Improved Unconstrained Energy Functional Method for Eigensolvers in Electronic Structure Calculations

Mauro Del Ben
Lawrence Berkeley National Laboratory
mdelben@lbl.gov

Osni Marques
Lawrence Berkeley National Laboratory
oamarques@lbl.gov

Andrew Canning
Lawrence Berkeley National Laboratory
acanning@lbl.gov

ABSTRACT

This paper reports on the performance of a preconditioned conjugate gradient based iterative eigensolver using an unconstrained energy functional minimization scheme. In contrast to standard implementations, this scheme avoids an explicit reorthogonalization of the trial eigenvectors and becomes an attractive alternative for the solution of very large problems. The unconstrained formulation is implemented in the first-principles materials and chemistry CP2K code, which performs electronic structure calculations based on a density functional theory approximation to the solution of the many-body Schrödinger equation. We study the convergence of the unconstrained formulation, as well as its parallel scaling, on a Cray XC40 at the National Energy Research Scientific Computing Center (NERSC). The systems we use in our studies are bulk liquid water, a supramolecular catalyst gold(III)-complex, a bilayer of MoS_2-WSe_2 and a divacancy point defect in silicon, with the number of atoms ranging from 2,247 to 12,288. We show that the unconstrained formulation with an appropriate preconditioner has good convergence properties and scales well to 230k cores, roughly 38% of the full machine.

ACM Reference format:
Mauro Del Ben, Osni Marques, and Andrew Canning. 2019. Improved Unconstrained Energy Functional Method for Eigensolvers in Electronic Structure Calculations . In *Proceedings of 48th International Conference on Parallel Processing, Kyoto, Japan, August 5–8, 2019 (ICPP 2019),* 11 pages.
DOI: 10.1145/3337821.3337914

1 INTRODUCTION

Many scientific applications require the solution of eigenvalue problems. For applications where some small percentage of the eigenpairs is required rather than the full spectrum, iterative eigensolvers are typically used. This is because the computational cost of a direct solver scales as the cube of the matrix dimension, while iterative solvers scale as the square of the number of required eigenpairs times the matrix dimension. The prefactor for the scaling of the iterative solver is much larger than for the direct solver. The crossover point for which method is the fastest varies with the class of system under study but is typically in the regime of about 10%. Electronic structure calculations in materials science and chemistry codes based on some approximate solution to the Schrödinger equation is an example of this class of problems, where typically a small percentage of the lowest eigenpairs is required. Iterative methods such as Conjugate Gradient (CG) and Davidson are often used for this class of problem. They can scale well on large parallel computers for large problems because most of the operations in the algorithms are large matrix multiplies, where at least one dimension of the matrices has the dimension of the full matrix being diagonalized. These types of operation can be implemented efficiently with parallel BLAS routines. The commonly used iterative solvers for materials and chemistry codes are often referred to as constrained energy functional approaches, as the constraint of orthogonality on the eigenvectors requires some form of reorthogonalization of the trial vectors. The function being minimized with this constraint is related to the energy of the physical system being modeled. The reorthogonalization step involves operations on the so-called small subspace matrix, which has the dimension of the number of required eigenpairs. In some applications this number may be as low as a few percent of the full matrix. This can result in the reorthogonalization step having poor parallel scaling compared to the other steps in the iterative solver, limiting the overall parallel scaling of the method. The use of unconstrained energy functionals avoids the explicit reorthogonalization step; therefore, it has the potential for better parallel scaling than both direct eigensolvers and constrained energy functional iterative solvers. In this paper, we present a study of unconstrained energy functional iterative eigensolvers in the context of the commonly used density functional theory (DFT) approach to solving the Schrödinger equation for materials and chemistry codes.

2 FORMALISM

For electronic structure calculations in materials science and chemistry, first-principles methods based on Density Functional Theory (DFT) in the Kohn-Sham (KS) formalism [1] are the most widely used approaches due to their computational efficiency and favorable scaling with system size. It is now possible to routinely model systems of thousands of atoms on existing high performance computers. The KS single particle eigenfunction equations are usually written in atomic units as

$$\hat{\mathbf{H}}\psi_i(\mathbf{r}) = \left[-\frac{1}{2}\nabla^2 + \mathbf{V}\right]\psi_i(\mathbf{r}) = \varepsilon_i\psi_i(\mathbf{r}) \tag{1}$$

where $\psi_i(\mathbf{r})$ are the wavefunctions for each electron in the system and ε_i is the energy of the electron. The probability of finding the i'th electron at position $\mathbf{r}$ is given by $\psi_i(\mathbf{r})\psi_i^*(\mathbf{r})$, where $*$ denotes the complex conjugate. The first term in the Hamiltonian operator

$\hat{\mathbf{H}}$ is the kinetic energy operator, and **V** is the potential. In the most commonly used local density approximation (LDA) using pseudopotentials to replace the nuclei and core electrons, **V** is given by

$$\mathbf{V} = \sum_{\mathbf{R_I}} v_{ion}(\mathbf{r} - \mathbf{R_I}) + \int \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \mathbf{d}^3\mathbf{r}' + \mu_{\mathbf{xc}}(\rho(\mathbf{r})) \qquad (2)$$

where $v_{ion}(\mathbf{r} - \mathbf{R_I})$ is the ionic pseudopotential of ion I, $\mathbf{R_I}$ is the position of ion I, $\rho(\mathbf{r}) = \sum_{\mathbf{i}} \psi_{\mathbf{i}}(\mathbf{r})\psi_{\mathbf{i}}^*(\mathbf{r})$ is the total charge density, and $\mu_{xc}(\rho(\mathbf{r}))$ is the LDA exchange-correlation potential. In our studies, we have used the generalized gradient approximation PBE functional [2], which is a commonly used extension to the LDA approximation giving more accurate results for many systems. These equations represent the electronic interactions in a system such as a periodic crystal, molecule or nanostruture. The first term in the potential is the interaction between the valence (bonding) electrons and the ions. The second term is the Hartree potential for the electron-electron interaction, the exchange-correlation potential comes from the reduction of the many body Schrödinger equation to the single body form. In the formalism of quantum mechanics the solution of these equations gives the wavefunction of each electron and then all observables of the system are represented by Hermitian operators, which act on the wavefunctions. The eigenvalues of the operator are the values the observables take so, for example, the Hamiltonian $\hat{\mathbf{H}}$ is the energy operator.

Since the potential **V** in Eq. 2 depends on the total charge density, which in turn depends on the wavefunctions, this eigenvalue problem is often referred to as being non-linear. The most commonly used approach in electronic structure calculations for solving this non-linear problem is the self-consistent field (SCF) method, where the problem is linearized in an inner loop by fixing the charge density, which is then updated at each step in the loop until convergence of the charge density and potential field. This procedure is illustrated in Fig. 1. One starts from an initial guess for the wavefunctions (often atomic orbitals or random) in the first step, from which a total charge density and Hamiltonian can be calculated.

In order to solve Eq. 1 in the SCF loop the wavefunctions $\psi_i(\mathbf{r})$ are usually expanded in some basis set or discretized on a real space grid. The most commonly used basis sets are plane waves (Fourier expansion) and atom centered Gaussian functions but other basis

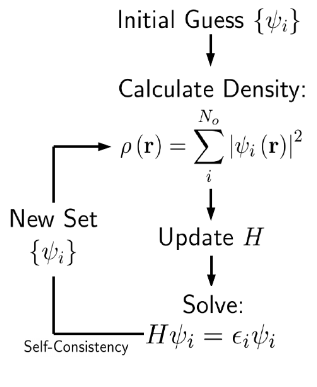

Initial Guess $\{\psi_i\}$
Calculate Density:
$\rho(\mathbf{r}) = \sum_i^{N_o} |\psi_i(\mathbf{r})|^2$
New Set $\{\psi_i\}$
Update H
Solve:
$H\psi_i = \epsilon_i\psi_i$
Self-Consistency

Figure 1: Schematic of the self consistent field (SCF) procedure as usually implemented in electronic structure codes.

