Numerical Nuclear Second Derivatives on a Computing Grid: Enabling and Accelerating Frequency Calculations on Complex Molecular Systems

Journal:	*Journal of Chemical Theory and Computation*
Manuscript ID	ct-2017-01235d
Manuscript Type:	Article
Date Submitted by the Author:	08-Dec-2017
Complete List of Authors:	Yang, Tzuhsiung; University of Wisconsin - Madison, Chemistry Berry, John; University of Wisconsin, Department of Chemistry

Numerical Nuclear Second Derivatives on a Computing Grid: Enabling and Accelerating Frequency Calculations on Complex Molecular Systems

Tzuhsiung Yang and John F. Berry*

Department of Chemistry, University of Wisconsin—Madison, 1101 University Avenue, Madison, Wisconsin 53706, United States

tyang29@wisc.edu, berry@chem.wisc.edu

RECEIVED DATE (to be automatically inserted after your manuscript is accepted if required according to the journal that you are submitting your paper to)

ABSTRACT: The computation of nuclear second derivatives (NSDs), or the nuclear Hessian, is an essential routine in quantum chemical investigations of ground and transition states, thermodynamic calculations, and the development of force fields. Analytic evaluation of NSDs (aNSD) requires the resolution of costly coupled-perturbed self-consistent field (CP-SCF) equations in second-order perturbation-like iterations. Previous work has contributed to the improvement of integral evaluation. Another approach to facilitate NSD computations focuses on hardware-enabled input/output reduction through faster integral recomputation by a computer algorithm such as direct self-consistent field (SCF) and parallelization of integral computation by faster multi-processor communication or specialized processors such as graphic processing units. Herein, we present a new method of improving NSD computations utilizing grid computing to enable numerical differentiation of analytic first derivatives to calculate the Hessian: nNSD@Grid. For accelerating routine NSDs using DFT, nNSD@Grid outcompetes aNSD methods as the ring size grows larger than three for a test set of linear polyacenes. For circumventing convergence issues associated with the CP-SCF procedure in aNSD for near-degenerate ground states, nNSD@Gird succeeded for the orbitally degenerate ^2E_g complex $Rh_2(O_2CCH_3)_4^+$ while aNSD failed to converge. For enabling NSDs using RIJCOSX-MP2, nNSD@Grid was 27 times faster and 46 times less memory-intensive than aNSD using 12 processors for naphthalene and enabled computation of NSDs for larger linear polyacenes. Practical examples using catalytically relevant transition metal containing complexes with or without near-degenerate ground states were also tested using nNSD@Grid and showed better time and convergence performance than aNSD. Grid computing using numerical methods can outperform analytic methods in terms of computing time, memory requirements, convergence, and treatable system size. The nNSD@grid method presented herein is one example of this concept and a pioneer for future implementations.

KEYWORDS: grid computing, Hessian evaluation, accelerating calculations, enabling calculations, numerical methods

1. INTRODUCTION

The computation of the second derivatives of the energy with respect to nuclear displacements χ and ζ (NSDs), or the nuclear Hessian, is an essential task in quantum chemical investigations for validating the curvature of potential energy surfaces at critical points, calculating or predicting vibrational spectroscopic features of molecules, and evaluating partition functions. Previous work on the evaluation of NSDs has focused on the analytic expression of NSDs (aNSD) for self-consistent field (SCF) based methods[1] and post SCF methods such as MP2,[2] CCSD,[3] CCSD(T),[4] and CASSCF.[5] For SCF-based methods, one starts with the expression for the total energy of the system

$$E = \mathrm{Tr}(\rho H) + \frac{1}{2}\mathrm{Tr}\big(\rho G(\rho)\big) + V_{nuc} + (1 - c_X)E_{XC} \ (1)$$

where ρ is the one-electron density matrix, $\rho_{\mu\nu} = \sum_i c^*_{\mu i} c_{\nu i}$ expressed in the linear combination of atomic basis functions, H are the one-electron integrals, $H_{\mu\nu} = (\mu|\nu)$, $G(\rho)$ are the two electron integrals, $G(\rho) = \sum_{\kappa\lambda} \rho_{\kappa\lambda} \left\{(\mu\nu|\kappa\lambda) - \frac{1}{2} c_X (\mu\kappa|\nu\lambda)\right\}$, c_X is the coefficient for exact exchange, i.e. $c_X = 1$ for HF and $c_X = 0$ for pure DFT, V_{nuc} is the nucleus-nucleus interaction energy, and E_{XC} is the exchange-correlation energy.

The analytical first derivative of the energy with respect to the nuclear displacement χ (aNFD) is

$\frac{\partial E}{\partial \chi} \equiv E^{\chi} = \mathrm{Tr}(\rho H^{\chi}) + \frac{1}{2}\mathrm{Tr}\big(\rho G^{\chi}(\rho)\big) + V_{nuc}^{\chi} + (1 - c_X)E_{XC}^{\chi} + \mathrm{Tr}(\rho^{\chi} H) + \frac{1}{2}\mathrm{Tr}\big(\rho^{\chi} G(\rho)\big)$ (2)

where the last two terms on the right-hand side require the resolution of the first derivative of the one-electron density matrix, which requires the resolution of $3N - 6(5)$ coupled perturbed SCF equations (CP-SCF) for a N-atom molecule.

The resolution of CP-SCF can be avoided for aNFD by using the orthonormality of one-electron density matrices:

$\mathrm{Tr}(\rho S) = 1$ (3)

where $S_{\mu\nu}$ are the overlap integrals, $S_{\mu\nu} = \langle \mu | \nu \rangle$. The first derivative of eq. 3 yields

$\mathrm{Tr}(\rho^{\chi} S) = -\mathrm{Tr}(\rho S^{\chi})$ (4)

and multiplication of eq. 4 by the SCF energy (ϵ) gives

$\mathrm{Tr}(\epsilon \rho^{\chi} S) = -\mathrm{Tr}(\epsilon \rho S^{\chi})$ (5).

The Hartree-Fock equation is

$\sum_{\nu} F_{\mu\nu} c_{\nu i} = \sum_{\nu} \epsilon_i S_{\mu\nu} c_{\nu i}$ (6)

where $F_{\mu\nu}$ is the Fock matrix, $F_{\mu\nu} \equiv \frac{\partial E}{\partial \rho_{\mu\nu}} = H_{\mu\nu} + \frac{1}{2} G(\rho_{\kappa\lambda})$. Multiplication of eq. 6 by the first derivative of the complex conjugate of orbital coefficients ${c_{\mu i}^{*}}^{\chi}$ from the right gives

$\sum_{\mu\nu} F_{\mu\nu} c_{\nu i} {c_{\mu i}^{*}}^{\chi} = \sum_{\nu} \epsilon_i S_{\mu\nu} c_{\nu i} {c_{\mu i}^{*}}^{\chi}$ (7).

Interchanging the dummy variables μ and ν in eq.7 gives

$\sum_{\mu\nu} F_{\mu\nu} c_{\mu i} {c_{\nu i}^{*}}^{\chi} = \sum_{\nu} \epsilon_i S_{\mu\nu} c_{\mu i} {c_{\nu i}^{*}}^{\chi}$ (8).

Adding eq. 7 to eq. 8 and use the commutation relationships $[\rho_{\mu\nu}, \mathrm{F}_{\mu\nu}] = 0$ and $[\rho_{\mu\nu}, \mathrm{S}_{\mu\nu}] = 0$

$\mathrm{Tr}(\rho^{\chi} F) = \mathrm{Tr}(\epsilon \rho^{\chi} S)$ (9).

Substituting eq. 5 into eq. 9 and expanding the Fock matrix gives

$-\mathrm{Tr}(\epsilon \rho S^{\chi}) = \mathrm{Tr}(\rho^{\chi} H) + \frac{1}{2}\mathrm{Tr}\big(\rho^{\chi} G(\rho)\big)$ (10)

where $\mathrm{Tr}(\epsilon\rho)$ is known as the energy-weighted one-electron density matrix W.

