arXiv:1610.08423v1 [physics.chem-ph] 26 Oct 2016

Co-iterative augmented Hessian method for orbital optimization

Qiming Sun*

Division of Chemistry and Chemical Engineering, California Institute of Technology, Pasadena, California 91125, USA

E-mail: osirpt.sun@gmail.com

abstract

Orbital optimization procedure is widely called in electronic structure simulation. To efficiently find the orbital optimization solution, we developed a new second order orbital optimization algorithm, co-iteration augmented Hessian (CIAH) method. In this method, the orbital optimization is embedded in the diagonalization procedure for augmented Hessian (AH) eigenvalue equation. Hessian approximations can be easily employed in this method to improve the computational costs. We numerically performed the CIAH algorithm with SCF convergence of 20 challenging systems and Boys localization of C_{60} molecule. We found that CIAH algorithm has better SCF convergence and less computational costs than direct inversion iterative subspace (DIIS) algorithm. The numerical tests suggest that CIAH is a stable, reliable and efficient algorithm for orbital optimization problem.

*To whom correspondence should be addressed

1 Introduction

Self-consistency field (SCF) is the cornerstone of the electronic structure simulation. It is required by almost all electronic structure methods. However, converging SCF to a reasonable solution is not a trivial problem. Direct inversion iterative subspace[1,2] (DIIS) was the most successful method to accelerate the SCF convergence. It was widely used in quantum chemistry program as the default optimization algorithm for SCF problem.

DIIS algorithm often has convergence problem for open shell systems, or other challenging systems which have small HOMO-LUMO gap. In the past, many SCF optimization techniques such as damping,[3] level shift[4] and enhanced DIIS algorithms EDIIS,[5] ADIIS[6] were proposed to improve the DIIS convergence.[7–14] Nevertheless, DIIS and its improvements have three main issues. (i) As an error-vector based minimization method, DIIS algorithm does not guarantee the SCF solution being at the true minimum. It is easy to have DIIS solution stuck at saddle point. (ii) DIIS algorithm does not honor the initial guess well. The optimization procedure may lead the wavefunction to anywhere in the variational space. (iii) DIIS algorithm does not have effective options to control the optimization procedure. One could tune the damping, level shift, DIIS subspace size, or extrapolation/interpolation constraints to influence the DIIS procedure. But the effects are not predictable.

The issues of DIIS algorithm can be easily addressed in the second order SCF optimization algorithm.[15,16] Second order algorithm (Newton or quasi-Newton methods) directly minimizes the function gradients with the assistance of Hessian matrix. The Hessian matrix can provide optimal displacement in the parameter space and the judgement whether a solution is at saddle point or local minimum. By tuning step size and trust region, one can gain the full control over the optimization procedure and constrain the solution to certain region.

However, there are new problems associated to the Newton methods. One is the cost to construct the Hessian matrix and its inverse matrix. Quasi-Newton methods address the problem by estimating the Hessian matrix (or its inverse) based on the change of gradients during the optimization. Such Hessian approximation may be a native choice for complicated object function. This

is not an optimal scheme for SCF problem because the simple structure of SCF Hessian matrix is not recognized by the gradients-oriented quasi-Newton methods. It is indeed unnecessary to couple the Hessian approximation to the gradients construction. If the Hessian is stripped from the gradients construction, one would gain larger flexibility to approximate the Hessian matrix. Since the Hessian matrix serves mainly as an auxiliary metric to adjust the descent direction of the displacement, the accuracy of Hessian matrix is not critical to the convergence. Thereby, many physical significant considerations rather than the pure numerical treatments can be brought into the Hessian approximation. For example, short-range approximation, frozen density approximation, density fitting technique etc can all be used with the Hessian construction.

Another practical problem one need consider for Newton's method is the treatment of the negative eigenvalue and the singularity of the Hessian matrix. Negative Hessian is common when the system is out of quadratic region. Non-invertible Hessian matrix is also regularly found near saddle point. They may result in the wrong direction or displacement in the parameter space. Level shift technique is often used to fix the singularity and adjust the descent direction. Here, the augmented Hessian (AH) method,[17–19] which can be traced back to the early work in the mutli-configuration self-consistent field (MCSCF) optimization, provided a decent solution to dynamically adjust the level shift of Hessian eigenvalues. In the region far away to the quadratic region, AH is close to the gradient descent method. When the gradients approach to zero, the AH equation turns to the normal Newton's equation.

Based on AH method, we propose a second order optimization algorithm, co-iterative augmented Hessian (CIAH) method for orbital optimization problem. The basic idea of CIAH algorithm is to decouple the Gradients and Hessian construction and embed the optimization procedure into the diagonalization procedure of AH method. The structure of orbital optimization parameters are considered in the algorithm. Particularly, small step is preferred by this algorithm than one shot "optimal" step. This will be more clear in Section 2 for the algorithm details. This algorithm is universal for a wide range of orbital optimization problem, from Hartree-Fock and Kohn-Sham energy minimization for restricted and unrestricted, closed and open shell, molecule and crystal,

to MCSCF optimization and orbital localization. In Section 3, we numerically verified the effectiveness of the algorithm with various kinds of SCF calculations and orbital localization.

