdot 10^{-13}$
$K = 3$	660052	713196	76	85		$7.45 \cdot 10^{-4}$
			76	1		$5.10 \cdot 10^{-11}$
			76	1		$1.10 \cdot 10^{-11}$
			76	1	-0.498563	$3.00 \cdot 10^{-12}$
$K = 4$	1175609	1246510	76	104		$5.26 \cdot 10^{-5}$
			76	1		$5.80 \cdot 10^{-11}$
			76	1		$9.00 \cdot 10^{-12}$
			76	1	-0.498618	$3.00 \cdot 10^{-12}$
$K = 5$	2091727	2194572	40	129		$3.29 \cdot 10^{-5}$
			40	1		$1.16 \cdot 10^{-10}$
			40	1		$1.40 \cdot 10^{-11}$
			40	1	-0.498648	$2.00 \cdot 10^{-12}$
$K = 6$	3744049	3926380	1	159		$1.59 \cdot 10^{-5}$
			1	1		$2.34 \cdot 10^{-10}$
			1	1		$3.30 \cdot 10^{-11}$
			1	1	-0.498667	$9.00 \cdot 10^{-12}$

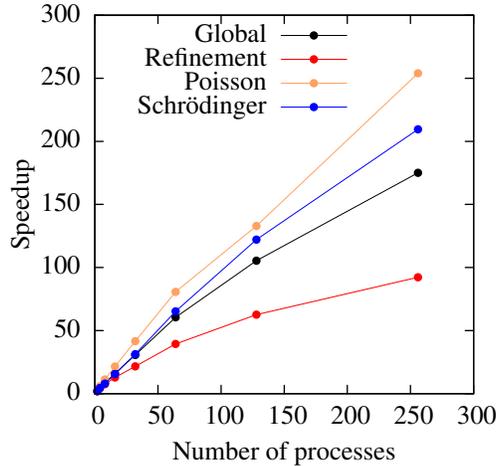


Figure 5: Speedup computing the electronic configuration of H.

taken for the other parts of the solution process are not affected by eigenvector recycling, which is an intuitive result. The crucial observation here, is that recycling the subspace in the GD solver puts the time taken for solution of the eigenvalue problem on a par with the other parts of the solution process, *i.e.*, significantly more so than without subspace recycling. The proposed scheme has been tested successfully to compute the electronic configuration of atoms and ions with up to 4 protons; see right panel of Fig. 4. For heavier atoms, difficulties arise in terms of convergence, so other preconditioning strategies should be employed. These subtleties can be addressed in an implementation that targets large or multi-atom systems.

Fig. 5 gives the comparative scaling of the time spent by different parts of the solution process with respect to the number of processes. The scalability of four main components of the application are considered: (i) global refinement, (ii) adaptive refinement, and the solution to (iii) Poisson’s and (iv) Schrödinger’s equations. It turns out that the refinement process is the least scalable part, being far from the rest of the components that exhibit good performance close to linear scaling. Because of that, the weight of the refinement in the total time becomes larger when increasing the number of processes (19% with 4 processes, 21% with 16 processes, and up to 31% with 256 processes). Despite that, the whole application presents competitive scalability results up to 256 processes.

## 5. Summary and outlook

In this contribution, a computational method for DFT computations was developed that makes use of grid adaptivity and eigenvector recycling. The method of grid adaptivity presented here uses a representation of the underlying Coulomb potential to locally refine the grid in places where the potential is badly represented. In successive self-consistent iterations the solution vectors from the previous iteration was given to the GD solver as a starting space for computation of the current configuration (subspace recycling). The overall result of employing both techniques to the Kohn-Sham equations of DFT, was to improve the converge of Hartree energy of the system with a reduction in the computational overhead. The direction of research taken here is similar to that of Napoli *et al.* [4], in that methods have been introduced that systematically improve on the algorithmic realisation of the model on which computational simulations are based. The schemes used here are applicable to a wide range of problems and it is not restricted to finite element methods. This is in contrast to many recent studies that seek specific enhancements or specialisations to better the quality of the solution, the effect of which is to narrow the range of applicability. For our implementation to go further beyond conventional methods of atomic simulations, a need is identified for its optimisation and deployment to heavy and multi-atom systems.