Using W avoids the necessity for resolution of CP-SCF equations in aNFD:

$$E^{\chi} = \mathrm{Tr}(\rho H^{\chi}) + \tfrac{1}{2}\mathrm{Tr}\big(\rho G^{\chi}(\rho)\big) + V_{nuc}^{\chi} + (1 - c_X)E_{XC}^{\chi} - \mathrm{Tr}(WS^{\chi}) \text{ (11).}$$

The derivative of eq. 11 with respect to the second nuclear displacement ζ gives the following expression for aNSD:

$$E^{\chi\zeta} = Tr\big(\rho H^{\chi\zeta}\big) + \tfrac{1}{2}Tr\left(\rho G^{\chi\zeta}(\rho)\right) + V_{nuc}^{\chi\zeta} + (1 - c_X)E_{XC}^{\chi\zeta} - Tr\big(WS^{\chi\zeta}\big) + Tr\big(\rho^{\zeta}H^{\chi}\big) +$$

$$Tr\left(\rho^{\zeta}G(\rho)\right) - Tr\big(W^{\zeta}S^{\chi}\big) \text{ (12)}$$

The resolution of CP-SCF equations to obtain ρ^{ζ} and W^{ζ} in the last three terms of eq. 11 can no longer be avoided.[6]

The resolution of CP-SCF equations requires numerous evaluations of two-electron integrals between occupied and virtual orbitals, which put high demands on CPU and storage due to their asymptotic cubic order scaling with respect to the number of atom-centered basis functions, i.e. $O(N_{basis}^{3})$, and $O(N_{basis}^{4})$ scaling of input and output (I/O),[1b] rendering the application of aNSD for large molecules intractable. To reduce the steep scaling of the CP-SCF equations, recent work has contributed to approximating the evaluation of the two-electron integrals by taking advantage of the fast decay of basis functions (e.g. the continuous fast-multipole expansion,[7] the chain-of-sphere approximation[8]) and function expansion of basis functions (e.g. the resolution of the identity,[9] occupied orbital RI-K,[10] J matrix engine,[11] pseudospectral[12]) to enable NSD evaluations for large systems. Detailed comparisons of these approximations can be found in the literature.[13]

It was noted by Almlof and coworkers in 1982 that development of central processing units (CPU) had allowed faster evaluations of integrals while the storage of these integrals exerted a high demand on the disk space, causing an I/O bottleneck. Hence, they proposed the integral handling method “direct SCF”, in which integrals from trial density matrices are recalculated for every SCF cycle resulting in a longer computing time in exchange for less temporary disk file storage enabling integral evaluations for larger molecules.[14] Based on the philosophy of direct SCF, Ahlrichs and coworkers suggested a combined integral handling, “semi-direct”, which selectively stores costly integral batches (i.e. integral batches formed by higher angular momentum bases) and reduces the amount of recalculations by minimizing the difference between the previous and the new trial density matrices.[15] Multiprocessor computers, such as high performance clusters (HPC), have allowed parallel computations of direct SCF[16] and direct CP-SCF[17] equations over multiple CPUs. Parallel computations based on semi-direct SCF were also developed.[18] Parallel implementation of post SCF methods has been developed as well.[19] The emergence of specialized computer hardware such as graphics processing units (GPU) has allowed quantum chemistry to achieve even faster and more capable parallel SCF[20] and post SCF [21] calculations. However, there are no implementations of CP-SCF on GPU architectures to our knowledge.

Despite ongoing theoretical efforts in approximating two-electron integrals and computing efforts in improving memory/storage management and parallelization of computations, the resolution of $3N-6(5)$ CP-SCF equations is still the most time consuming step in the aNSD.[17b] In addition, convergence of the CP-SCF equations presents another challenge for aNSD methods,[22] especially for systems with a small HOMO-LUMO gap (*vida infra*). One way to mitigate these challenges is to numerically differentiate analytic nuclear first derivatives (aNFDs), yielding numerical nuclear derivatives, nNSDs. Numerical differentiation truncated at the first order is

$$E^{\chi\zeta}(q_i) \approx \frac{E^{\chi}(q_i+h,q_j,\dots)-E^{\chi}(q_i-h,q_j,\dots)}{2h} \quad (13)$$

for $i=1,\dots,3N, j=1,\dots,3N$, and $j \neq i$ and where h is the step size for numerical differentiation.

The truncation error has recently been analyzed and shown to be negligible for most vibrations of chemical significance.[23] The advantage of nNSD over aNSD is that it circumvents the resolution of the costly CP-SCF equations, reducing the convergence issue and the asymptotic cubic scaling behavior; the scaling of nNSD is simply that of aNFD. In addition, nNSD is necessary when the theoretical methods employed do not have analytic expressions for NSD, such as OO-RIJCOSX-MP2. However, the pre-scaling factor of nNSD presents another challenge for conventional HPCs due to the fact that a molecule with N atoms will require 3N (forward or backward) or 6N (central difference) independent evaluations of the aNFD.

To remedy the unfavorable prefactor of nNSD, we exploit here recent developments in computing hardware, optical fibers, the Internet, and protocols for internetwork communication. Computing grids have arisen from these tools as a new form of computing infrastructure taking advantage of the curation of computer clusters at most academic institutions. Grid computing consists of a collection of distributed computing resources that are geographically disperse[24] yet allows users to access thousands of nodes across the globe. However, due to the decentralized nature of the network, less communication among nodes can be achieved than on a conventional HPC. This fact renders parallel computations that demand frequent communication and file sharing among nodes slower despite the increased amount of CPU hours available. On the other hand, calculations that are embarrassingly parallel (a feature of numerical differentiations such as nNSD!) can benefit greatly from grid computing. In this article, we present our new approach to performing nNSD on a grid platform: nNSD@Grid. We describe its implementation using the computing grid Open Science Grid (OSG),[25] including issues with and a solution to hardware inhomogeneity associated with the computing grid. At the level of Density Functional Theory (DFT), the performance of nNSD@Grid compared to aNSD is benchmarked to a systematic increase in molecule size and is found to excel for larger catalytically relevant transition metal containing complexes. The efficieny of nNSD@Grid in terms of computing time and memory for RIJCOSX-MP2 is also investigated. Finally, nNSD@Grid is used to accelerate nNSD using theoretical methods that do not yet have an analytic expression, such as OO-RIJCOSX-MP2, for their NSD.