2 Algorithm

Provided E the energy functional of the one-particle orbital rotation U

$$U = e^{\hat{R}} \tag{1}$$

$$\hat{R} = \sum_{pq} R_{pq} a_p^\dagger a_q \tag{2}$$

$$R_{pq} = -R_{qp}^* \tag{3}$$

the energy minimization problem can be considered as a non-linear search problem for the optimized $\mathbf{R}^*$ where the stationary condition holds

$$\frac{\partial E}{\partial R_{pq}}\Big|_{\mathbf{R}^*} = 0$$

In the kth iteration, one may predict a step $\mathbf{x}^{(k+1)} = \mathbf{R}^{(k+1)} - \mathbf{R}^{(k)}$ to approach the solution $\mathbf{R}^*$ by solving the AH matrix

$$\begin{pmatrix} 0 & \mathbf{g}^{(k)\dagger} \\ \mathbf{g}^{(k)} & \mathbf{H}^{(k)} \end{pmatrix} \begin{pmatrix} 1 \\ \lambda \mathbf{x}^{(k+1)} \end{pmatrix} = \varepsilon \begin{pmatrix} 1 \\ \lambda \mathbf{x}^{(k+1)} \end{pmatrix} \tag{4}$$

$$g_{pq}^{(k)} = \frac{\partial E}{\partial R_{pq}}\Big|_{\mathbf{R}^{(k)}}$$

$$H_{pq,rs}^{(k)} = \frac{\partial^2 E}{\partial R_{pq} \partial R_{rs}}\Big|_{\mathbf{R}^{(k)}}$$

The AH matrix here plays the role to damp the solution of Newton's method

$$\mathbf{g}^{(k)} + \lambda (\mathbf{H}^{(k)} - \varepsilon) \mathbf{x}^{(k+1)} = 0 \tag{5}$$

with level shift

$$\varepsilon = \lambda \mathbf{g}^{(k)\dagger} \mathbf{x}^{(k+1)}$$

The level shift circumvents the descent direction problem when the optimizer is far away from the quadratic region. When the optimizer approaches the local minimum, ε rapidly decays to zero. Eq (**??**) turns to the standard Newton's equation. The scaling factor λ is commonly used in AH algorithm to adjust the step size.[19,20] The AH eigenvalue equation can be solved iteratively using Davidson diagonalization program.

The CIAH algorithm is based on the iterative diagonalization procedure of the AH equation. To solve the AH eigenvalue problem (**??**), one has to evaluate the expensive matrix-vector production many times to generate the diagonalization subspace. For example, in the Hartree-Fock method, it involves the JK contraction which is as expensive as the Fock matrix construction. The costs can be effectively reduced in the CIAH program. Before moving to the details of the program, we would like first discuss the step size treatment of CIAH method.

In the CIAH program, we don't have sophisticated step size adjustment. A special feature of orbital optimization is that the matrix elements of the unitary transformation (**??**) must lie in the range $(-1,1)$. Ideally, the difference between the initial guess and the final answer can be filled with a series of small displacements. Therefore, we simply removed the λ parameter in Eq (**??**) and scaled down the largest element of the displacement vector $\mathbf{x}$ to a small predefined threshold δ

$$\begin{cases} \mathbf{x}^{(k)} & \max(\mathbf{x}^{(k)}) < \delta \\ \frac{\delta}{\max(\mathbf{x}^{(k)})} \mathbf{x}^{(k)} & \text{otherwise} \end{cases} \tag{6}$$

The step size thresholds are slightly different in different optimization problems. For SCF and MCSCF wavefunction, we prefer smaller step size $\delta = 0.03$ to provide a smooth convergence procedure because we usually have reasonable initial guess. In the orbital localization problem, the initial guess and final answer are very different. The optimization often starts with canonical orbitals and ends up with local orbitals. A larger threshold $\delta = 0.05$ is used so that the optimizer can

move quickly in the parameter space. Nonetheless, it should be noted that large step is generally not favored. In CIAH algorithm, small step has significant advantage over large step. This will become more clear in the rest part of the paper.

Because of the small step strategy, it can be expected that the Hessians in the adjacent iterations should be close to each other. Approximately, one can keep the Hessians unchanged and update only the gradients during several iterations. This requires small modification on the AH diagonalization program. In the conventional AH algorithm, one expands and diagonalizes the $(n+1)$-rank AH matrix in the subspace representation. In the modified version, we only keep track of the subspace corresponding to the *n*-rank Hessians matrix.When the system moved to the new point, we construct the gradients representation with the old *n*-rank basis $\{v\}$ and obtain the new representation of the AH eigenvalue equation.

$$H_{ij}^{(k+1)} = \langle v_i^{(k)} | H[\mathbf{R}^{(s)}] | v_j^{(k)} \rangle, \quad s < k$$
$$g_i^{(k+1)} = \langle v_i^{(k)} | g[\mathbf{R}^{(k+1)}] \rangle$$

On top of the Hessian reservation treatment, we embedded the function optimization iteration into the Davidson diagonalization iteration. Within each cycle of CIAH updating, the AH diagonalization solver enlarges the subspace by one basis vector $\mathbf{v}$ in terms of the Davidson preconditioner

$$\mathbf{v}^{(k+1)} = (\mathbf{H}_0 - \varepsilon)^{-1}(\mathbf{H}\mathbf{x}^{(k)} - \varepsilon\mathbf{x}^{(k)}) \tag{7}$$

The Hessian matrix representation is therefore improved gradually during the CIAH optimization cycles. Due to the error in the diagonalization solver, the displacement vector $\mathbf{x}$ might not be optimal in the early stage of the optimization. The error can be corrected during the optimization iterations. When the AH diagonalization solver gets enough bases to accurately represent the gradients and Hessian matrix, errors in the early iterations can be all removed and the optimal displacement will be generated. The small step strategy plays an important role in this treatment because it reduces the negative effects caused by the poor displacement vector in the early opti-

mization stage.