such as numerical orbitals and wavelets are also used. Introducing a set of basis functions $\{\phi_\alpha(\mathbf{r})\}$ of size N_b for the wavefunctions $\psi_i(\mathbf{r})$ in Eq. 1 we can now write the wavefunctions in terms of coefficients of each basis function,

$$\psi_i(\mathbf{r}) = \sum_\alpha \mathrm{C}_{\alpha\mathrm{i}}\phi_\alpha(\mathbf{r}), \qquad (3)$$

and the Hamiltonian as an $N_b \times N_b$ matrix by calculating integrals of the form $H_{\alpha\beta} = \int \phi_\alpha(\mathbf{r})\mathbf{H}\phi_\beta(\mathbf{r})\mathbf{dr}$. Here and in what follows we use the following convention: i, j, k indices identify eigenvectors, α, β, γ indices identify basis functions, and N_o represents the number of the lowest eigenvectors of interest, corresponding to the occupied electronic states necessary to build the total electronic charge density $\rho(\mathbf{r})$. In the most general case, such as Gaussian basis as used in CP2K, the basis functions $\{\phi_\alpha\}$ are non-orthogonal, and the overlap matrix for the basis set $S_{\alpha\beta} = \int \phi_\alpha(\mathbf{r})\phi_\beta(\mathbf{r})\mathbf{dr}$ is different from the identity ($\mathbf{S} = \mathbf{I}$ for an orthogonal basis such as plane waves). This requires, in matrix form, for the inner part of the SCF loop (see Fig. 1), the solution of the associated generalized eigenvalue problem

$$\mathbf{HC} = \mathbf{SCE} \qquad (4)$$

where **C** is the matrix of generalized eigenvectors $\{\mathrm{C}_i\}_{i=1...N_o}$ (coefficients of the basis functions) obeying the orthogonality condition $\mathbf{C}^T\mathbf{SC} = \mathbf{I}$. **E** is the diagonal matrix of eigenvalues. Again, for an orthonormal basis $\mathbf{S} = \mathbf{I}$ and Eq. 4 reduces to a standard eigenvalue problem. For the ground state, the solution of the system only requires the lowest eigenvalues and eigenvectors of **H** corresponding to the wavefunction of each electron in the system, which are also used to construct the charge density in the SCF loop.

In the case of a Gaussian basis, historically the diagonalization of the Hamiltonian (solution of the eigenvalue problem) in the SCF loop in Fig. 1 was performed with direct, dense eigenvalue solvers as implemented e.g. in ScaLAPACK [3]. Typically, direct eigenvalue solvers involve the reduction of a dense matrix to tridiagonal form, whose eigenvalues and eigenvectors are easier to compute, followed by a backtransformation of the eigenvectors. On modern large scale parallel computers, those operations can lead to parallel scalability limitations, which have been addressed by ingenious redesigns of the reduction and backtransformation phases [4, 5]. Efficient implementations are available in the EigenExa [6, 7] and ELPA [8] packages. As previously mentioned, for electronic structure problems with larger basis set sizes, the percentage of required eigenpairs is relatively low. For example, in the case of the commonly used plane wave basis, only a few percent of the eigenpairs is required and historically iterative solvers have been used for this basis.

Iterative solvers based on Lanczos or Arnoldi algorithms have been used in some electronic structure applications [9–11]. However, their use has been limited since iterative solvers such as CG and Davidson can take better advantage of good initial guesses for the eigenvectors. In the SCF formulation of the problem we always, at each SCF step, have a good initial guess for the eigenvectors from the previous SCF step. A block version of Lanczos (or Arnoldi) would be required to take advantage of these guesses, but with additional overheads (e.g. for maintaining the orthogonality of the Lanczos vectors). In a CG formulation, the eigenvalue problem is usually cast in terms of a minimization problem subject to the orthogonality constraint,

$$min \text{ Tr}\left[\mathbf{C}^{\mathrm{T}}\mathbf{H}\mathbf{C}\right], \quad \mathbf{C}^{\mathrm{T}}\mathbf{S}\mathbf{C} = \mathbf{I}. \tag{5}$$

$\mathbf{C}$ is now the matrix of trial generalized eigenvectors with dimension $N_b \times N_o$, where N_b is the size of the basis set and N_o is the number of electronic wavefunctions, which are often referred to as bands, states or orbitals. At the solution of the minimization of this function, we obtain the total electronic energy of the system $\sum_i \varepsilon_i$ (see Eq. 1). This approach is therefore referred to as minimization of the energy functional in the electronic structure community, and it is used in various packages intended for the computation of materials and chemical properties [12, 13]. See [10] for a (non-exhaustive) list of minimization algorithms used in electronic structure packages. In general, an important feature of iterative solvers is that they only require the action of an operator over a vector, i.e. matrix vector multiplications, where efficient storage and computational strategies for sparse matrices can also be employed [14, 15]. In particular, the techniques discussed in [14] are used in the CP2K code [16]. For efficiency reasons and to increase parallelism, iterative solvers are usually implemented in block form: the solvers iterate on the set of trial vectors $\{\mathbf{C_i}\}$ at the same time, allowing for the computation of a set of eigenpairs simultaneously (in contrast to a one vector at a time, often called *band-by-band* formulation) [17]. The trial vectors can be kept orthogonal through the diagonalization of the subspace $N_b \times N_b$ eigenvector overlap matrix $\mathbf{C}^{\mathbf{T}}\mathbf{SC}$ for non-orthogonal basis and $\mathbf{C}^{\mathbf{T}}\mathbf{C}$ for orthogonal basis, which can be done with ScaLAPACK. This can alternatively be achieved through Gram-Schimdt, Cholesky or QR methods [18, 19]. In some cases, these methods have improved parallel scaling over the use of direct diagonalization. However, they are all limited in their scaling to large core counts because of global communication costs on operations on a relatively small sub-space matrix, compared to operations on the full Hamiltonian matrix.

Various other techniques have been proposed to deal with the orthogonality of the trial vectors. In [20], for example, the authors discuss a strategy based on Pulay's DIIS [21], an extrapolation technique that in principle does not require reorthogonalizations. However, those authors observe that reorthogonalizations is required for good convergence and stability of the strategy. Spectrum slicing and polynomial filters have been proposed in [22, 23] to reduce the costs of reorthogonalization. Although promising, spectrum slicing may still suffer from scaling issues depending on the number of eigenvalues that lie in a given slice. In turn, polynomial filters can be expensive as they may require an additional and large number of matrix-vector products.

The unconstrained energy functional approach for electronic structure calculations offers the possibility of eliminating explicit diagonalization and operations on the small overlap matrix. In this paper, we are studying this method in the context of electronic structure problems, but we note that it can be applied to the determination of a desired number of the lowest eigenvalues and corresponding eigenvectors of any symmetric or Hermitian matrix. The unconstrained approach used for electronic structure calculations [24] is based on the relaxation of the orthogonality constraint in Eq. 5 by means of a set of vectors $\mathbf{X}$ that spans the same subspace as $\mathbf{C}$ but is not required to be orthogonal. This transformation is given by

$$\mathbf{C} = \mathbf{X}\mathcal{S}^{-\frac{1}{2}}, \quad \mathcal{S} = \mathbf{X}^{\mathrm{T}}\mathbf{X}. \tag{6}$$

Note that we are using $\mathbf{S}$ for the overlap matrix (size $N_b \times N_b$) of the basis functions and $\mathcal{S}$ for the overlap matrix (size $N_o \times N_o$) of the trial vectors $\mathbf{X}$, where N_o is much smaller than N_b. The minimization problem of Eq. 5 can now be rewritten as

$$E[\mathbf{X}] = \mathbf{min} \text{ Tr}\left[\mathcal{S}^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{H}\mathbf{X}\right], \tag{7}$$

relaxing the orthogonality constraint and introducing the so-called unconstrained energy functional $E\,[\mathbf{X}]$. Near the solution, $\mathcal{S}^{-1}$ is close to the identity matrix $\mathbf{I}$, and most implementations use the first order expansion $(2\mathbf{I} - \mathcal{S})$ to approximate it, avoiding the matrix inversion of $\mathcal{S}$. Approximations to higher order have also been studied [25] for a plane wave basis code, although no major benefit has been observed over the first order approximation. [26] introduces the *orbital minimization method* (OMM), an unconstrained approach that uses a basis of numerical atomic orbitals, which leads to sparse operators. This method has been implemented in the SIESTA package [27]. Therefore, the unconstrained approaches based on expansions of $\mathcal{S}^{-1}$ never involve operations solely on the small $\mathcal{S}$ matrix, like in the constrained approaches. This is a key ingredient in unconstrained methods to improved parallel scaling for large systems by avoiding the inversion of a small matrix. A caveat to this is that the removal of the constraint, and a different functional to minimize, changes the convergence properties. One of the main focuses of our studies was therefore to look at the convergence properties of the unconstrained functional and develop new preconditioners to improve the convergence.

While our studies have focused on the use of the unconstrained approach applied to the standard cubic scaling DFT method, the approach has been mostly used for the so-called $O(N)$ linear scaling electronic structure methods, where N is the number of atoms in the system. These methods typically involve some constraint of spatial locality for the wavefunctions or charge density. The unconstrained approach is favorable for $O(N)$ methods because its soft orthogonality constraint allows the spatial locality to be maintained during the iteration procedure. An explicit orthogonalization would lose the locality, as the true eigenvectors of the Hamiltonian have exponentially vanishing tails even in the most localized cases. A review of $O(N)$ methods including the use of unconstrained functionals can be found in reference [28]. The SIESTA package [27] also provides an unconstrained $O(N)$ implementation with spatially localized wavefunctions expanded in atomic orbitals.