2. IMPLEMENTATION

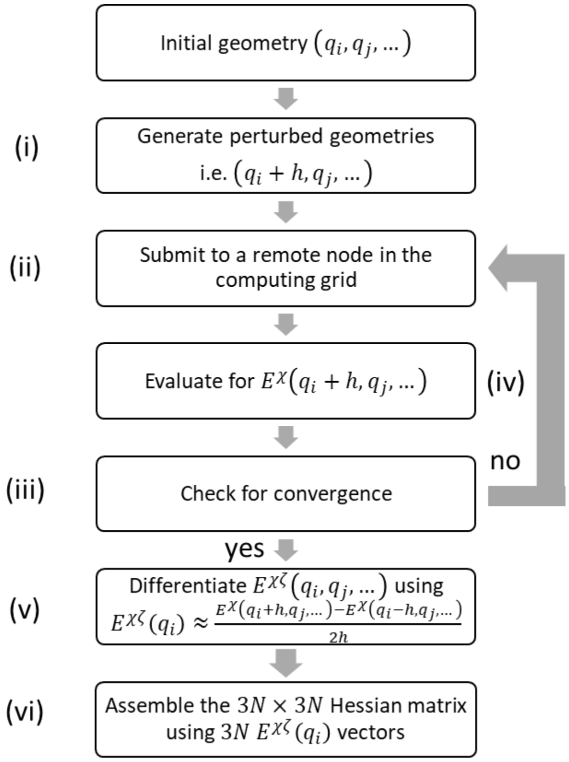
To implement and test the nNSD@Grid method with the OSG, unless specified otherwise, we used the hybrid generalized gradient approximation (hGGA) functional B3LYP[26] under the zeroth order regular approximation for scalar relativistic corrections (ZORA)[27] using the one-center approximation with the resolution of the identity and the chain-of-sphere approximations (RIJCOSX-ZORA-B3LYP) with the SARC-def2-SVP basis set (SARC-def2-TZVPP is employed on transition metals)[28] and the corresponding density fitting basis set[29] on C_6H_6 (186 primitive Gaussian functions (GFs) and 516 primitive Gaussian basis sets for density fitting (DFGFs)), $C_{10}H_8$ (296 GFs and 828 DFGFs), $C_{14}H_{10}$ (406 GFs and 1140 DFGFs), $C_{18}H_{12}$ (516 GFs and 1452 DFGFs), $C_{22}H_{14}$ (626 GFs and 1764 DFGFs), $C_{26}H_{16}$ (736 GFs and 2076 DFGFs), and $C_{30}H_{18}$ (846 GFs and 2388 DFGFs). A step size of 0.005 bohr is used for the numerical differentiation. For each aNFD calculation in one nNSD@Grid, we used *one processor by job*. For comparison, aNSD performed on our HPC was executed with one core and twelve processors by core. All calculations except RIJCOSX-B2PLYP[30] calculations (Orca 4.0.0[31] was used instead) were performed with the Orca 3.0.3 program package[32] for its compatibility with various types of computer platforms in the computing grid.

a. The workflow of nNSD@Grid.

To implement nNSD@Grid, we take advantage of one of the available computing grids in the United States, OSG. OSG allows users to access computing resources at other organizations through the protocols Globus and the job batch system HTCondor, in the form of HTCondor-G.[33] The strengths of HTCondor-G to the implementation of nNSD@Grid are the ease to bring users' input and software packages to various locations and the HTCondor utility Direct Acyclic Graph Manager (DAGMan). DAGMan allows the execution of jobs with declared dependencies.[34] This feature is essential for nNSD@Grid because fault tolerance of nNSD is stringent—any error in one of the aNFDs results in artifacts in the calculated nuclear Hessian—and the numerical differentiation can only be performed after all aNFDs are complete.

DAGMan provides a mechanism for job preparation (PRE SCRIPT), execution (JOB), monitoring and processing (POST SCRIPT), retry (RETRY), and child job execution (CHILD; for example, if thermodynamic parameters are to be calculated based on the nuclear Hessian). We therefore utilize these functions of DAGMan to (i) generate perturbed coordinates ($\chi+\Delta,\zeta,\ldots$) based on an initial coordinate ($\chi,\zeta,\ldots$) and a given increment of perturbation (Δ), (ii) submit concurrently duplicates of all aNFD calculations based on the perturbed coordinates, (iii) verify convergence of the aNFDs and (iv) retry if not converged and remove duplicates after the first in the duplicate set is complete, (v) differentiate the aNFDs numerically to produce the nNSD, and (vi) assemble all $3N$ nNSDs into the nuclear Hessian matrix (Scheme 1).

Scheme 1. Workflow of nNSD@Grid. All the arrows are controlled by the DAGMan utility. (i) A PRE SCRIPT is used to generate perturbed geometries based on a given initial geometry, (ii) each perturbed geometry is submitted as a JOB to the OSG to compute the aNFD (i.e. eq. 12), (iii) a POST SCRIPT checks the convergence of the computation after each completion of a JOB, (iv) RETRY is demanded if convergence was not detected in (iii), and (v) a POST SCRIPT to numerically differentiate E^{χ} at the first order (eq. 13) and (vi) assemble each numerically differentiated $E^{\chi}(q_i)$ to give the nuclear Hessian matrix.


Initial geometry $(q_i, q_j, \ldots)$
(i)
Generate perturbed geometries i.e. $(q_i + h, q_j, \ldots)$
(ii)
Submit to a remote node in the computing grid
Evaluate for $E^{\chi}(q_i + h, q_j, \ldots)$
(iv)
no
(iii)
Check for convergence
yes
(v)
Differentiate $E^{\chi\zeta}(q_i, q_j, \ldots)$ using $E^{\chi\zeta}(q_i) \approx \frac{E^{\chi}(q_i+h,q_j,\ldots)-E^{\chi}(q_i-h,q_j,\ldots)}{2h}$
(vi)
Assemble the $3N \times 3N$ Hessian matrix using $3N$ $E^{\chi\zeta}(q_i)$ vectors

b. Effect of grid inhomogeneity in computing time.

Grid computing suffers from computer hardware inhomogeneity, which reduces the efficiency of nNSD@Grid since *all* aNFS evaluations for a given set of coordinates have to finish before numerical differentiation can be performed to generate the nuclear Hessian matrix. Because of the opportunistic nature of grid computing, users have marginal control over the computer hardware on which their computing jobs are being executed. This issue causes the completion time of nNSD@Grid to fluctuate for replicates of any given calculation. To investigate how the fluctuation in computing time affects nNSD@Grid, we performed nNSD@Grid on C_6H_6 six times and the computing time of all the aNFDs in each replicate is shown on the left in Figure 1. Because there are multiple computations of aNFD for each nNSD@Grid, the computing times for all these calculations are shown in percentile bars with the thicker portion representing the mid-range 25–75% timings. It can be conceived that 75% of all the aNFDs for C_6H_6 finished within 100 seconds after execution. However, the whole nNSD can take up to almost double the time of the 75th percentile due to the variation in hardware on the computing grid. In addition, the variation in the total computing time (the 100th percentile) of the six replicates ranges by 33%.

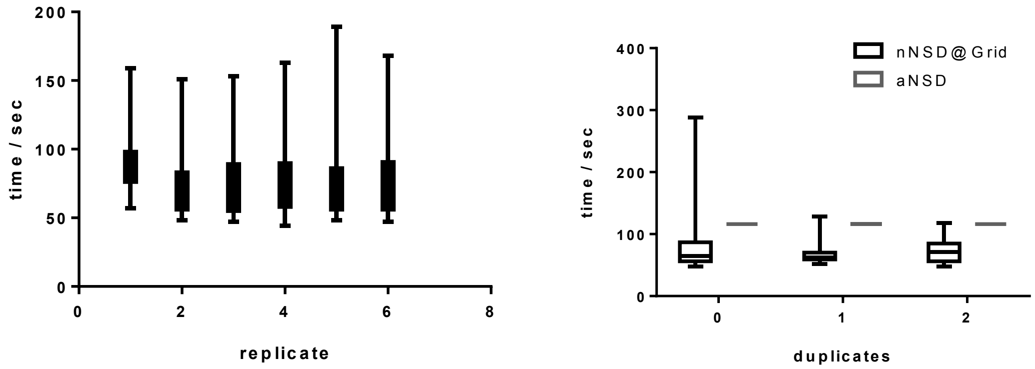


Figure 1. Variation in the computing time within each set of aNFDs and among replicates of nNSD@Grid on C_6H_6 (left) and the computing time performance of nNSD@Grid on C_6H_6 using duplicates of submissions (right)

c. Mitigation of grid inhomogeneity with duplicate submissions

Inhomogeneity of the computing grid causes a skewed distribution of computing times within a set of aNFDs for one nNSD@Grid calculation, delaying the completion of the nNSD@Grid calculation. To mitigate this inherent issue of grid computing, we take advantage of the high throughput aspect of the cyberinfrastructure. The probability of two identical submissions finishing at the same time in the computing grid is almost negligible; one calculation almost always finishes earlier than the other. Hence, we may submit duplicates of a set of aNFDs to remedy the delay due to hardware heterogeneity of the computing grid. The results of the distribution of computing time of nNSD@Grid on C_6H_6 using one and two duplicates as compared to that without duplicates are shown on the right in Figure 1. It can be seen that the completion time is reduced by a factor of two by using one duplicate of nNSD@Grid, making the time performance of nNSD@Grid comparable to that of aNSD. Including two duplicates only marginally improves the computing time for nNSD@Grid compared to that using one duplicate. To reduce the amount of computing resources wasted in computing duplicates of the individual aNFDs, we utilized the POST SCRIPT function in DAGMan to remove the slow duplicate executions on-the-fly once one aNFD calculation is complete in a set of duplicates.