When the system is out of the quadratic region, the main purpose of the orbital Hessian is to adjust the direction of the displacement. The accuracy of the Hessian matrix is not highly important in this circumanstance. One can take rough approximations for orbital Hessian to reduce the computational costs, eg projecting the Hessian from low level basis sets, or superposition of fragment Hessian. Integral techniques can be considered as well such as the density fitting technique, high truncation tolerance for the AO-direct JK matrix contraction, sparse meshgrids (for DFT numeric integration), or even single precision integrals.

Unlike the Hessian approximations, it is less flexible to approximate the orbital gradients. Orbital gradients have two aspects of usage, the convergence criteria and a rough optimization direction. The gradients must be accurately evaluated when it was used as the convergence criteria. But as for the optimization direction, approximated orbital gradients are acceptable. The Tayler expansion of the orbital gradients around given point $\mathbf{R}^{(k)}$ is

$$g[\mathbf{R}] = g[\mathbf{R}^{(k)}] + H[\mathbf{R}^{(k)}] \cdot (\mathbf{R} - \mathbf{R}^{(k)}) + \dots \tag{8}$$

If the new point $\mathbf{R}$ is close to the expansion point, the gradients at $\mathbf{R}$ can be approximated by the first order expansion. It should be noted that the first order approximation may cause large error in orbital gradients, especially when approximated Hessians are employed. Our solution is to insert keyframes which are the exact gradients in certain iterations (while the Hessians are still approximated). A keyframe is triggered by two conditions: (i) if the gradients are out of trust region with respect to the previous keyframe, (ii) if the number of iterations between two keyframes is more than the predefined keyframe intervals. The keyframe here not only provides the adjustment to the optimization track, but also a reference to check whether convergence criteria are matched.

Here we briefly summerized the algorithm (Table 1 is an example of the evolution of each quantities during the optimization procedure).

1. Given an initial value $\mathbf{R}^{(0)}$, the optimizer starts to build the AH equation with one trial

vector and the AH matrix is a 2×2 matrix. Diagonalizing the AH matrix provides the first displacement $\mathbf{x}^{(0)}$ which is then scaled down to the predefined step-size threshold as shown by Eq. (**??**).

2. For *k*th iteration, the displacement $\mathbf{x}^{(k)}$ is used to update the Davidson subspace basis $\mathbf{v}^{(k+1)}$ using Eq. (**??**) and the gradients $\mathbf{g}^{(k+1)}$ with the first order approximation (**??**). The new basis and the approximated gradients are used to build the new AH matrix for the next displacement $\mathbf{x}^{(k+1)}$. This step is applied many times unitil the number of iterations reaches predefined upper limit (go to step 4) or the keyframe is triggered (go to step 3).

3. In the keyframe ($\mathbf{g}^{(4)}$, $\mathbf{g}^{(7)}$ in Table 1), gradients are evaluated exactly based on the cumulated displacements $(\mathbf{R}^{(0)} + \mathbf{x}^{(1)} + \cdots + \mathbf{x}^{(k)})$. Based on the exact gradients, there are three different conditions for the program flowchart: (i) If the norm of gradients is smaller than the required convergence criterion, we will call other convergence checks and prepare to finish the optimization; (ii) if the gradients are very different to the last approximate gradients (out of trust region, in which the ratio of new and old gradients' norm is over 3), the optimizer will move to step 4; (iii) otherwise, we insert the exact gradients into the AH equation (**??**) to generate a better displacement vector then go back to step 2

4. We move the system to new point ($\mathbf{R}^{(1)}$ in Table 1), then discard all basis vectors of the diagonalization solver and rebuild the AH matrix (as step 1). The last displacement is used as the initial guess of the Davidson diagonalization solver. The program will go back to step 2 and start a new cycle of optimization. We labelled such a cycle from step 2 - step 4 as a macro iteration.

The number of iterations (micro iterations) within each macro iteration is around 10 in our implementation. In the SCF procedure, JK contraction is the computational dominant operation. The number of JK contractions in CIAH algorithm is equal to the total number of micro iterations (due to the matrix-vector operations in the AH solver) plus the number of keyframes. However, the real cost is not as high as the number of JK contractions indicated. By using the Hessian

approximations, the time-consuming JK contractions only occur within the keyframes. In the CIAH algorithm, the small step size is the guarantee for the Hessian and gradients approximations. These approximations might lead to more macro iterations but generally it reduces the cost of JK operations thus improves the overall performance.

3 Numerical assessment

The CIAH algorithm were implemented in the open-source electronic structure program package PySCF.[21] It has been implemented in many orbital optimization modules, including SCF energy minimization for molecule and crystal, orbital localization and MCSCF orbital optimization. The algorithm works well for all of these scenarios. Since theories beyond the pure orbital optimization are involved in the MCSCF orbital optimization and SCF energy minimization for crystal, the relevant algorithm details and convergence performance will be documented in our upcoming papers.[22] In this paper, we demonstrate the properties of CIAH algorithm with molecular systems for SCF energy minimization and orbital localization.