We note that a set of the aforementioned algorithms have been incorporated into ELSI [29], a software infrastructure that provides an interface to multiple strategies for solving eigenvalue problems that arise in electronic structure calculations.

In the next section of this paper we discuss the methods and algorithms as implemented for our studies in the CP2K first-principles code [16]. To our knowledge, this is the first time the unconstrained approach has been implemented in a Gaussian basis electronic structure code. In particular, we focus on the new preconditioners developed for the unconstrained CG eigensolver. Section 4 presents the computational details of our performance runs. Section 5 presents convergence and parallel scaling results on a Cray XC40. One of the main goals of this work is to revisit these unconstrained functional approaches in the context of very large scale electronic structure calculations on modern large scale parallel many-core computers, to study their parallel scaling properties in comparison to other commonly used approaches such as direct solvers. The final section in the paper presents conclusions based on our work.

3 METHOD AND ALGORITHMS

CP2K [16] is a massively parallel computer program for first-principles quantum chemistry and materials simulations using a DFT approach. DFT is implemented in the framework of the Gaussian and plane wave (GPW) approach [30, 31]. The GPW method makes use of a dual basis representation for the charge density and wavefunctions, i.e. delocalized plane waves and localized atomic centered Gaussian functions. This allows for an efficient construction of the Hamiltonian matrix in linear scaling time with respect to system size. For this reason, the diagonalization of the Hamiltonian within the SCF procedure becomes the calculation bottleneck.

In this work, the unconstrained functional we have implemented in the CP2K code corresponds to Eq. 7. The inverse trial vector overlap matrix $\mathcal{S}^{-1}$ is then expanded to first order about the identity matrix, i.e. $\mathcal{S}^{-1} = (2\mathbf{I} - \mathcal{S})$, which becomes, writing it in terms of the basis function overlap matrix S where $\mathcal{S} = \mathbf{X}^{\mathrm{T}}\mathbf{S}\mathbf{X}$,

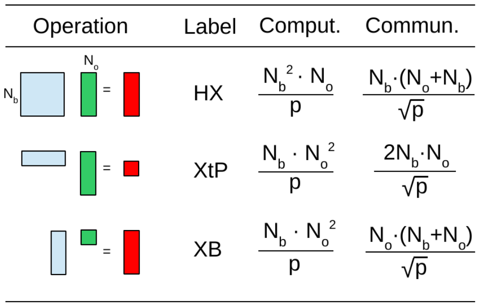
$$E[\mathbf{X}] = \mathrm{Tr}\left[\left(2\mathbf{I} - \mathbf{X}^T\mathbf{S}\mathbf{X}\right)\mathbf{X}^T\mathbf{H}\mathbf{X}\right]. \tag{8}$$

It has been shown [32–35] that the minimization of this functional with respect to $\mathbf{X}$: a) converges to the same energy as that of the correct functional of Eq. 7, the sum of the first N_o eigenvalues of Eq. 4, b) at convergence the subspace spanned by $\mathbf{X}$ is the same as that spanned by the true generalized eigenvectors $\mathbf{C}$, and c) by minimization $\mathbf{X}$ self-orthogonalizes, i.e. at the minimum $\mathbf{X}^T\mathbf{S}\mathbf{X} = \mathbf{I}$. The only condition to ensure convergence is that all N_o eigenvalues calculated are all negative. This is realized by a rigid shift of the eigenspectrum of $\mathbf{H}$ by a proper scalar quantity η employing the transformation $\mathbf{H} \longrightarrow \mathbf{H} - \eta \cdot \mathbf{S}$. The gradient of the unconstrained functional of Eq. 8 reads :

$$\mathbf{G} = 4\mathbf{H}\mathbf{X} - 2\mathbf{S}\mathbf{X}\mathcal{H} - 2\mathbf{H}\mathbf{X}\mathcal{S} \tag{9}$$

where $\mathcal{H} = \mathbf{X}^{\mathrm{T}}\mathbf{H}\mathbf{X}$ is of the same dimensions as $\mathcal{S}$, i.e. $N_o \times N_o$. This gradient can be used directly to carry out the unconstrained functional minimization. In this work, we use a preconditioned conjugate gradient (PCG) procedure to minimize the function. By using an appropriate preconditioner (discussed later in this section) one obtains the search direction $\mathbf{D}$ by employing one of the many possible formulae (Polak-Ribiére in our case). The improved trial vectors to be used in the next PCG iteration are then obtained as $\mathbf{X}^{\mathrm{New}} = \mathbf{X} + \alpha\mathbf{D}$, with α optimally chosen by a line search procedure to give the maximum functional energy minimization. One of the advantages of the unconstrained functional of Eq. 8 is that, due to its quadratic form, the optimal α can be calculated analytically by solving a 4th order polynomial [26], which requires only a little extra computation to get the polynomial coefficients.

The approach described here so far requires only matrix multiplication operations that, combined with the development of reduced-communication / communication avoiding-parallel algorithms and a possible hybrid GPU-CPU implementation, give a promising avenue in terms or computational efficiency. A summary of the basic



Operation	Label	Comput.	Commun.
N_b, N_o, =	HX	$\frac{N_b^2 \cdot N_o}{p}$	$\frac{N_b \cdot (N_o + N_b)}{\sqrt{p}}$
=	XtP	$\frac{N_b \cdot N_o^2}{p}$	$\frac{2N_b \cdot N_o}{\sqrt{p}}$
=	XB	$\frac{N_b \cdot N_o^2}{p}$	$\frac{N_o \cdot (N_b + N_o)}{\sqrt{p}}$

Figure 2: Matrix multiplications (in parallel) involved in the unconstrained functional minimization scheme. The scaling of the computational cost (floating point operations) and communication volume (real or complex numbers communicated) per processor p, are reported assuming dense linear algebra and Cannon's algorithm [36]. In the table, N_b is the basis set size and N_o is the number of required eigenpairs.

parallel matrix multiplication operations necessary to implement the unconstrained functional minimization method is given in Fig. 2. It is shown that the asymptotic scaling of the method with system size is cubic, since both N_o and N_b grow linearly with system size. The main point here is that all the matrix multiplications involve matrices where at least one of the matrices has the large dimension N_b of the basis set size, therefore amenable to large scale parallelization. This is in contrast to constrained methods that always contain operations on the small $N_o \times N_o$ matrices, which limits parallel scaling. The caveat here is the convergence of the method. It has already been shown that the absence of a good preconditioner, for localized basis atomic orbitals, makes the unconstrained approach converge very slowly if not even diverge [26]. We address the preconditioning of the method in the next section.

It should be noted, from our studies of the unconstrained approach, that without the development of a tailored preconditioner, truncating the Taylor expansion of $\mathcal{S}$ beyond the first order brings little or no benefit in terms of time to solution compared to traditional methods. For this reason, we have focused on developing an effective preconditioner, leaving the exploration of higher order expansions to later studies.

Pioneering work on preconditioning the conjugate gradient procedure for solving the generalized eigenvalue problem within electronic structure calculations with localized basis function have introduced preconditioners based on the overlap matrix (over basis functions) scaled by the kinetic energy matrix [37, 38]. Such approaches, often used for preconditioning applications based on plane wave basis functions, aim to suppress the high energy components of the eigenspectrum, i.e. suppress eigenstates ϵ_i with $i >> N_o$, which show predominantly a kinetic energy component. Such approaches have also been tested for the unconstrained functional method and more recently for atomic orbital basis functions [26].

The approach we follow here is to obtain a better preconditioner by using information from the Hessian $\mathbf{A}$ of the unconstrained functional approach. $\mathbf{A}$ is formally a square matrix of size $(N_oN_b)\times(N_oN_b)$ obtained as the second derivative of the functional of Eq. 8

with respect to the trial vector coefficients $X_{\alpha i}$. In particular, given $E[\mathbf{X}]$ from Eq. 8 or the gradient $\mathbf{G}$ from Eq. 9, an element $A_{\alpha i,\beta j}$ of $\mathbf{A}$ is given by

$$\begin{aligned} A_{\alpha i,\beta j} =& \frac{\partial^2 E[\mathbf{X}]}{\partial X_{\alpha i}\partial X_{\beta j}} = \frac{\partial G_{\beta j}}{\partial X_{\alpha i}} = \\ & 2H_{\alpha\beta}\left[2\delta_{ij} - \mathcal{S}_{ij}\right] - 2S_{\alpha\beta}\mathcal{H}_{ij} - 2\left[SX\right]_{\beta i}\left[HX\right]_{\alpha j} \\ & - 2\left[HX\right]_{\beta i}\left[SX\right]_{\alpha j} - 2\delta_{ij}\left[HXX^TS + SXX^TH\right]_{\alpha\beta}, \end{aligned} \tag{10}$$

accounting for the fact that in our case $\mathbf{H}$ and $\mathbf{S}$ are real and symmetric (note that $\mathcal{H} = \mathbf{X}^T\mathbf{H}\mathbf{X}$ and $\mathcal{S} = \mathbf{X}^T\mathbf{S}\mathbf{X}$). One can thus use the Hessian for preconditioning the unconstrained functional conjugate gradient procedure, similarly to the case of a Newton or quasi-Newton step. In this case, starting from the gradient $\mathbf{G}$, one should calculate the preconditioned gradient $\mathbf{P}$ as $P_{jb} = \sum_{ia} A^{-1}_{jb,ia}G_{ia}$, implying the inversion of $\mathbf{A}$, which is clearly intractable with a direct solver due the size of the Hessian.