3. APPLICATIONS

a. Performance of nNSD@Grid compared to aNSD

i. Time performance for a systematic increase in system size

Hessian matrices for compounds having various numbers of linearly fused benzene rings were computed using either nNSD@Grid or aNSD. The computing time for either method is shown in Figure 2. It can be found that nNSD@Grid outcompetes aNSD as the number of fused benzene rings becomes larger than three. The advantage of nNSD@Grid in terms of timing becomes more prominent as the system size grows.

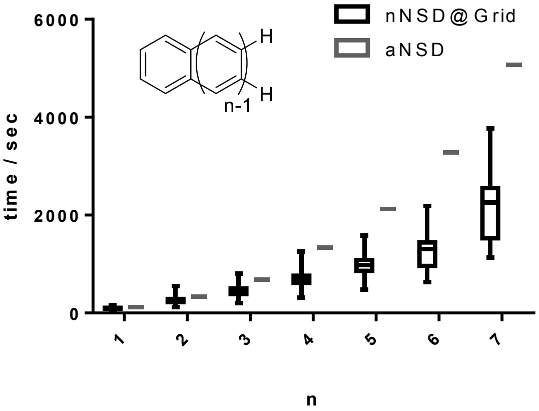


Figure 2. Computing time on various sizes of fused benzene molecules performed by nNSD@Grid (black percentile bars) versus aNSD (grey bars).

ii. Convergence performance for multiconfigurational/multireference states

Degenerate/Near-degenerate ground states can cause difficulties for the evaluation of CP-SCF equations due to the instability of the first derivatives of the density matrix and the energy weighted density matrix caused by large changes in the coefficients of atomic orbitals due to nuclear displacements. This problem renders the convergence of CP-SCF equations slow or difficult but is mitigated by nNSD. To demonstrate the advantage of nNSD@Grid over aNSD in degenerate/near-degenerate ground states, we performed nNSD@Grid and aNSD calculations on the $[Rh_2(O_2CCH_3)_4]^+$ ion. $[Rh_2(O_2CCH_3)_4]^+$ has an orbitally degenerate doublet ground state ^2E_g in the high symmetry D_{4h} group (Figure 3)[35] and axially ligated derivatives of the complex have abnormal Landé g factors arising from strong first order spin-orbital coupling.[36] aNSD of $[Rh_2(O_2CCH_3)_4]^+$ was performed using the SCF convergence method of Pulay (the direct inversion of the iterative space),[37] of Pople,[38] or the conjugated gradient method. All three convergence methods failed to converge the CP-SCF equations after 128 cycles. In contrast, nNSD@Grid completed resolution of the nuclear Hessian in 11-46-61-74-169 min (for the 1st, 25th, 50th, 75th, and 100th percentiles).

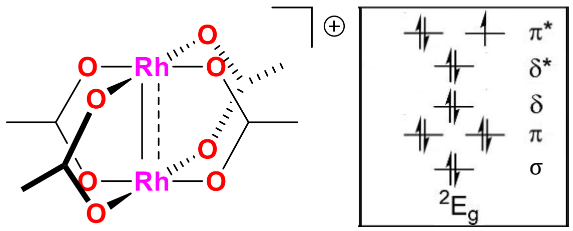


Figure 3. Structure (left) and electronic configuration (right) of $^2[Rh_2(O_2CCH_3)_4]^+$.

b. Enabling theoretical methods that are prohibitively expensive for large molecules using aNSD by nNSD@Grid

i. Hessian evaluation using RIJCOSX-MP2 for various numbers of fused benzene rings

aNSD calculations using canonical MP2 scale formally as $O(N_{basis}^6)$.[2] The steep scaling renders aNSD intractable for large molecules on a conventional HPC. In comparison, energy and gradient calculations using canonical MP2 are one order more tractable, formally scaling as $O(N_{basis}^5)$ with system size.[39] Therefore, nNSD@Grid is computationally more accessible for larger molecules compared to aNSD. To study the computing time and memory performance of nNSD@Grid *vs.* aNSD, we computed the RIJCOSX-MP2 Hessian matrix of the seven polyacenes using nNSD@Grid or aNSD and the results are shown in Figure 4. It is evident that nNSD@Grid outperforms aNSD in both computing time and memory usage, where nNSD@Grid is 8 times faster and 13 times less memory-intensive for C_6H_6 and 27 times faster and 46 times less memory-intensive for $C_{10}H_8$ than is aNSD using 12 processors. aNSD could not even be completed for larger linear polyacenes using RIJCOSX-MP2 without further approximations.

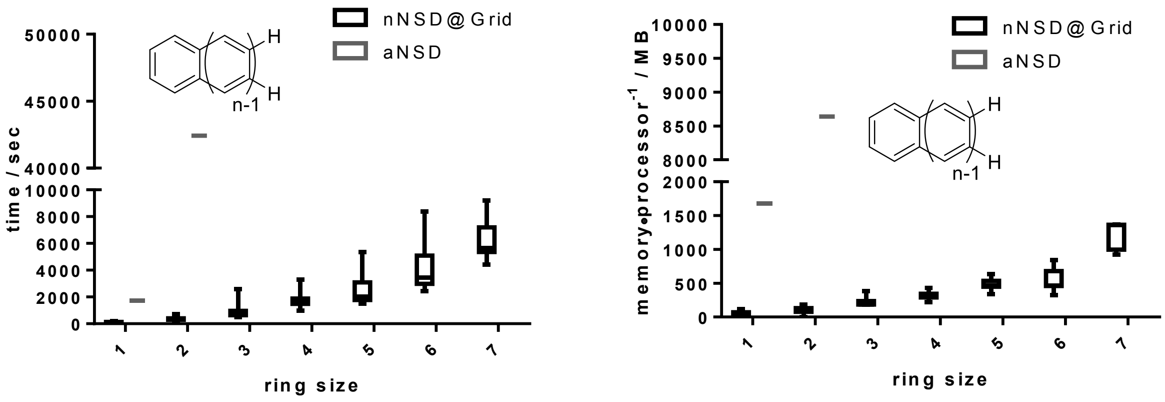


Figure 4. Computing time (left) and memory usage per processor (right) for various molecules of linearly fused benzene rings by nNSD@Grid (black percentile bars) and aNSD (only trackable for C_6H_6 and $C_{10}H_8$; grey bars) using RIJCOSX-MP2 are shown.

ii. Hessian evaluation using RIJCOSX-B2PLYP for various numbers of fused benzene rings

Double hybrid density functionals (DHDFs) are an emerging rung on the Jacob's Ladder of density functionals that demonstrate improved performance, compared to the already successful hybrid DFT, for their improved recovery of nonlocal correlation energy.[40] The computing time performance of nNSD@Grid using a DHDF, RIJCOSX-B2PLYP, is shown in Figure 5. It can be found that RIJCOSX-B2PLYP frequencies can easily be routine for systems as big as $C_{30}H_{18}$, finishing within 6 hours (*vida supra* for the number of GFs and DFGFs). Inspecting the distribution of total computing time among various components of the RIJCOSX-B2PLYP aNFD calculations, it becomes apparent that more than 50% of the computing time was invested in preparing and solving the MP2 correction of the nonlocal exchange (Figure 5). Any improvement, such as parallel calculation, in the MP2 timing would significantly accelerate the computing time of nNSD@Grid. However, parallel calculation of each aNFD of nNSD@Grid using more than one processor in one remote computer node significantly reduces the number of available nodes in the computing grid, hence, lengthening the queue time of each aNFD. One potential way to mitigate the time investment in the MP2 module is to partition the correlation space into small orbital fragments and distribute each fragment to each remote node in the computing grid, transforming the MP2 amplitude calculation into an embarrassingly parallel task, further taking advantage of the high scalability of the computing grid. Future development of nNSD@Grid to incorporate various types of fragmentation methods, such as divide-expand-consolidate,[41] is foreseeable.