In all SCF calculations, the convergence criteria is set to 10^{-10} for the change in energy and 10^{-5} in gradients norm. Point group symmetry broken is allowed if it can decrease the total energy. The initial guess orbitals for CIAH solver are fed from the regular DIIS-SCF iterations when the change of SCF energy is less than 1.0 E_{h}. The DIIS-SCF calculations are initialized with the superposition of atomic density for all-electron systems and core Hamiltonian for pseudo potential systems. In the DIIS-SCF iterations, unless otherwise specified, DIIS subspace size is 8 and level shift is 0.2. For ROHF methods, DIIS extrapolation is applied with Roothaan's open-shell Fock matrix.

3.1 Hessian approximations

We use the triplet state of Cr_2 and Fe-porphine molecules with ROHF/cc-pVDZ and UB3LYP/cc-pVDZ calculations to inspect the HF and DFT Hessian approximations, including the density-

fitting (DF), the basis projection, and sparse grids (for DFT). In the DF approximation, we employed Weigend Coulomb-fit basis as the auxiliary functions which are inaccurate for exchange integrals. In the sparse grids approximation, we computed the second order derivative of XC functionals with small meshgrids in which the number of radial grids and angular grids are (15,86) for light elements and (30,110) for transition metal while the normal meshgrids are (75,302)/(90,434). In the basis projection approximation, we expand the Hessian matrix on single-zeta basis then transform back to cc-pVDZ basis. The single-zeta basis is derived from cc-pVDZ basis by removing the outermost one shell for each angular momentum. The single-zeta basis significantly reduces the number of basis functions, from 86 to 54 for Cr_2 and from 439 to 159 for Fe-porphine. Level shift is not applied in the initial guess of $^3Cr_2$ because level shift leads to a symmetric solution which is high in total energy.

In Figure 1, we compare the SCF convergence for standard CIAH (without applying above approximations) and the approximate CIAH iterations. DF Hessian and sparse grids have high quality approximations to the Hessian matrix. Their convergence curves are close to the standard CIAH curve in all tests. The basis projection approximation has large error because it misses large fraction of the Hessian matrix. Except the early stage of the optimization, such error leads to obvious deviation to the standard CIAH convergence curve. In three of the tests, this poor approximation can converge the SCF to the right answer but require two times of the macro iterations or more. For $^3Cr_2$ ROHF, there are several local minimum answers. Depending on the numerical fluctuation during the optimization, basis projection can occasionally converge to the right answer.

In Table 2, we document the number of macro iterations, keyframes and JK contractions for each Hessian approximations. In the standard CIAH procedure, the costs to compute keyframe is roughly equal to the costs of JK contraction . The total costs are determined by the number of JK contractions which is dominated by the Hessian construction. In the approximate schemes, the Hessian construction becomes less expensive. The dominant step turns to the keyframe construction. For schemes like DF and the sparse grids, the number of keyframes is around 25% of the number of JK contractions. One could get 2 - 3 times speed up for the overall performance.

For example, by using the density fitting for ^{3}Fe-porphine UHF Hessian, the Hessian construction takes only 25% of the total computing time while it is over 70% of the computing time in the standard CIAH treatment. With regard to the costs of JK operations, even the poor basis projection approximation has significant performance advantage over standard CIAH optimization since it requires much less keyframes than the number of JK contractions of the standard CIAH scheme.

3.2 CIAH vs. DIIS

Based on the tests of Hessian approximations, the combination of DF and sparse grids provides an efficient scheme to approximate the Hessian matrix without the penalty of the convergence rates. They are adopted in our following SCF calculations. Applying basis projection at early optimization stage may further improve the computing time. It can be considered as a special treatment of initial guess. For simplicity, the basis projection technique is not included in the tests.

Table 3 presents the results of the CIAH and DIIS for some challenging SCF systems. CIAH algorithm shows better overall convergence than DIIS algorithm. Except $^3UF_4$ U-LSDA/LANL2DZ, CIAH algorithm is able to converge all test systems. Using DIIS, 5 systems do not meet the convergence requirements within 500 SCF iterations. For the converged systems, there are 8 answers that CIAH and DIIS algorithms show the good agreements. Aside from the 8 systems, CIAH and DIIS predicts closed solutions in 3 systems: $^3Cr_2$ U-LSDA/3-21G, $^1UF_4$ B3LYP/LANL2DZ and $^3UF_4$ U-B3LYP/LANL2DZ. In the 3 systems, CIAH solution is about 1 mE_h (or less) lower than DIIS solution. For the rest systems (except $^3UF_4$ U-LSDA which is not converged in CIAH), noticeable differences can be found between the two algorithms: the total energy predicted by CIAH algorithm is lower. Most of these systems are associated to the unrestricted calculations. These CIAH solutions have larger spin-contamination than that appeared in DIIS solutions. In these systems, the obvious spin-symmetry broken happens on the early stage of CIAH optimization which is not even observed in the DIIS iterations. One possibility is the side effects of level shift which limits the DIIS variational space. Some DIIS solutions actually correspond to the saddle point. For example, feeding the converged DIIS solution of $^3UF_4$ UHF/LANL2DZ to CIAH solver, with higher

convergence tolerance, CIAH solver takes 7 extra iterations to move to the expected lower-energy answer $E = -448.736476864$.