Instead, one can use an iterative procedure to obtain $\mathbf{P}$ as a solution of the associated linear system of equations, that is solving

$$\sum_{ia} A_{jb,ia}P_{ia} = G_{jb} \tag{11}$$

with respect to the P_{ia}. Note that in this case it is not necessary to calculate explicitly the full Hessian matrix, since it is possible to directly compute the matrix-vector product $\mathbf{Q}$ (with elements $Q_{jb} = \sum_{ia} A_{jb,ia}P_{ia}$), according to:

$$\begin{aligned} \mathbf{Q} = \; & 2\mathbf{HP}\left[2\mathbf{I} - \mathcal{S}\right] - 2\mathbf{SP}\mathcal{H} - 2\left[\mathbf{SX}\right]\mathbf{P}^T\left[\mathbf{HX}\right] \\ & - 2\left[\mathbf{HX}\right]\mathbf{P}^T\left[\mathbf{SX}\right] - 2\left[\mathbf{HXX}^T\mathbf{S} + \mathbf{SXX}^T\mathbf{H}\right]\mathbf{P}. \end{aligned} \tag{12}$$

Thus, within the PCG unconstrained functional minimization one can find the preconditioned gradient $\mathbf{P}$ by solving the linear system of Eq. 11 iteratively using an inner PCG procedure. We label this approach as `FullHess-`K`Iter`, where K denotes the number of iterations employed in the inner PCG to solve Eq. 11. Again, all the operations needed to implement this case are matrix multiplications of the kind listed in Fig. 2. There still remains the problem of preconditioning the inner PCG, which we address in the following.

We name the preconditioner for the inner PCG the inner-preconditioner, the simplest of which considered here is given by the inverse of the overlap matrix defined over basis functions, that is $\mathbf{S}^{-1}$. We name this inner-preconditioner `InvOvl`, so when used to solve the inner PCG the overall label reads `FullHess-`K`Iter-InvOvl`, meaning solving the inner PCG for the Hessian by using the $\mathbf{S}^{-1}$ as inner-preconditioner and K iterations. Despite the simplicity of $\mathbf{S}^{-1}$, this preconditioner is robust and cheap to compute. In fact, since S is positive definite, one can use Cholesky factorization for computing and applying the preconditioner, additionally one needs to compute the $\mathbf{S}^{-1}$ only once at the beginning of the SCF loop.

The second inner-preconditioner considered here is obtained as an approximated form of the Hessian given in Eqs. 10 and 12. In particular, we assume that when approaching convergence, the $\mathbf{P}$ matrix elements approach zero, being related to the gradient, decreasing at each iteration. In this way the mixing terms, 3rd and 4th in Eq. 12, are likely becoming less relevant than the others. In addition to this, when approaching convergence $[2\mathbf{I} - \mathcal{S}]$ becomes

Inner Preconditioner	Hessian Solver: None	Hessian Solver: PCG with K Iterations
$\mathbf{S}^{-1}$	`InvOvl`	`FullHess-`K`Iter-InvOvl`
$\tilde{\mathbf{A}}^{-1}$	`AprxHess`	`FullHess-`K`Iter-AprxHess`

Table 1: The different preconditioners used in this paper. PCG is the preconditioned conjugate gradient procedure to solve Eq. 11, with K iterations. None means that the inner-preconditioner is used for preconditioning the unconstrained functional minimization.

closer to the identity matrix due to self-orthogonalization of the trial vectors, such that the first term in Eq. 12 tends to $\mathbf{HP}$. Finally to decouple the dependence on $\mathcal{H}$ in the second term of Eq. 12, that is $\mathbf{SP}\mathcal{H}$, we assume a rigid shift λ of the eigenspectrum [39, 40] given from an estimation of the highest eigenvalue calculated ($\lambda \approx -\epsilon_o$). These assumptions lead to an approximate Hessian,

$$\tilde{\mathbf{A}} = 2\mathbf{H} - 2\lambda\mathbf{S} - 2\left[\mathbf{HXX}^T\mathbf{S} + \mathbf{SXX}^T\mathbf{H}\right], \tag{13}$$

representing a square matrix of size $N_b \times N_b$, which can be applied as inner-preconditioner in the same way as $\mathbf{S}^{-1}$. We name this inner-preconditioner `AprxHess` and, similarly to the previous case, the notation `FullHess-`K`Iter-AprxHess` indicates solving the inner PCG by using the $\tilde{\mathbf{A}}^{-1}$ as inner-preconditioner and K iterations. Finally, we also consider the case for which the inner PCG is not solved, but rather the `InvOvl` or `AprxHess` are used for preconditioning the unconstrained functional minimization procedure (see Table 1 for a summary of the naming convention for the various preconditioners). More technical details on how the various iterative levels for the full SCF procedure work together is given in what follows.

As mentioned in the previous sections, the major computational steps of the unconstrained functional minimization (calculation of the energy functional, gradient, line search, etc.) can be implemented as matrix multiplication operations. Within CP2K, in addition to the PBLAS [3] library for parallel dense basic linear algebra, the parallel matrix multiplications are carried out by using the distributed block-compressed sparse row, DBCSR, library [14, 41]. DBCSR is a library designed to efficiently perform sparse matrix matrix multiplication, it is MPI and OpenMP parallel, and can exploit accelerators (such as GPU). The reason for using sparse linear algebra is that, due to the localized basis set, the $\mathbf{H}$ and $\mathbf{S}$ matrices are sparse, and this can be exploited to reduce the computational cost. In particular, for the $\mathbf{HX}$ and $\mathbf{SX}$ matrix multiplications, which are the most computationally demanding (see Fig. 2), one can take great advantage of sparsity even if $\mathbf{X}$ is dense, reducing the cost from cubic to quadratic. For this reason, even if overall our implementation is cubic scaling, we use sparse linear algebra matrix operations by employing the DBCSR library to reduce time to solution.

The unconstrained functional minimization is invoked within the SCF loop (see Fig. 1) to solve the eigenvalue problem at each step, providing a new set of wavefunctions which will then be used to update the electronic density. The construction of the new density

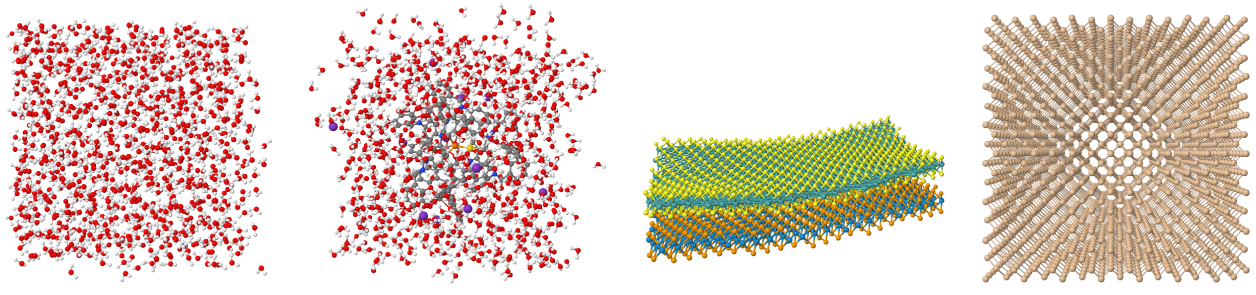


Figure 3: Systems used in the numerical experiments. (a) 1024 molecules of bulk liquid water (Water-1024 in the text), (b) supramolecular catalyst gold(III)-complex (Complex in the text), (c) bilayer of MoS_2-WSe_2 (BiLayer in the text), and (d) divacancy point defect in silicon (SiDivac in the text). The systems are in increasing order of "complexity" for convergence.