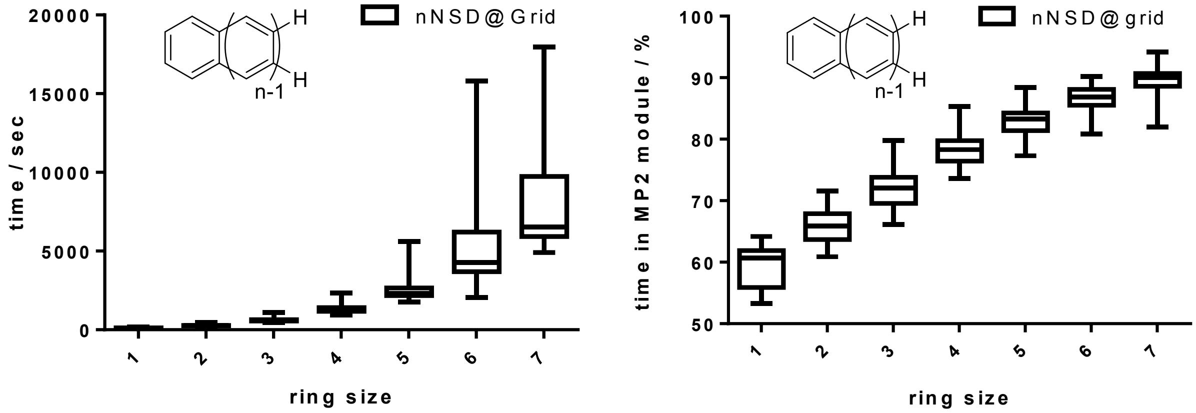


Figure 5. Computing time (left) and fraction of total computing time spent in the MP2 module (right) for various molecules of linearly fused benzene rings by nNSD@Grid using RIJCOSX-B2PLYP are shown.

c. Accelerating computation of theoretical methods without aNSD expressions by nNSD@Grid

Theoretical methods that have aNFD expressions but do not yet have aNSD expressions are less commonly employed for frequency calculations due to the unfavorable prefactor of nNSD for large molecules. This limitation hampers routine performance of these methods for reactivity relevant studies despite their improved accuracy. nNSD@Grid allows attenuation of the unfavorable prefactor at the expense of a longer computation time of each aNFD owing to the fewer accessible multiprocessor computing slots in the computing grid. The efficiency of nNSD@Grid compared to conventional nNSD on a conventional HPC using central differences can be approximately equated to:

Computing time for one aNFD using nNSD@Grid = t (14)

Computing time for one NSD using nNSD on an HPC = $\frac{6Nt}{(np)^{\alpha}}$ (15)

where *N* is the number of atoms, n is the number of sets of processors dedicated to solving each aNFD, p is the number of processors in each n set dedicated to solving one aNFD, α is the scalability of aNFD with respect to the number of processors.

Dividing eq. 15 by eq. 14 gives the relative speed of nNSD@Grid *vs.* nNSD on an HPC as $\frac{6N}{(np)^{\alpha}}$.

For example, for one C_6H_6 molecule running on 12 processors in 1 node, $N = 12$, $n = 1$, and $p = 12$, nNSD completed in 459 seconds compared to at most 158 seconds by nNSD@Grid. Using a phenomenological inverse relationship ($\alpha = 1$) between the computing time and $n \times p$ (Figure 6), the computing time of nNSD is extrapolated to 135 seconds on 36 processors, which is comparable to that of nNSD@Grid in spite of the high demand on the number of processors for a molecule as small as C_6H_6.

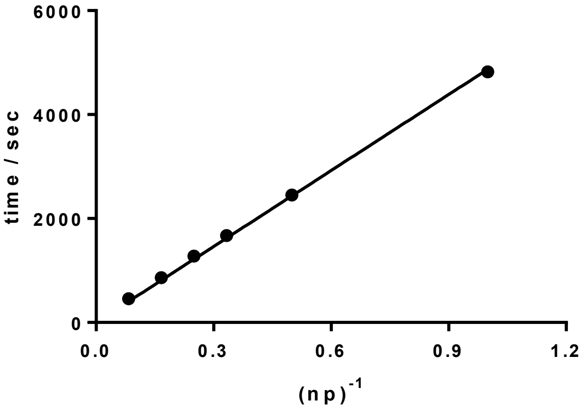


Figure 6. Linear correlation between the amount of time to complete nNSD for C_6H_6 and the inverse of the mutplication of the number of computer cores by the number of processors per computer core. The fitted linear regression is $time = 4872 \times (np)^{-1}$.

After demonstrating the advantage of nNSD@Grid over nNSD in the demand of the number of processors in each node, we now show the applicability of nNSD@Grid using OO-RIJCOSX-MP2, an improved formulation of RIJCOSX-MP2 for radicals, to linear polyacenes.

MP2 provides less accurate results for open-shell systems than DFT due to the only partial recovery of electron correlation from the poor HF reference.[42] It has been shown that some electron correlation can be further recovered at the MP2 level by basing the calculation on the Hylleraas functional which is made stationary with respect to both amplitude variations and orbital rotations.[42] Despite its improvement over RIJCOSX-MP2, OO-RIJCOSX-MP2 calculations require more computing time for orbital optimizations. We first show the computing time and memory performance of OO-RIJCOSX-MP2. Comparing Figure 7 with Figure 4, it can be seen that OO-RIJCOSX-MP2 has an overall similar memory usage to RIJCOSX-MP2 using nNSD@Grid. On the other hand, the nNSD@Grid computing time performance is 10 times slower for OO-RIJCOSX-MP2 than RIJCOSX-MP2. In closer inspection of the contributions from various integral computations, we found that 99% of the computing time was invested in the MP2 module for OO-RIJCOSX-MP2 compared to 92.7% for RIJCOSX-MP2 (not shown). The longer time investment in the MP2 module is a result of orbital optimization in addition to the calculation of the canonical MP2 excitations. The longer computing time in the MP2 module hampers the efficiency of nNSD@Grid as only one processor is used to solve each OO-RIJCOSX-MP2 aNFD.

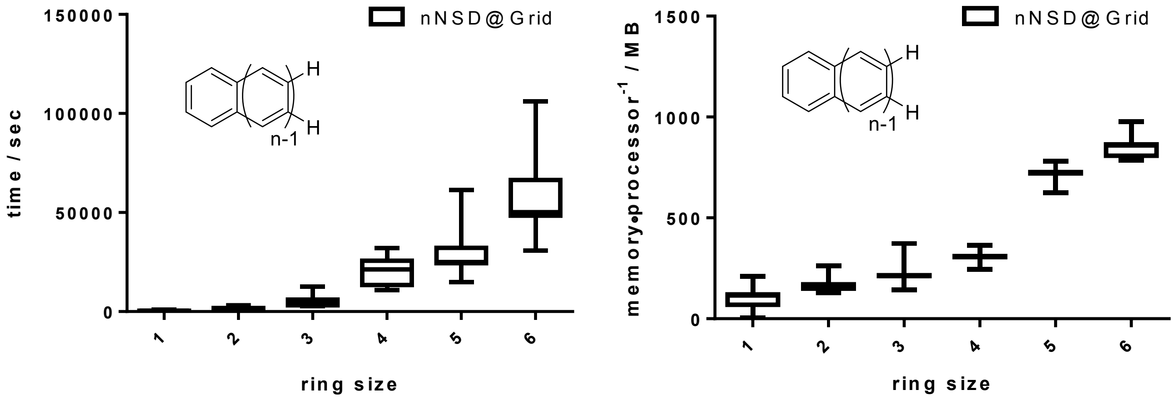


Figure 7. Computing time (left) and memory usage per processor (right) for various molecules of linearly fused benzene rings by nNSD@Grid using OO-RIJCOSX-MP2 are shown.