3.3 Orbital localization

CIAH algorithm can be applied with various type of orbital localization methods. Here we only demonstrated the Boys localization for the HF occupied orbitals and virtual orbitals of buckyball RHF/cc-pVTZ calculation as an example in Figure 2. Although not presented in this paper, ER and PM localization can be applied with the same solver.

Unlike the SCF initial guess, orbital localization often starts with canonical orbitals, which is typically very different to the final answer. For buckyball, the canonical orbitals cannot be directly taken as the initial guess for Boys localization because the orbital gradients are strictly zero (at saddle point). One solution is to put small noise on the initial guess to kick the system out of the saddle point. This is marked as "random" in Figure 2. Another initial guess we tried is pre-localization which is marked as "atomic" in Figure 2. In the atomic initial guess, we compare the canonical orbitals ψ with a set of reference AOs χ to define the rotation $\tilde{U}$ in terms of the SVD of the projection $\langle\chi|\psi\rangle$

$$\langle\chi|\psi\rangle = U\lambda V^\dagger$$
$$\tilde{U} = VU^\dagger$$

Transformation $\tilde{U}$ thus defines the initial guess orbitals $\tilde{U}\psi$ which are close to the reference AOs.

For occupied orbitals, the two kinds of initial guess produce small difference on the optimization procedure and converge to the same solution. The random initial guess is slightly worse at beginning. After about 7 steps to move out of the saddle point, it convergence as good as atomic initial guess. The optimization becomes difficult when diffused orbitals are involved. Many local minimum solutions are very close to the answer. For buckyball virtual orbitals, different initial guess (atomic and random), different step size (0.03, 0.05, 0.1) lead to different solutions in which

the values of object function are differed around 0.05 au. Regard to the system size and the total value of object function (12023 au), the difference is negligible.

One may hope that the large step size is superior since the initial guess is so different to the answer. In fact, the tests for the three step size (0.03, 0.05 and 0.1) present the closed convergence performance. They all take around 30 iterations to move away from the initial guess and the next 30 iterations wandering around the quadratic region. Once the solver steps into the quadratic region, they rapidly converge to the solution. Although not obvious in the figure, we observed during the optimization that the step size 0.1 sometimes overshoots the displacement and causes oscillation in the object function value. This observation numerically supports that the small step is able to provide the better gradients and Hessian estimations.

4 Conclusion

In this work, we demonstrated a general second order algorithm CIAH for orbital optimization problem. In the CIAH algorithm, the optimization is embedded in the augmented Hessian diagonalization procedure. The evaluation of gradients and Hessian are decoupled. Various approximations can be used for the Hessian matrix. By analyzing and comparing three Hessian approximations: density fitting, sparse grids, basis projection, we find that the combination of density fitting and sparse grids is able to produce accurate Hessian matrix. Our numerical tests of SCF convergence suggests that CIAH is a stable, reliable and efficient algorithm for SCF energy minimization problem. Apart from the molecular SCF energy minimization, CIAH algorithm can be used in many orbital optimization scenarios, such as MCSCF orbital optimization, SCF energy minimization with periodic boundary condition. Our numerical tests of orbital localization verifies the capability of CIAH algorithm to localize a large number of diffuse orbitals.

CIAH algorithm offers a new possible solution for a wide range of optimization problem. To make CIAH general purposed optimization algorithm, there remain some problems requiring further study. First is the constraint optimization. In the orbital optimization, the normalization con-

straints are imposed indirectly by the exponential ansatz for the unitary transformation. Applying the normalization constraints in a direct manner can avoid the expensive matrix exponential operation, which is particularly useful for huge systems. Moreover, the constraint optimization would allow CIAH being used in the geometry optimization and the transition state search. Second is the step size for a general purpose optimization problem since the small step assumption might not be effective for the general parameter space.

References

(1) Pulay, P. *Chem. Phys. Lett.* **1980**, *73*, 393 – 398.

(2) Pulay, P. *J. Comput. Chem.* **1982**, *3*, 556–560.

(3) Cancès, E.; Le Bris, C. *Int. J. Quantum Chem.* **2000**, *79*, 82–90.

(4) Saunders, V. R.; Hillier, I. H. *Int. J. Quantum Chem.* **1973**, *7*, 699–705.

(5) Kudin, K. N.; Scuseria, G. E.; Cancès, E. *J. Chem. Phys.* **2002**, *116*, 8255–8261.

(6) Hu, X.; Yang, W. *J. Chem. Phys.* **2010**, *132*.

(7) Cancès, E. *J. Chem. Phys.* **2001**, *114*, 10616–10622.

(8) Thøgersen, L.; Olsen, J.; Yeager, D.; Jørgensen, P.; Sałek, P.; Helgaker, T. *J. Chem. Phys.* **2004**, *121*, 16–27.

(9) Thøgersen, L.; Olsen, J.; Köhn, A.; Jørgensen, P.; Sałek, P.; Helgaker, T. *J. Chem. Phys.* **2005**, *123*.

(10) Høst, S.; Olsen, J.; Jansík, B.; Thøgersen, L.; Jørgensen, P.; Helgaker, T. *J. Chem. Phys.* **2008**, *129*.

(11) Host, S.; Jansik, B.; Olsen, J.; Jorgensen, P.; Reine, S.; Helgaker, T. *Phys. Chem. Chem. Phys.* **2008**, *10*, 5344–5348.