System	Label	Atoms	Basis	H Size (N_b)	Eigenvec. (N_o)	N_o/N_b	Gap(AU)
Bulk Liquid Water (Fig. 3a)	Water-1024	3,072	TZVP	29,696	4,096	0.14	0.128
	Water-2048	6,144	TZVP	59,392	8,192	0.14	
	Water-4096	12,288	TZVP	118,784	16,384	0.14	
Solvated Catalyst Complex (Fig. 3b)	Complex	2,590	TZVP	26,339	3,605	0.14	0.052
MoS_2-WSe_2 Bilayer (Fig. 3c)	BiLayer	2,247	TZVP	51,681	9,737	0.19	0.035
Divacancy Defect in Silicon (Fig. 4b)	SiDivac	2,742	TZVP	46,614	5,484	0.12	0.013
	SiDivac-SZV	2,742	SZV	10,968	5,484	0.50	
	SiDivac-DZVP	2,742	DZVP	35,646	5,484	0.15	
	SiDivac-TZV2P	2,742	TZV2P	79,518	5,484	0.07	

Table 2: Basic physical and computational parameters of the systems employed in the numerical experiments. H Size is the dimension of the Hamiltonian matrix that needs to be diagonalized at each SCF step (basis set size), Eigenvec. (N_o) is the number of eigenvectors to be computed (number of wavefunctions needed to build the electronic density), and Gap is the energy difference between eigenvalues $N_o + 1$ and N_o in atomic units (AU), which is the unit employed to express H.

is usually performed employing some mixing scheme which allows for accelerating convergence by using, for example, an extrapolation (such as DIIS [21]). Due to the density mixing and extrapolation schemes, it is not necessary to achieve full convergence in the solution of the inner diagonalization. Instead, it is important to provide an improved solution compared to the previous iteration, which will then be included in the extrapolation scheme. This implies that the PCG procedure carrying out the unconstrained functional minimization at each SCF step does not need to be fully converged, but just reduce the residual of the gradient by a reasonable amount. In general, only a small number of iterations (4-8) are performed and the PCG is considered converged and interrupted if the residual of the gradient is reduced by at least 0.1-10% from its initial value depending on the system type. These parameters may seem rather loose in terms of convergence tolerance but it has been shown, from a long history of experience in the field, to be the most effective approach to achieve faster overall SCF convergence in terms of time to solution. The reason is that, due to the non-linear nature of the SCF procedure, rather than having fully converged eigenvectors at each SCF step it is much more efficient to achieve an improved approximation to the subspace spanned by the final converged SCF eigenvectors. A more detailed discussion on the convergence criterion is given in Section 4.

As described earlier in this section, two different ways have been considered for preconditioning the unconstrained iterations. The first employed a preconditioner obtained from an $N_b \times N_b$ matrix, `InvOvl` and `AprxHess`. These preconditioners are calculated and applied to the trial vectors by using Cholesky factorization and inversion. The second, `FullHess`, finds the preconditioned gradient by solving for the Hessian of the unconstrained functional by employing the PCG method, invoked with K iterations and using as inner-preconditioner one of those introduced before (see earlier in this section for more details and Table 1 for a summary of the notation). Since the `FullHess` and `AprxHess` type preconditioners are built from the initial trial solution $\mathbf{X}$, special care needs to be taken. In fact, if the SCF procedure starts with a poor initial guess (such as those based on atomic orbitals) the immediate use of `FullHess` or `AprxHess` may lead to bad preconditioning, slow convergence and even divergence. To avoid such a condition, in our our implementation we allow for the first few steps of the SCF to use the `InvOvl` preconditioner (robust and independent of trial vectors). When a given convergence is achieved on the density the `FullHess` or `AprxHess` is used instead. Note that the `AprxHess` is not built at each SCF step, but every n steps (usually 10-20) to minimize the cost of the Cholesky factorization and inversion. Regarding the `FullHess`, a similar convergence criterion for the inner PCG is used as in the case of the unconstrained functional minimization, that is the PCG is considered converged if the residual of the gradient is reduced by at least 0.1-10% with a maximum of K iterations.

4 COMPUTATIONAL DETAILS

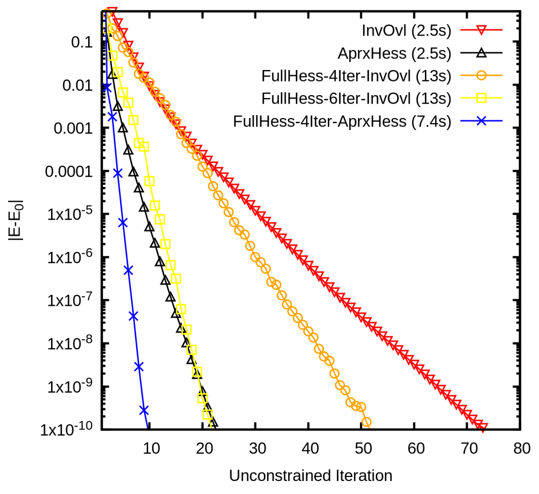
All calculations have been performed on the Cori computer, at the National Energy Research Scientific Computing Center (NERSC) [42] of the U.S. Department of Energy. Cori is a Cray XC40 system with two partitions, the one we used consists of over 9600 nodes, each node has one Intel Xeon-Phi 7250 processor, code named Knights Landing (KNL), connected with an Aries interconnect. KNL belongs to the many-core architecture family, each chip having 68 cores layed out in a 2D mesh (with 2 cores per "tile" sharing an L2 cache). For the calculations reported in this manuscript we used 64 cores for computation and reserved 4 for specialized tasks (system interrupts, etc.). The cores employed for specialization are not included in our performance measurement results (strong scaling plots, etc.). Our implementation is a hybrid MPI+OpenMP, optimal for many-core architectures such as Cori. The CP2K software package was compiled with the Intel compiler (vesion 18.0.5) and linked to the following libraries: MKL (Intel Math Kernel Library, version 2018.5.274), ELPA [8] (version 2017.05.002), and LIBXSMM [43] (a library optimized for small matrix-matrix multiplications used by DBCSR, version master-1.10-971).

To assess the performance of the unconstrained functional minimization in the solution of the generalized eigenvalue problem within the SCF framework for electronic structure calculations we have selected four benchmark systems. The system sizes range from 2,247 to 12,288 atoms, and are shown in Fig. 3. In more detail, we have considered various sizes of bulk liquid water (Fig. 3a), a solvated supramolecular complex of gold(III) used in homogeneous catalysis (Fig. 3b), a bilayer of transition metal dichalcogenide (MoS_2-WSe_2) , a substrate employed for quantum information applications (Fig. 3c), and a divacancy point defect in silicon, a prototype of a solid state qbit (Fig. 3d). The associated calculation parameters are given in Table 2. The water systems are used mainly to assess performance for strong scaling as this system can easily be scaled up in size, compared to the other three systems, by adding more water molecules. The other three systems have been considered to study convergence of the methods in various environments, from more homogeneous (like SiDivac) to highly heterogeneous (such as the Bilayer). These systems are increasingly more "complex" in terms of convergence, due to a reduced gap between the N_o and $N_o + 1$ states (see Table 2). Additionally, a wide range of N_o/N_b, up to almost 20% has been considered, for a reasonable comparison with direct solvers, which become more competitive for larger ratios. Unless otherwise stated, the employed basis set is the TZVP of the MOLOPT family [44], representing a family of compact and accurate contracted atom-centered Gaussian basis specifically designed to produce a well conditioned overlap matrix S, avoiding linear dependency problems and leading to stable calculations.

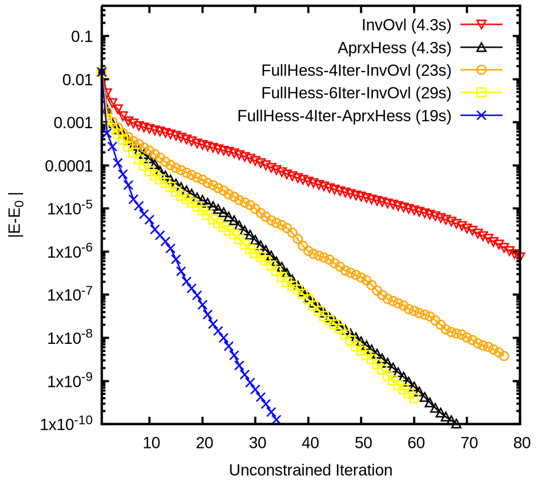
The initial guess for the SCF is given as a superposition of atomic orbitals which is in general not optimal. Therefore, as explained in the previous section, the initial preconditioner is in all cases of `InvOvl` type, then switched after a given number of SCF iterations ($\approx$ 20 for all systems) to a preconditioner based on the Hessian matrix, if required. The number of PCG iterations for the unconstrained minimization is set to a minimum of 2 and a maximum of 8; the PCG is considered converged if the initial gradient is reduced by 0.1%. The inner PCG solver for the inverse Hessian, if used, employs between 4-6 inner iterations and it is considered converged if the initial gradient is reduced by 0.01%. It may appear that these convergence parameters are rather loose, but (as explained in Section 3) for each SCF iteration it is only necessary to have an improved solution since the final goal is to converge the outer SCF loop, rather than each individual linearized eigenvalue problem at each SCF step. In general, if not otherwise stated, all calculations have been performed using 80 KNL nodes, with a configuration of 1280 MPI task $\times$ 4 OpenMP threads. Tests of running the four systems with different numbers of threads showed that a choice of four OpenMP threads gave overall best performance for the systems studied here (see next section).