4. PRACTICAL EXAMPLES

a. Hessian evaluation using nNSD@Grid for a large open-shell transition-metal-containing complex

E-Ag$_2$(^{t}bu$_3$tpy)$_2$OTf(NSO$_3$-2,6-dFPh) (^{t}bu$_3$tpy = 4-,4'-,4"-tri-tert-butyl-2-,2':6'-2"-terpyridine, OTf = triflate, and dFPh = difluorophenyl), whose structure is shown in Figure 8, is a proposed metal-nitrene reactive intermediate in silver-catalyzed intermolecular nitrene transfer reactions with a triplet electronic ground state.[43] Such complexes are important for atom-efficient addition of nitrogen functionalities to organic molecules. With SARC-def2-TZVPP on Ag and SARC-def2-SVP on the rest of the elements adding up to 2705 GFs and 7094 DFGFs, nNSD@Grid took 181-239-257-286-483 minutes for E-$Ag_2(^t bu_3tpy)_2OTf(NSO_3$-2,6-dFPh) with 432-686-728-814-981 MB memory usage per processor. In contrast, an aNSD calculation completed in 3730 minutes using 12 processors and 4343 MB per processor. Overall, nNSD@Grid is 7.7 times faster in computing time and 4.4 times less memory intense than aNSD using 12 processors on a molecule consisting of 156 atoms, among them two transition metal atoms.

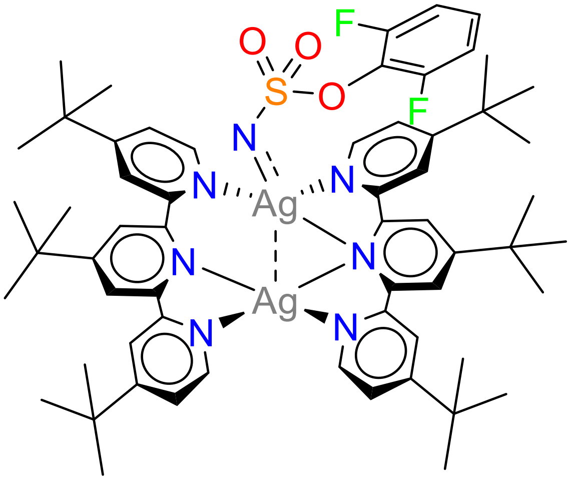


Figure 8. Structure of E-$Ag_2(^t bu_3tpy)_2OTf(NSO_3$-2,6-dFPh), a proposed reactive intermediate in silver-catalyzed nitrene transfer, containing 156 atoms with two heavy metals.

b. Hessian evaluation using nNSD@Grid for a reactive catalytic intermediate with a multiconfigurational ground state

The 91 atom $^2[Rh_2(esp)_2\text{-nitrene}]^+$ (esp = α, α, α', α'-tetramethyl-1,3-benzenedipropanoate and nitrene = NSO_3CH_3) complex, shown in Figure 9, is the proposed reactive intermediate formed by one of the leading transition metal catalysts for nitrene transfer. This complex has been shown to be multiconfigurational with three major configuration state functions that contribute to the ground state CASSCF wavefunction: 1) A triplet nitrene ($S = 1$) antiferromagnetically coupled to $[Rh_2(esp)_2]^+$ ($S = 1/2$), 2) a nitrene radical anion ($S = 1/2$) bound to a closed shell $[Rh_2(esp)_2]^{2+}$ unit, and 3) a nitrene radical cation ($S = 1/2$) bound to a closed shell $Rh_2(esp)_2$ unit (Figure 9).[35] On this complex system, aNSD experiences difficulty due to the instability in solving the CP-SCF equations; no convergence (first of the seven batches) was conceivable after 80 iterations, which spent 1440 minutes on 12 processors. In contrast, nNSD@Grid on $^2[Rh_2(esp)_2NSO_3CH_3]^+$ completes in 95-134-178-239-878 minutes.

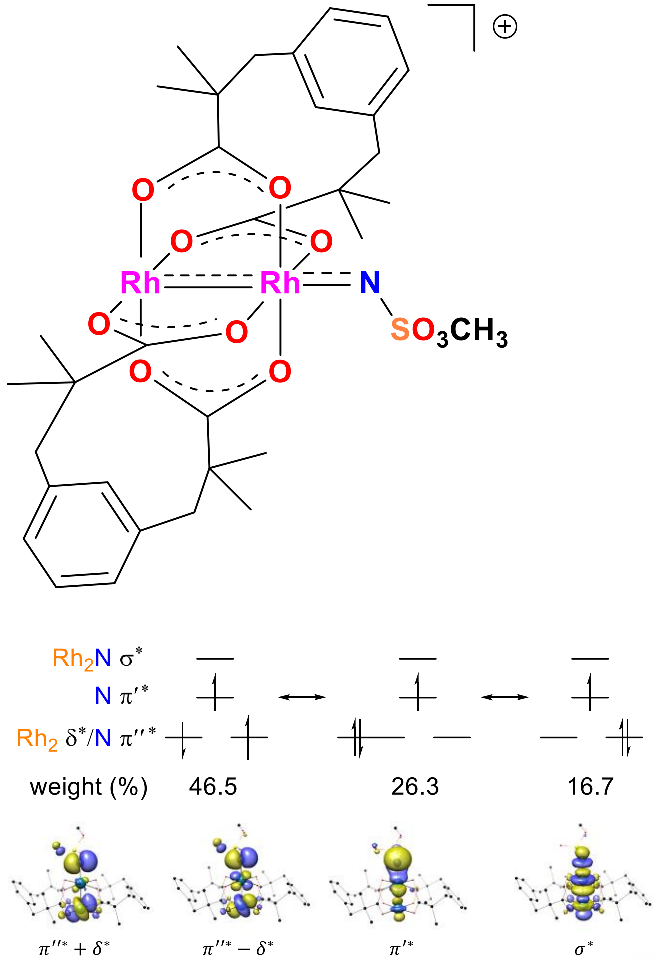


Figure 9. Structure of (top), the three dominant CASSCF configurations of (bottom), and the four frontier orbitals (lower right) in $^2[Rh_2(esp)_2NSO_3CH_3]^+$.

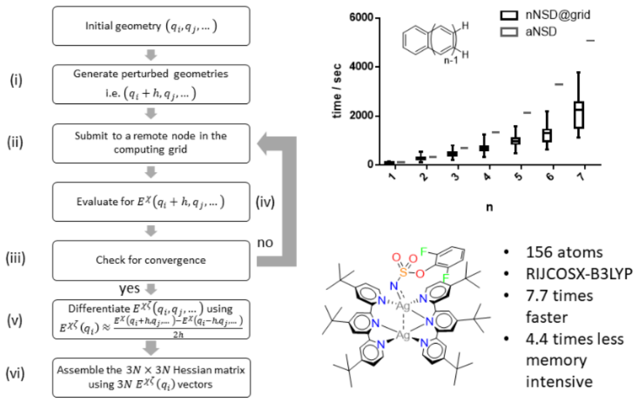
5. CONCLUSION AND OUTLOOK

Continuous developments in computing hardware, optical fibers, the Internet, and protocols for internetwork communication have enabled us to develop a new computing-grid-enabled numerical nuclear second derivative method for computing nuclear Hessian matrices: nNSD@Grid. nNSD@Grid can outperform aNSD in terms of computing time and the convergence behavior for B3LYP (and other similar or lower scaling density functionals) and of treatable system size using less memory for higher level computational methods. For theoretical methods without an aNSD expression, an excessive number of processors is required to achieve the temporal performance of nNSD@Grid on a conventional HPC.