(12) Wang, Y. A.; Yam, C. Y.; Chen, Y. K.; Chen, G. *J. Chem. Phys.* **2011**, *134*.

(13) Chen, Y. K.; Wang, Y. A. *J. Chem. Theory Comput.* **2011**, *7*, 3045–3048, PMID: 26598147.

(14) Daniels, A. D.; Scuseria, G. E. *Phys. Chem. Chem. Phys.* **2000**, *2*, 2173–2176.

(15) Fischer, T. H.; Almlöf, J. *J. Phys. Chem.* **1992**, *96*, 9768–9774.

(16) Neese, F. *Chem. Phys. Lett.* **2000**, *325*, 93 – 98.

(17) Lengsfield, B. H. *J. Chem. Phys.* **1980**, *73*, 382–390.

(18) Jørgensen, P.; Swanstrom, P.; Yeager, D. L. *J. Chem. Phys.* **1983**, *78*, 347–356.

(19) Jensen, H.-J. A.; Jørgensen, P. *J. Chem. Phys.* **1984**, *80*, 1204–1214.

(20) Ghosh, D.; Hachmann, J.; Yanai, T.; Chan, G. K.-L. *J. Chem. Phys.* **2008**, *128*.

(21) Sun, Q. Python module for quantum chemistry program. `https://github.com/sunqm/pyscf.git`, 2014.

(22) Sun, Q.; Chan, G. K.-L. *arXiv* **2016**,

(23) Sharma, S.; Sivalingam, K.; Neese, F.; Kin-Lic, C. *Nat. Chem.* **2014**, *6*, 927–933.

(24) Groenhof, A. R.; Swart, M.; Ehlers, A. W.; Lammertsma, K. *J. Phys. Chem. A* **2005**, *109*, 3411–3417, PMID: 16833677.

Table 1: The evolution of gradients and Hessians during the CIAH iterations

Macro iter	AH subspace size	Hessians	Gradient	Displacement
1	1	$\mathbf{H}^{(0)} = \frac{\partial^2 E}{\partial R \partial R'}[\mathbf{R}^{(0)}]$	$\mathbf{g}^{(0)} = \frac{\partial E}{\partial R}[\mathbf{R}^{(0)}]$	$\mathbf{x}^{(1)}$
	2	$\mathbf{H}^{(0)}$	$\mathbf{g}^{(1)} \approx \mathbf{g}^{(0)} + \mathbf{H}^{(0)}\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$
	3	$\mathbf{H}^{(0)}$	$\mathbf{g}^{(2)} \approx \mathbf{g}^{(1)} + \mathbf{H}^{(0)}\mathbf{x}^{(2)}$	$\mathbf{x}^{(3)}$
	4	$\mathbf{H}^{(0)}$	$\mathbf{g}^{(3)} = \frac{\partial E}{\partial R}[(\mathbf{R}^{(0)} + \mathbf{x}^{(1)} + \cdots + \mathbf{x}^{(3)})]$	$\mathbf{x}^{(4)}$
	5	$\mathbf{H}^{(0)}$	$\mathbf{g}^{(4)} \approx \mathbf{g}^{(3)} + \mathbf{H}^{(0)}\mathbf{x}^{(4)}$	$\mathbf{x}^{(5)}$
	6	$\mathbf{H}^{(0)}$	$\mathbf{g}^{(5)} \approx \mathbf{g}^{(4)} + \mathbf{H}^{(0)}\mathbf{x}^{(5)}$	$\mathbf{x}^{(6)}$
	7	$\mathbf{H}^{(0)}$	$\mathbf{g}^{(6)} = \frac{\partial E}{\partial R}[(\mathbf{R}^{(0)} + \mathbf{x}^{(1)} + \cdots + \mathbf{x}^{(6)})]$	$\mathbf{x}^{(7)}$
	8	$\mathbf{H}^{(0)}$	$\mathbf{g}^{(7)} \approx \mathbf{g}^{(6)} + \mathbf{H}^{(0)}\mathbf{x}^{(7)}$	$\mathbf{x}^{(8)}$
$\mathbf{R}^{(1)} = \mathbf{R}^{(0)} + \mathbf{x}^{(1)} + \cdots + \mathbf{x}^{(8)}$				
2	1	$\mathbf{H}^{(1)} = \frac{\partial^2 E}{\partial R \partial R'}[\mathbf{R}^{(1)}]$	$\mathbf{g}^{(8)} = \frac{\partial E}{\partial R}[\mathbf{R}^{(1)}]$	$\mathbf{x}^{(9)}$
	2	$\mathbf{H}^{(1)}$	$\mathbf{g}^{(9)} \approx \mathbf{g}^{(8)} + \mathbf{H}^{(1)}\mathbf{x}^{(9)}$	$\mathbf{x}^{(10)}$

Table 2: Number of macro iterations and keyframes and JKs required for different CIAH approximations in $^3Cr_2$ and ^{3}Fe-porphine SCF optimization.