5 RESULTS

We start by assessing the convergence properties of the unconstrained functional method both for a single diagonalization, i.e. the solution to full accuracy of a single generalized eigenvalue problem (one step in the SCF loop for the linearized eigenvalue problem), and the convergence of the whole SCF procedure for the full non-linear eigenvalue problem as used in electronic structure problems. In particular, Fig. 4 shows the convergence for the diagonalization of two systems, Complex (Fig. 4a) and SiDivac (Fig. 4b) systems. Results for the Water and Bilayer systems were extremely similar to the Complex and SiDivac systems, respectively, and due to space limitations we focus here on the results for the Complex and SiDivac systems. We compare the various preconditioners (Table 1) and report the convergence with respect to the unconstrained functional energy. Here we use an initial guess based on partially optimized wavefunctions corresponding to what we would use from a previous SCF step in the full SCF loop. As expected, the preconditioner based on `InvOvl` shows an improved convergence rate when it is used to solve the inner-PCG for the inverse Hessian; additionally, better convergence is systematically achieved by using more inner iterations (from 4 to 6). AprxHess gives a better preconditioning and leads to faster convergence: using it as is (without solving the inner-PCG), leads to similar convergence as using `FullHess-6Iter-InvOvl`. In terms of convergence rate, as expected, `FullHess-4Iter-AprxHess` is the best in all cases, giving convergence by one decimal place at each iteration for the Complex system (blue stars in Fig. 4a). The main difference between the two systems is the gap in the eigenvalue spectrum (see Table 2), so that SiDivac is a more challenging calculation due to the smaller gap. In fact, it is clear from the two plots that the convergence rate is faster for Complex compared to SiDivac. In the plots of Fig. 4 the number in parenthesis shows the average time associated with a single iteration of the unconstrained-PCG procedure. `InvOvl` and AproxHess have the same time per iteration because the $\mathbf{S}^{-1}$ and $\tilde{\mathbf{H}}^{-1}$ matrices have the same size and are computed with the same technique (Cholesky factorization

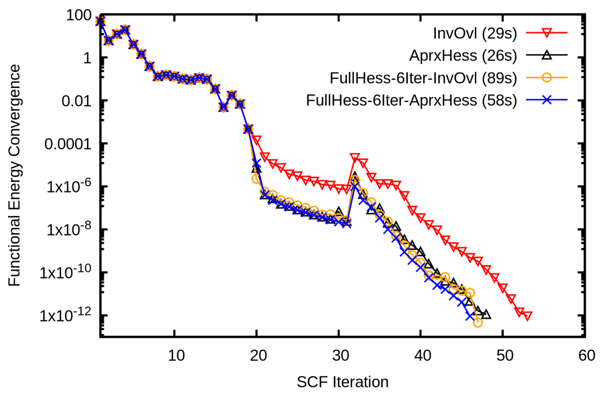


(a)

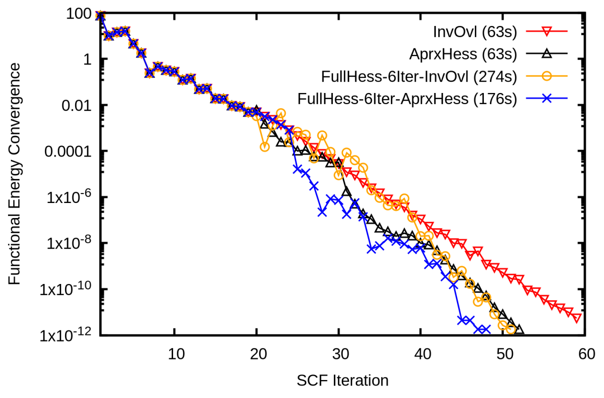


(b)

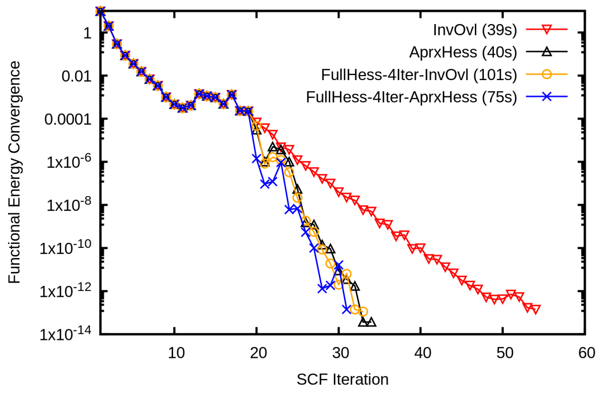
Figure 4: Convergence of the energy (unconstrained objective function) for a single unconstrained functional diagonalization (unconstrained subspace minimization) for the (a) Complex (Fig. 3b) and (b) SiDivac (Fig. 3d) systems. Five setups have been tested. The time for a single unconstrained-PCG iteration is reported in parenthesis.



(a)



(b)



(c)

Figure 5: Convergence of the SCF procedure for the (a) Complex, (b) BiLayer and (c) SiDivac systems. The figures show the convergence of the energy (unconstrained objective function) at the last iteration of the unconstrained-PCG procedure at each SCF step. Four setups have been tested, and the average time for a single SCF step is reported in parenthesis.

and inversion). The `FullHess` based preconditioner, even if having better convergence properties, is more expensive due to the solution of the inner-PCG. Interestingly `FullHess-4Iter-InvOvl` and `FullHess-6Iter-InvOvl` have similar time per unconstrained-PCG step, while `FullHess-4Iter-AprxHess` is almost two times faster. The reason for such a difference is due to the implementation

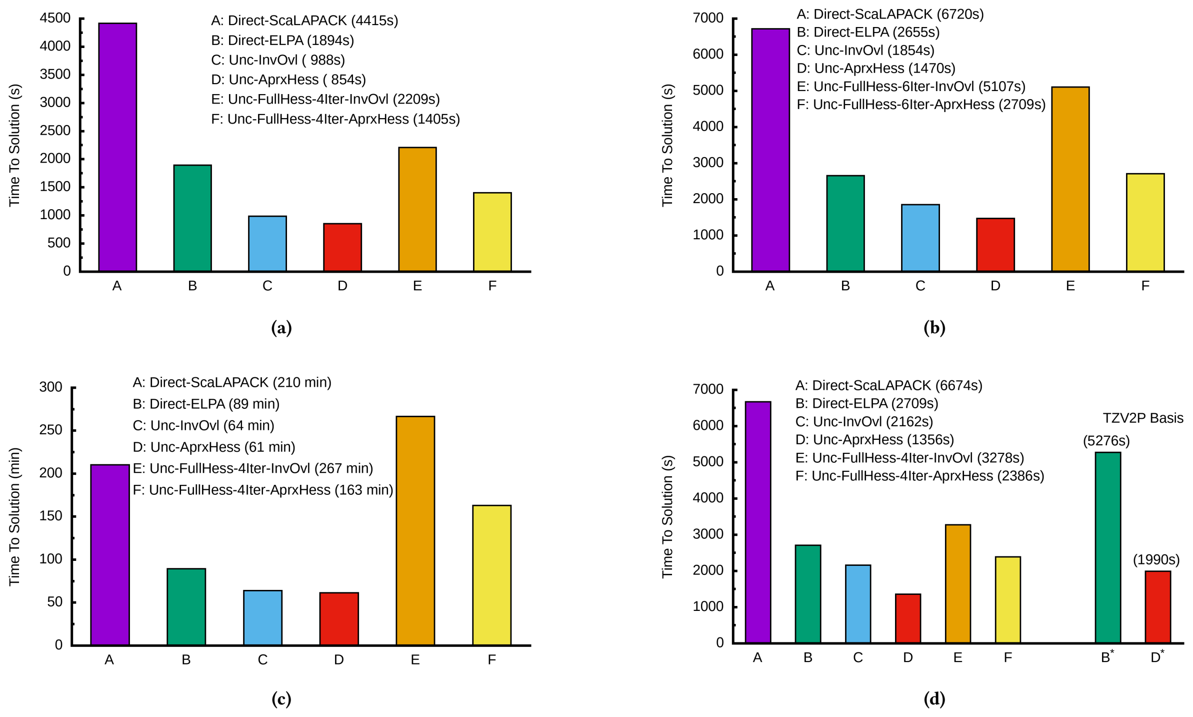


Figure 6: Time to solution for full SCF convergence using the unconstrained minimization method and setups, compared to direct solvers (ScaLAPACK and ELPA). (a) Water-1024, (b) Complex, (c) BiLayer and (d) SiDivac. The actual times are given in parenthesis. For SiDivac, B* and D* are times obtained with a larger basis ($\approx 1.7\times$ larger than in B and D, with 160 KNL nodes).

of the inner-PCG, for which the convergence is assessed as a percentage reduction of the initial gradients. This implies that using the better `AprxHess` preconditioner gives faster convergence (fewer iterations) per step, additionally using more steps with `InvOvl` generates a better sequence of trial vectors when approaching full convergence.