To our knowledge, the study reported herein is the first example of the utilization of grid computing to improve the applicability of NSD to various chemical systems using a broad range of theoretical methods. We envision future developments of computing-grid-enabled quantum chemical methods, especially for methods that are embarrassingly parallel that can readily be adapted to grid computing.

ACKNOWLEDGMENTS

J.F.B. thanks the Center for Selective C−H Functionalization supported by the National Science Foundation (CHE-1700982). The computational facility at Madison is supported in part by National Science Foundation Grant CHE-0840494 and at the UW—Madison Center For High Throughput Computing (CHTC) in the Department of Computer Sciences. The CHTC is supported by UW—Madison, the Advanced Computing Initiative, the Wisconsin Alumni Research Foundation, the Wisconsin Institutes for Discovery, and the National Science Foundation, and is an active member of the Open Science Grid, which is supported by the National Science Foundation award 1148698 and the U.S. Department of Energy's Office of Science.


Initial geometry $(q_i, q_j, \ldots)$
(i)
Generate perturbed geometries i.e. $(q_i + h, q_j, \ldots)$
(ii)
Submit to a remote node in the computing grid
Evaluate for $E^{\chi}(q_i + h, q_j, \ldots)$
(iv)
no
(iii)
Check for convergence
yes
(v)
Differentiate $E^{\chi\zeta}(q_i, q_j, \ldots)$ using $E^{\chi\zeta}(q_i) \approx \frac{E^{\chi}(q_i+h,q_j,\ldots)-E^{\chi}(q_i-h,q_j,\ldots)}{2h}$
(vi)
Assemble the $3N \times 3N$ Hessian matrix using $3N$ $E^{\chi\zeta}(q_i)$ vectors
6000
4000
2000
0
time / sec
H
H
n-1
nNSD@grid
aNSD
1
2
3
4
5
6
7
n
O
O
F
S
O
F
N
N
Ag
N
N
N
Ag
N
N
N
• 156 atoms
• RIJCOSX-B3LYP
• 7.7 times faster
• 4.4 times less memory intensive

1. (a) Handy, N. C.; Tozer, D. J.; Laming, G. J.; Murray, C. W.; Amos, R. D., Analytic Second Derivatives of the Potential Energy Surface. *Isr. J. Chem.* **1993,** *33*, 331-344; (b) Deglmann, P.; Furche, F.; Ahlrichs, R., An efficient implementation of second analytical derivatives for density functional methods. *Chem. Phys. Lett.* **2002,** *362*, 511-518; (c) Johnson, B. G.; Fisch, M. J., An implementation of analytic second derivatives of the gradient-corrected density functional energy. *J. Chem. Phys.* **1994,** *100*, 7429-7442; (d) Komornicki, A.; Fitzgerald, G., Molecular gradients and hessians implemented in density functional theory. *J. Chem. Phys.* **1993,** *98*, 1398-1421; (e) Fournier, R., Second and third derivatives of the linear combination of gaussian type orbitals–local Spin density energy. *J. Chem. Phys.* **1990,** *92*, 5422-5429; (f) Dunlap, B. I.; Andzelm, J., Second derivatives of the Local-density-functional total energy when the local potential is fitted. *Phys. Rev. A* **1992,** *45*, 81-87.

2. (a) Harrison, R. J.; Fitzgerald, G. B.; Laidig, W. D.; Bartlett, R. J., Analytic MBPT(2) second derivatives. *Chem. Phys. Lett.* **1986,** *124*, 291-294; (b) Handy, N. C.; Amos, R. D.; Gaw, J. F.; Rice, J. E.; Simandiras, E. D., The Elimination of Singularities in Derivative Calculations. *Chem. Phys. Lett.* **1985,** *120*, 151-158.

3. Koch, H.; Jensen, H. J. A.; Jorgensen, P.; Helgaker, T.; Scuseria, G. E.; Schaefer, H. F., Coupled Cluster Energy Derivatives - Analytic Hessian for the Closed-Shell Coupled Cluster Singles and Doubles Wave-Function - Theory and Applications. *J. Chem. Phys.* **1990,** *92*, 4924-4940.

4. (a) Szalay, P. G.; Gauss, J.; Stanton, J. F., Analytic UHF-CCSD(T) second derivatives: implementation and application to the calculation of the vibration-rotation interaction constants of NCO and NCS. *Theor. Chem. Acc.* **1998,** *100*, 5-11; (b) Stanton, J. F.; Gauss, J., Analytic second derivatives in high-order many-body perturbation and coupled-cluster theories: computational considerations and applications. *Int. Rev. Phys. Chem.* **2000,** *19*, 61-95; (c) Gauss, J.; Stanton, J. F., Analytic CCSD(T) second derivatives. *Chem. Phys. Lett.* **1997,** *276*, 70-77.

5. Camp, R. N.; King, H. F.; Mciver, J. W.; Mullally, D., Analytical Force-Constants for Mcscf Wave-Functions. *J. Chem. Phys.* **1983,** *79*, 1088-1089.

6. (a) Peng, H. W., Perturbation theory for the self-consistent field. *Proc. R. Soc. A* **1941,** *178*, 499-505; (b) Allen, L. C., Hartree-Fock Equations with a Perturbing Field. *Phys. Rev.* **1960,** *118*, 167-175.

7. White, C. A.; Johnson, B. G.; Gill, P. M. W.; Headgordon, M., The Continuous Fast Multipole Method. *Chem. Phys. Lett.* **1994,** *230*, 8-16.

8. Neese, F.; Wennmohs, F.; Hansen, A.; Becker, U., Efficient, approximate and parallel Hartree-Fock and hybrid DFT calculations. A 'chain-of-spheres' algorithm for the Hartree-Fock exchange. *Chem. Phys.* **2009,** *356*, 98-109.

9. (a) Baerends, E. J.; Ellis, D. E.; Ros, P., Self-consistent molecular Hartree-Fock-Slater calculations - I. The computational procedure. *Chem. Phys.* **1973,** *2*, 41-51; (b) Dunlap, B. I.; Connolly, J. W. D.; Sabin, J. R., On some approximations in applications of Xα theory. *J. Chem. Phys.* **1979,** *71*, 3396-3402.

10. Manzer, S.; Horn, P. R.; Mardirossian, N.; Head-Gordon, M., Fast, accurate evaluation of exact exchange: The occ-RI-K algorithm. *J. Chem. Phys.* **2015,** *143*.

11. White, C. A.; HeadGordon, M., A J matrix engine for density functional theory calculations. *J. Chem. Phys.* **1996,** *104*, 2620-2629.

12. Friesner, R. A., New Methods for Electronic-Structure Calculations on Large Molecules. *Annu. Rev. Phys. Chem.* **1991,** *42*, 341-367.

13. (a) Bykov, D.; Petrenko, T.; Izsak, R.; Kossmann, S.; Becker, U.; Valeev, E.; Neese, F., Efficient implementation of the analytic second derivatives of Hartree-Fock and hybrid DFT energies: a detailed analysis of different approximations. *Mol. Phys.* **2015,** *113*, 1961-1977; (b) Kossmann, S.; Neese, F., Comparison of two efficient approximate Hartee-Fock approaches. *Chem. Phys. Lett.* **2009,** *481*, 240-243; (c) Rebolini, E.; Izsak, R.; Reine, S. S.; Helgaker, T.; Pedersen, T. B., Comparison of Three Efficient Approximate Exact-Exchange Algorithms: The Chain-of-Spheres Algorithm, Pair-Atomic Resolution-of-the-Identity Method, and Auxiliary Density Matrix Method. *J. Chem. Theory Comput.* **2016,** *12*, 3514-3522.