		$^3Cr_2$			^{3}Fe-porphine		
		Macro iters	keyframes	JKs	Macro iters	keyframes	JKs
ROHF	Standard CIAH	18	51	304	3	8	30
	DF Hessians	19	51	308	3	8	30
	Projected basis	18	37	184	14	24	63
	All in one	14	33	159	14	24	63
UB3LYP	Standard CIAH	17	38	150	9	21	84
	DF Hessians	17	39	170	10	21	87
	sparse grids	19	41	157	9	20	85
	DF +sparse grids	17	39	153	9	20	85
	Projected basis	26	49	183	22	39	147
	All in one	25	47	164	22	39	147

Table 3: CIAH, level shift 0.2 is applied in DIIS and the initial guess of CIAH. For the unconverged solutions, the oscillated decimal place and beyond are excluded.

Molecule	Method	CIAH[e]	*E*	DIIS	*E*
$^1Cr_2$[a]	LSDA/3-21G	18 (40)	-2073.907478426	191	-2073.907480839
	B3LYP/3-21G	5 (11)	-2078.075421673	31	-2078.075421674
$^3Cr_2$[a]	U-LSDA/3-21G	55 (175)	-2073.949040592	328	-2073.948413591
	U-B3LYP/3-21G	75 (135)	-2078.244163328	56	-2078.175189832
$^1UF_4$[a]	LSDA/LANL2DZ	15 (38)	-448.6767448225	184	-448.6767448158
	B3LYP/LANL2DZ	23 (56)	-451.0320849544	>500	-451.03129955
$^3UF_4$[a]	ROHF/LANL2DZ	18 (53)	-448.7341401101	431	-448.7176513294
	UHF/LANL2DZ	8 (27)	-448.7364768642	424	-448.7203771973
	U-LSDA/LANL2DZ	>500	-448.7307	193	-448.7310709215
	U-B3LYP/LANL2DZ	130 (188)	-451.0971319546	132	-451.0970715829
$^1Ru_4CO$[b]	RHF/LANL2DZ	13 (34)	-484.6768737985	125	-484.6768611861
	B3LYP/LANL2DZ	8 (19)	-488.4169663521	99	-488.4169663513
$^3Ru_4CO$[b]	ROHF/LANL2DZ	12 (30)	-484.7310915959	>500	-484.675
	UHF/LANL2DZ	29 (82)	-484.9692302601	>500	-484.6947
	U-B3LYP/LANL2DZ	123 (211)	-488.4555228944	280	-488.4348633206
^{3}Fe-porphine[c]	ROHF/cc-pVDZ	3 (8)	-2244.597443185	22	-2244.597443185
	UHF/cc-pVDZ	5 (14)	-2244.708766895	>500	-2244.6360
	U-B3LYP/cc-pVDZ	9 (20)	-2251.599470970	32	-2251.599470969
$^1Fe_4S_8C_4H_{12}^{2-}$[d]	RHF/cc-pVDZ	18 (44)	-8387.977176947	>500	-8387.7108
	B3LYP/cc-pVDZ	8 (21)	-8399.568031935	165	-8399.568031918

[a] Geometry is taken from Ref[14]
[b] Geometry is taken from Ref[6]
[c] Geometry is taken from Ref[24]
[d] Geometry is taken from Ref[23]
[e] Number in parenthesis is the total keyframes