Next, we discuss the convergence for the full SCF procedure for the different systems considered depending on the various preconditioners and input setups. The results are reported in Fig. 5, in particular for Complex (Fig. 5a), Bilayer (Fig. 5b), and SiDivac (Fig. 5c). Here again we don't show the results for the water system because it does not bring any additional information compared to Complex as results were very similar for both systems. In the plots the convergence is reported in terms of the energy (the unconstrained objective function of Eq. 4) as obtained at the last iteration of the unconstrained-PCG procedure at the end of each SCF step. Here we use an SCF initial guess for the electronic density based on atomic orbitals. This choice is in general not optimal, but it represents the most general case (one instead can use an initial density obtained for example from a previous molecular dynamics or geometry optimization step). For this reason (as discussed in Section 3), preconditioners based on the Hessian of the unconstrained functional, which depends themselves on the trial vectors $\mathbf{X}$, cannot be used for the initial steps of the SCF, $\mathbf{X}$ being too far from convergence. To avoid slow convergence or even divergence when using a Hessian based preconditioner in combination with atomic initial guess, we start all SCF calculations reported here by employing the `InvOvl` preconditioner. After a given number of SCF iterations, a fairly good convergence for the trial vectors end electronic density is reached, the preconditioner is switched to the desired Hessian based preconditioner (if required). This is clearly shown in the plots of Fig. 5; in fact, for all approaches considered the initial SCF steps show the same convergence due to the usage of the same `InvOvl` preconditioner. The various curves then show different profiles when a different preconditioning treatment is turned on (around $\approx$ 20 SCF steps). In general, similar convergence rates are observed for the Hessian based preconditioners, in all cases faster (less iterations) than the overlap based `InvOvl`. Additionally, for Hessian based approaches, the solution of the inner-PCG for the Hessian gives slightly faster convergence than the plain `AprxHess`.

The situation is different when comparing the time to solution since the time per iteration is different for each preconditioner. In the plots of Fig. 5 the number in parenthesis gives the average time required for a single SCF step. As shown in the plot, the iterative solution of the inner-PCG gives a systematically higher computational cost compare to non-iterative approaches (i.e. `InvOvl` and `AprxHess`). As noted in the previous case, the difference in timing within iterative approaches comes from the different convergence rate of the inner-PCG, being faster for `AprxHess` than for `InvOvl`. We note that for the SCF procedure, the calculation complexity associated with the convergence of the density matrix, results in the inhomogeneous Bilayer being the most challenging system.

The time to solution for the four different systems for different precondtioners are reported in the histograms of Fig. 6. The results are also compared to the same calculations performed by using direct solvers, ScaLAPACK (MKL implementation) and ELPA. For all systems the best time to solution is achieved by using the `AprxHess` preconditioner, which outperforms all other approaches including the direct solvers. We note that the spread between the direct solvers and unconstrained minimization becomes larger when increasing the basis set size (see Fig. 6d). The reason is due to the $O(N_b^3)$ scaling of the diagonalization for the direct solver at each SCF cycle, whereas the only step scaling cubically with respect to N_b for the unconstrained approach is associated with the calculation of the preconditioner (either `InvOvl` or `AprxHess`) performed only once (for `InvOvl`) or very few (1-2 for `AprxHess`) for the whole SCF procedure. Therefore, as a lower percentage of the eigenpairs are required as the basis set size is increased the direct solvers become less competitive with respect to the iterative solvers.

Finally, we measure the performance of the unconstrained functional method in terms of scaling with number of threads Fig. 7 and parallel scalability (strong scaling), shown in Fig. 8. Fig. 7 shows the performance of the threaded implementation by keeping a constant number of MPI tasks and increasing the number of OpenMP threads per MPI task. The advantage of having a good threaded implementation is that when running at scale one can greatly reduce communication without loss of computational efficiency by increasing the ratio of OpenMP threads *vs* MPI tasks. For the strong scaling (Fig. 8), we consider three bulk water systems of increasing size and test the results over a large range of nodes, up to 3,600 KNL nodes on the Cray XC40, corresponding to around 38% of the full system. Overall good performance is obtained in all cases over a large range of nodes. The drop off in performance for large node counts is caused by the increased communication/computation ratios, particularly for the sparse linear algebra operations. In this respect, the recent progress in the use of one-sided MPI and a 2.5D algorithm to reduce communication in the DBCSR [45] are promising avenues to improve parallel scalability. Weak scaling performance for these types of electronic structure methods is more difficult to quantify. Although the overall scaling as we go to larger systems becomes cubic in the number of atoms, there are parts of the algorithm that scale with lower powers of the number of atoms. For example, the construction of the Hamiltonian, exploiting the sparsity of the basis set, scales linearly with system size and can take a significant percentage of the run time for small systems. This means we cannot simply scale the overall computational cost of the algorithm up by the cube of the system size to obtain weak scaling plots.

6 CONCLUSIONS

In this paper, we presented the performance of a preconditioned conjugate gradient based iterative eigensolver implemented in the CP2K code using an unconstrained energy functional minimization scheme that avoids explicit reorthogonalization of the trial eigenvectors. Four benchmark systems of up to 12,288 atoms were chosen to have a large spread of physics and eigenvalue spectrum properties to test our new approach. We studied the convergence properties with a range of different new preconditioners. In particular, we found that the convergence of the unconstrained approach can be very slow without the choice of a good preconditioner, although the best preconditioners can be costly to calculate. There is therefore a trade-off between the best preconditioner for fastest convergence such as the full Hessian and lower cost preconditioners that give the best time to solution. We found our approximate Hessian method `AprxHess` gave the best time to solution for all four systems studied. The difficulty of finding a good preconditioner is closely related to the use of a Gaussian basis where, unlike for a plane wave basis, the Hamiltonian matrix is not diagonally dominant. In the case of a plane wave basis the same simple preconditioner (basically the inverse squared of the wave number and variants of that) can be used for both the constrained and unconstrained functionals [25], which is not the case for a Gaussian basis. Finally, we showed for large systems our method has very good parallel scaling to 230k cores on the Cray XC40 computer and outperforms traditionally used direct solvers. This is mainly due to the fact that the unconstrained approach has no operations on any small subspace type matrices (i.e. matrices with dimension of the desired number of eigenvalues).

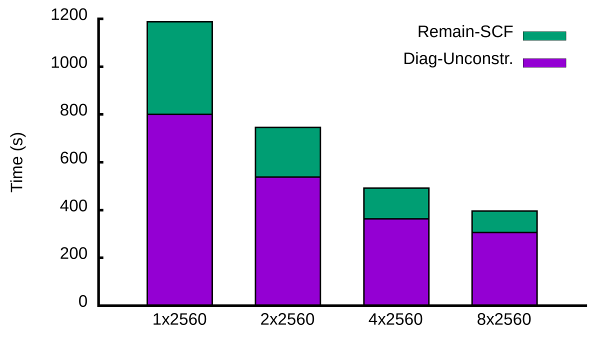


Figure 7: Strong scaling in terms of number of OpenMP threads per MPI task for a fixed number of MPI tasks (2560). The considered system is the bulk liquid water with 1024 molecules, corresponding to method D in Fig. 6a.

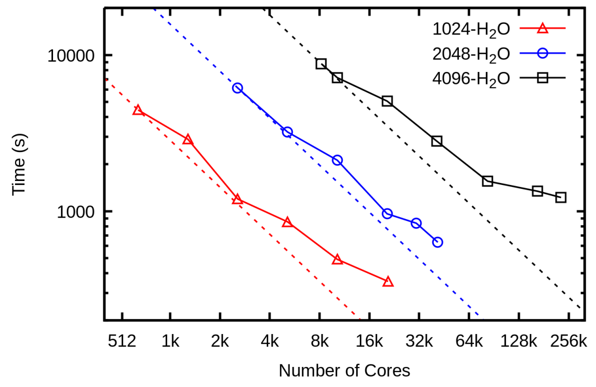


Figure 8: Strong scaling in terms of time to solution for bulk liquid water with 1024, 2048 and 4096 molecules, corresponding to method D in Fig. 6a.

In future work we plan to do more direct comparisons between our unconstrained functional approach and other constrained functional approaches which are starting to be used more in electronic structure codes using a Gaussian basis set. It should also be noted that, although we presented the unconstrained approach in the context of Gaussian basis electronic structure methods, it can be applied to any matrix. However, the gains in terms of performance and parallel scaling will be the greatest over traditional methods where the problem size is very large and a small percentage of the lowest eigenvalues and eigenvectors is required. In future work we also plan to study the unconstrained approach for matrices from other scientific application areas.