*Acknowledgements.* The anonymous referees whose comments improved the presentation of this work are gratefully acknowledged. The work was supported by the Polish Ministry of Science and Higher Education NN519 402837 and by the Spanish Ministry of Science and Innovation TIN2009-07519. The resources provided by the Barcelona Supercomputing Center are also acknowledged.

## References

- [1] P. Hohenberg, W. Kohn, Phys. Rev. 136 (1964) B864–B871.
- [2] W. Kohn, L. J. Sham, Phys. Rev. 140 (1965) A1133–A1138.
- [3] T. L. Beck, Rev. Mod. Phys. 72 (2000) 1041.
- [4] E. D. Napoli, S. Blügel, P. Bientinesi, Comput. Phys. Commun. 183 (2012) 1674.
- [5] M. N. Guimarães, F. V. Prudente, J. Phys. B: At. Mol. Opt. Phys. 38 (2005) 2811.

- [6] J. E. Pask, P. A. Sterne, *Modelling Simul. Mater. Sci. Eng.* 13 (2005) R71.
- [7] S. Yamakawa, S. Hyodo, *Phys. Rev. B* 71 (2005) 035113.
- [8] J.-L. Fattebert, R. D. Hornung, A. M. Wissink, *J. Comp. Phys.* 223 (2007) 759–773.
- [9] E. Tsuchida, M. Tsukada, *Phys. Rev. B* 52 (1995) 5573.
- [10] E. Tsuchida, M. Tsukada, *Phys. Rev. B* 54 (1996) 7602.
- [11] C. Vömel, S. Z. Tomov, O. A. Marques, A. Canning, L.-W. Wang, J. J. Dongarra, *J. Comput. Phys.* 227 (2008) 7113–7124.
- [12] T. D. Young, R. Armiento, *CTP* 53 (2010) 1017.
- [13] W. Bangerth, G. Kanschat, *deal.II Differential Equations Analysis Library*, Technical reference, 2011. <http://www.dealii.org>.
- [14] W. Bangerth, C. Burstedde, T. Heister, M. Kronbichler, *ACM Trans. Math. Softw.* 38 (2011) 14.
- [15] V. Hernandez, J. E. Roman, V. Vidal, *ACM Trans. Math. Softw.* 31 (2005) 351–362.
- [16] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, B. Smith, H. Zhang, *PETSc Users Manual*, Technical Report ANL-95/11 - Revision 3.2, Argonne National Laboratory, 2011.
- [17] T. D. Young, namespace `ewalena`, *Documentation and technical reference*, Institute of Fundamental Technological Research PAN, pre-alpha edition, 2011. <http://www.ewalena.sourceforge.net>.
- [18] D. W. Kelly, J. P. de S. R. Gago, O. Zienkiewicz, I. Babuska, *Int. J. Num. Math. Engrg* 19 (1983) 1621.
- [19] M. Ainsworth, J. T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*, John Wiley and Sons, 2000.
- [20] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst (Eds.), *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2000.

- [21] E. Romero, J. E. Roman, A parallel implementation of Davidson methods for large-scale eigenvalue problems in SLEPc, 2012. Submitted.
- [22] R. B. Morgan, D. S. Scott, *SIAM J. Sci. Statist. Comput.* 7 (1986) 817–825.
- [23] G. L. G. Sleijpen, H. A. van der Vorst, *SIAM Rev.* 42 (2000) 267–293.
- [24] J. Olsen, P. Jørgensen, J. Simons, *Chem. Phys. Lett.* 169 (1990) 463–472.