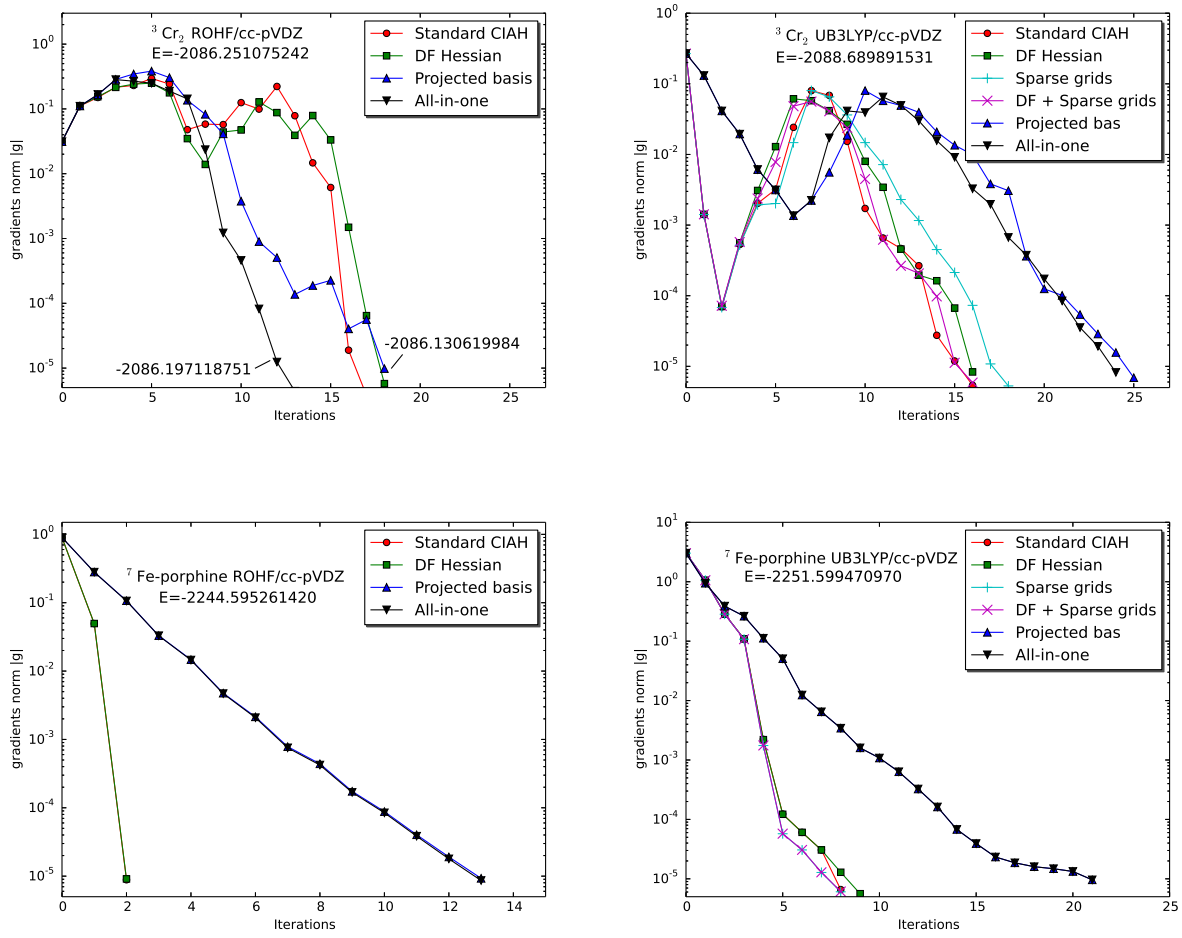

Figure 1: SCF convergence for different Hessian approximations.

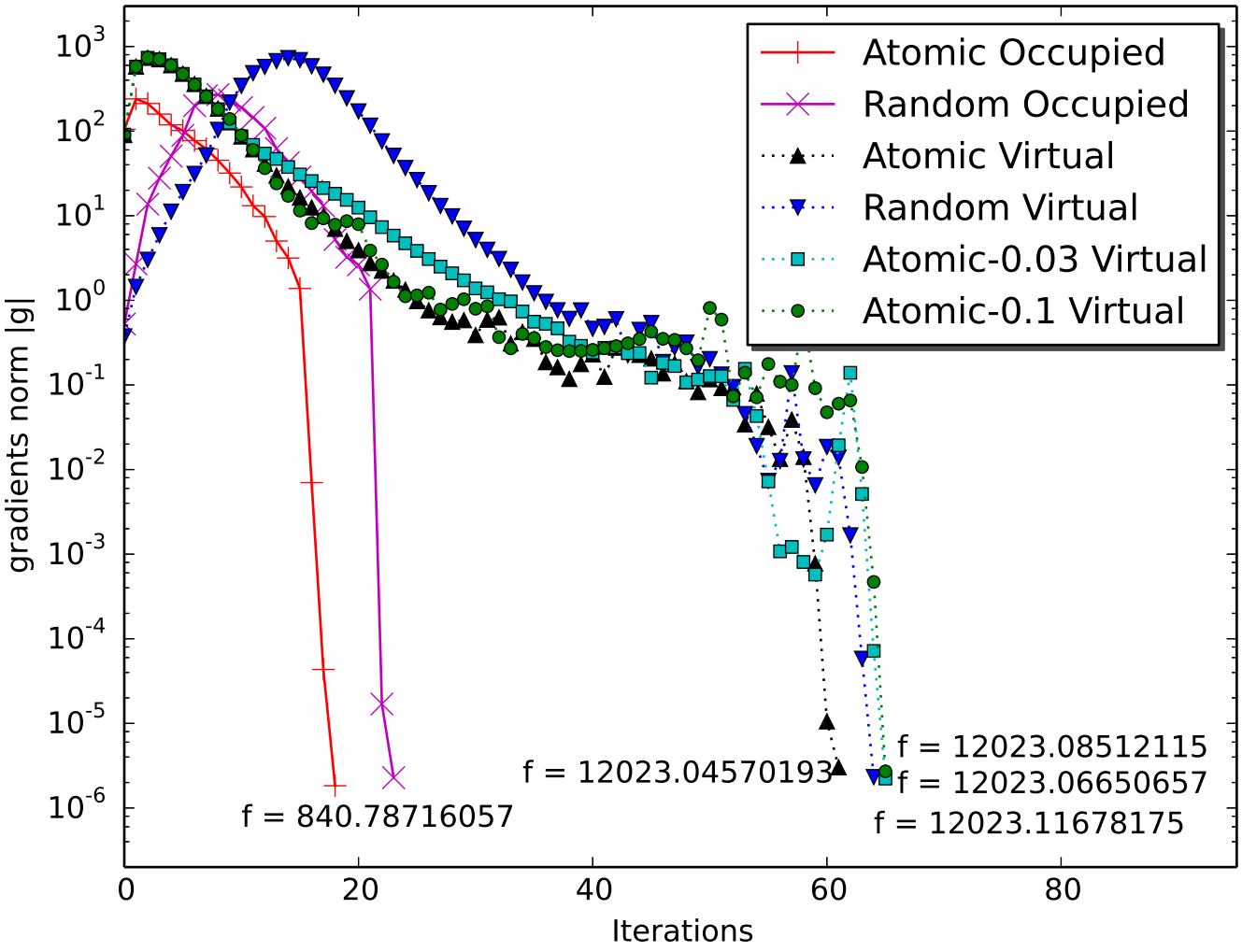

Figure 2: Boys localization for C_{60} molecule. The object function is $f = \sum_i \langle ii|(\mathbf{r}_1 - \mathbf{r}_2)^2|ii\rangle$