ACKNOWLEDGMENTS

Funding for this project was provided under the LBNL Laboratory-Directed Research and Development program. This research used resources of the National Energy Research Scientific Computing Center, a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231.

REFERENCES

[1] W. Kohn, L. J. Sham, Self-consistent equations including exchange and correlation effects, Phys. Rev. 140 (1965) A1133–A1138.
[2] J. P. Perdew, K. Burke, M. Ernzerhof, Generalized gradient approximation made simple, Phys. Rev. Lett. 77 (1996) 3865–3868.
[3] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, R. C. Whaley, ScaLAPACK Users' Guide, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.
[4] T. Auckenthaler, V. Blum, H.-J. Bungartz, T. Huckle, R. Johanni, L. Krmer, B. Lang, H. Lederer, P. Willems, Parallel solution of partial symmetric eigenvalue problems from electronic structure calculations, Parallel Comp. 37 (2011) 783–794. 6th International Workshop on Parallel Matrix Algorithms and Applications (PMAA'10).
[5] G. Ballard, J. Demmel, N. Knight, Avoiding communication in successive band reduction, ACM Trans. Parallel Comput. 1 (2015) 11:1–11:37.
[6] T. Imamura, S. Yamada, M. Machida, Development of a high-performance eigensolver on a peta-scale next-generation supercomputer system, in: Progress in Nuclear Science and Technology, Atomic Energy Society of Japan, 2011, pp. 643–650.
[7] EigenExa, High Performance Eigen-Solver, https://www.r-ccs.riken.jp/labs/lpnctrt/en/projects/eigenexa, 2018.
[8] A. Marek, V. Blum, R. Johanni, V. Havu, B. Lang, T. Auckenthaler, A. Heinecke, H.-J. Bungartz, H. Lederer, The ELPA library: scalable parallel eigenvalue solutions for electronic structure theory and computational science, J. Phys. Condens. Matter 26 (2014) 213201.
[9] C. Voemel, S. Tomov, O. Marques, A. Canning, L.-W. Wang, J. Dongarra, State-of-the-art eigensolvers for electronic structure calculations of large scale nano-systems, J. Comput. Phys. 227 (2008) 7113 – 7124.
[10] Y. Saad, J. Chelikowsky, S. Shontz, Numerical methods for electronic structure calculations of materials, SIAM Review 52 (2010) 3–54.
[11] G. Jordan, M. Marsman, Y.-S. Kim, G. Kresse, Fast iterative interior eigensolver for millions of atoms, J. Comput. Phys. 231 (2012) 4836 – 4847.
[12] A. Levitt, M. Torrent, Parallel eigensolvers in plane-wave Density Functional Theory, Comput. Phys. Commun. 187 (2015) 98–105.
[13] E. Ovtchinnikov, Computing several eigenpairs of Hermitian problems by conjugate gradient iterations, J. Comput. Phys. 227 (2008) 9477 – 9497.
[14] U. Bortnik, J. VandeVondele, V. Weber, J. Hutter, Sparse matrix multiplication: The distributed block-compressed sparse row library, Parallel Comp. 40 (2014) 47 – 58.
[15] A. Buluç, J. T. Fineman, M. Frigo, J. R. Gilbert, C. E. Leiserson, Parallel sparse matrix-vector and matrix-transpose-vector multiplication using compressed sparse blocks, in: Proceedings of SPAA '09, ACM, New York, NY, USA, 2009, pp. 233–244.
[16] CP2K, https://www.cp2k.org, 2019.
[17] E. D. Napoli, M. Berljafa, Block iterative eigensolvers for sequences of correlated eigenvalue problems, Comput. Phys. Commun. 184 (2013) 2478 – 2488.
[18] C. Bekas, A. Curioni, Very large scale wavefunction orthogonalization in density functional theory electronic structure calculations, Comput. Phys. Commun. 181 (2010) 1057 – 1068.
[19] T. Auckenthaler, T. Huckle, R. Wittmann, A blocked QR-decomposition for the parallel symmetric eigenvalue problem, Parallel Comput. 40 (2014) 186–194.
[20] G. Kresse, J. Furthmüller, Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set, Phys. Rev. B 54 (1996) 11169–11186.
[21] P. Pulay, Convergence acceleration of iterative sequences. the case of SCF iteration, Chem. Phys. Lett. 73 (1980) 393 – 398.
[22] G. Schofield, J. R. Chelikowsky, Y. Saad, A spectrum slicing method for the Kohn-Sham problem, Comput. Phys. Commun. 183 (2012) 497 – 505.
[23] C. Bekas, E. Kokiopoulou, Y. Saad, Computation of large invariant subspaces using polynomial filtered Lanczos iterations with applications in density functional theory, SIAM J. Matrix Anal. Appl. 30 (2008) 397–418.
[24] I. Štich, R. Car, M. Parrinello, S. Baroni, Conjugate gradient minimization of the energy functional: A new method for electronic structure calculation, Phys. Rev. B 39 (1989) 4997–5004.
[25] B. G. Pfrommer, J. Demmel, H. Simon, Unconstrained energy functionals for electronic structure calculations, J. Comput. Phys. 150 (1999) 287–298.
[26] F. Corsetti, The orbital minimization method for electronic structure calculations with finite-range atomic basis sets, Comput. Phys. Commun. 185 (2014) 873 – 883.
[27] SIESTA, https://departments.icmab.es/leem/siesta, 2019.
[28] S. Goedecker, Linear scaling electronic structure methods, Rev. Mod. Phys. 71 (1999) 1085–1123.
[29] V. W. zhe Yu, F. Corsetti, A. García, W. P. Huhn, M. Jacquelin, W. Jia, B. Lange, L. Lin, J. Lu, W. Mi, A. Seifitokaldani, A. Vázquez-Mayagoitia, C. Yang, H. Yang, V. Blum, ELSI: A unified software interface for Kohn-Sham electronic structure solvers, Comput. Phys. Commun. 222 (2018) 267 – 285.
[30] G. Lippert, J. Hutter, M. Parrinello, A hybrid gaussian and plane wave density functional scheme, Mol. Phys. 92 (1997) 477–488.
[31] J. VandeVondele, M. Krack, F. Mohamed, M. Parrinello, T. Chassaing, J. Hutter, Quickstep: Fast and accurate density functional calculations using a mixed gaussian and plane waves approach, Comput. Phys. Commun. 167 (2005) 103 – 128.
[32] F. Mauri, G. Galli, R. Car, Orbital formulation for electronic-structure calculations with linear system-size scaling, Phys. Rev. B 47 (1993) 9973–9976.
[33] F. Mauri, G. Galli, Electronic-structure calculations and molecular-dynamics simulations with linear system-size scaling, Phys. Rev. B 50 (1994) 4316–4326.
[34] P. Ordejón, D. A. Drabold, M. P. Grumbach, R. M. Martin, Unconstrained minimization approach for electronic computations that scales linearly with system size, Phys. Rev. B 48 (1993) 14646–14649.
[35] P. Ordejón, D. A. Drabold, R. M. Martin, M. P. Grumbach, Linear system-size scaling methods for electronic-structure calculations, Phys. Rev. B 51 (1995) 1456–1476.
[36] L. E. Cannon, A cellular computer to implement the Kalman filter algorithm, Ph.D. Dissertation. Montana State University., 1969.
[37] C. Gan, P. Haynes, M. Payne, Preconditioned conjugate gradient method for the sparse generalized eigenvalue problem in electronic structure calculations, Computer Physics Communications 134 (2001) 33 – 40.
[38] D. Bowler, M. Gillan, Length-scale ill conditioning in linear-scaling DFT, Computer Physics Communications 112 (1998) 103 – 111.
[39] J. VandeVondele, J. Hutter, An efficient orbital transformation method for electronic structure calculations, J. Chem. Phys. 118 (2003) 4365–4369.
[40] V. Weber, J. VandeVondele, J. Hutter, A. M. N. Niklasson, Direct energy functional minimization under orthogonality constraints, The Journal of Chemical Physics 128 (2008) 084113.
[41] DBCSR, https://www.cp2k.org/dbcsr, 2019.
[42] NERSC, http://www.nersc.gov, Last accessed August 7, 2019.
[43] LIBXSMM, https://github.com/hfp/libxsmm, Last accessed August 7, 2019.
[44] J. VandeVondele, J. Hutter, Gaussian basis sets for accurate calculations on molecular systems in gas and condensed phases, The Journal of Chemical Physics 127 (2007) 114105.
[45] A. Lazzaro, J. VandeVondele, J. Hutter, O. Schütt, Increasing the efficiency of sparse matrix-matrix multiplication with a 2.5d algorithm and one-sided MPI, in: Proceedings of the Platform for Advanced Scientific Computing Conference, PASC '17, ACM, New York, NY, USA, 2017, pp. 3:1–3:9.