14. Almlof, J.; Faegri, K.; Korsell, K., Principles for a Direct Scf Approach to Lcao-Mo Abinitio Calculations. *J. Comput. Chem.* **1982,** *3*, 385-399.

15. Häser, M.; Ahlrichs, R., Improvements on the direct SCF method. *J. Comput. Chem.* **1989,** *10*, 104-111.

16. (a) Furlani, T. R.; King, H. F., Implementation of a Parallel Direct Scf Algorithm on Distributed-Memory Computers. *J. Comput. Chem.* **1995,** *16*, 91-104; (b) Harrison, R. J.; Guest, M. F.; Kendall, R. A.; Bernholdt, D. E.; Wong, A. T.; Stave, M.; Anchell, J. L.; Hess, A. C.; Littlefield, R. J.; Fann, G. L.; Nieplocha, J.; Thomas, G. S.; Elwood, D.; Tilson, J. L.; Shepard, R. L.; Wagner, A. F.; Foster, I. T.; Lusk, E.; Stevens, R., Toward high-performance computational chemistry .2. A scalable self-consistent field program. *J. Comput. Chem.* **1996,** *17*, 124-132; (c) Foster, I. T.; Tilson, J. L.; Wagner, A. F.; Shepard, R. L.; Harrison, R. J.; Kendall, R. A.; Littlefield, R. J., Toward high-performance computational chemistry .1. Scalable Fock matrix construction algorithms. *J. Comput. Chem.* **1996,** *17*, 109-123; (d) Brode, S.;

Horn, H.; Ehrig, M.; Moldrup, D.; Rice, J. E.; Ahlrichs, R., Parallel Direct Scf and Gradient Program for Workstation Clusters. *J. Comput. Chem.* **1993,** *14*, 1142-1148; (e) Harrison, R. J.; Shepard, R., Ab-Initio Molecular Electronic-Structure on Parallel Computers. *Annu. Rev. Phys. Chem.* **1994,** *45*, 623-658.

17. (a) Baker, J.; Wolinski, K.; Malagoli, M.; Pulay, P., Parallel implementation of Hartree-Fock and density functional theory analytical second derivatives. *Mol. Phys.* **2004,** *102*, 2475-2484; (b) Korambath, P. P.; Kong, J.; Furlani, T. R.; Head-Gordon, M., Parallelization of analytical Hartree-Fock and density functional theory Hessian calculations. Part I: parallelization of coupled-perturbed Hartree-Fock equations. *Mol. Phys.* **2002,** *100*, 1755-1761.

18. Mitin, A. V.; Baker, J.; Wolinski, K.; Pulay, P., Parallel stored-integral and semidirect Hartree-Fock and DFT methods with data compression. *J. Comput. Chem.* **2003,** *24*, 154-160.

19. (a) Dudley, T. J.; Olson, R. M.; Schmidt, M. W.; Gordon, M. S., Parallel coupled perturbed CASSCF equations and analytic CASSCF second derivatives. *J. Comput. Chem.* **2006,** *27*, 352-362; (b) Janowski, T.; Ford, A. R.; Pulay, P., Parallel calculation of coupled cluster singles and doubles wave functions using array files. *J. Chem. Theory Comput.* **2007,** *3*, 1368-1377; (c) Ishimura, K.; Pulay, P.; Nagase, S., New parallel algorithm for MP2 energy gradient calculations. *J. Comput. Chem.* **2007,** *28*, 2034-2042.

20. (a) Cawkwell, M. J.; Sanville, E. J.; Mniszewski, S. M.; Niklasson, A. M. N., Computing the Density Matrix in Electronic Structure Theory on Graphics Processing Units. *J. Chem. Theory Comput.* **2012,** *8*, 4094-4101; (b) Ufimtsev, I. S.; Martinez, T. J., Quantum chemistry on graphical processing units. 1. Strategies for two-electron integral evaluation. *J. Chem. Theory Comput.* **2008,** *4*, 222-231; (c) Yasuda, K., Two-electron integral evaluation on the graphics processor unit. *J. Comput. Chem.* **2008,** *29*, 334-342; (d) Yasuda, K., Accelerating density functional calculations with graphics processing unit. *J. Chem. Theory Comput.* **2008,** *4*, 1230-1236; (e) Ufimtsev, I. S.; Martinez, T. J., Quantum Chemistry on Graphical Processing Units. 2. Direct Self-Consistent-Field Implementation. *J. Chem. Theory Comput.* **2009,** *5*, 1004-1015.

21. (a) Hohenstein, E. G.; Luehr, N.; Ufimtsev, I. S.; Martinez, T. J., An atomic orbital-based formulation of the complete active space self-consistent field method on graphical processing units. *J. Chem. Phys.* **2015,** *142*; (b) Bhaskaran-Nair, K.; Ma, W. J.; Krishnamoorthy, S.; Villa, O.; van Dam, H. J. J.; Apra, E.; Kowalski, K., Noniterative Multireference Coupled Cluster Methods on Heterogeneous CPU-GPU Systems. *J. Chem. Theory Comput.* **2013,** *9*, 1949-1957; (c) Apra, E.; Kowalski, K., Implementation of High-Order Multireference Coupled-Cluster Methods on Intel Many Integrated Core Architecture. *J. Chem. Theory Comput.* **2016,** *12*, 1129-1138; (d) DePrince, A. E.; Hammond, J. R., Coupled Cluster Theory on Graphics Processing Units I. The Coupled Cluster Doubles Method. *J. Chem. Theory Comput.* **2011,** *7*, 1287-1295; (e) Vogt, L.; Olivares-Amaya, R.; Kermes, S.; Shao, Y.; Amador-Bedolla, C.; Aspuru-Guzik, A., Accelerating resolution-of-the-identity second-order Moller-Plesset quantum chemistry calculations with graphical processing units. *J. Phys. Chem. A* **2008,** *112*, 2049-2057.

22. Weber, V.; Daul, C., Improved coupled perturbed Hartree-Fock and Kohn-Sham convergence acceleration. *Chem. Phys. Lett.* **2003,** *370*, 99-105.

23. Liu, K. Y.; Liu, J.; Herbert, J. M., Accuracy of finite-difference harmonic frequencies in density functional theory. *J. Comput. Chem.* **2017,** *38*, 1678-1684.

24. De Roure, D.; Jennings, N. R.; Shadbolt, N. R., The Semantic Grid: Past, present, and future. *P Ieee* **2005,** *93*, 669-681.

25. (a) Pordes, R. P., D.; Kramer, B.; Olson, D.; Livny, M.; Roy, A.; Avery, P.; Blackburn, K.; Wenaus, T.; Wurthwein, F.; Foster, I.; Gardner, R.; Wilde, M.; Blatecky, A.; McGee, J.; Quick, R., The open science grid. *J. Phys. Conf. Ser.* **2007,** *78*, 012057; (b) Garzoglio, G.; Alderman, I.; Altunay, M.; Ananthakrishnan, R.; Bester, J.; Chadwick, K.; Ciaschini, V.; Demchenko, Y.; Ferraro, A.; Forti, A.; Groep, D.; Hesselroth, T.; Hover, J.; Koeroo, O.; La Joie, C.; Levshina, T.; Miller, Z.; Packard, J.; Sagehaug, H.; Sergeev, V.; Sfiligoi, I.; Sharma, N.; Siebenlist, F.; Venturi, V.; Weigand, J., Definition and Implementation of a SAML-XACML Profile for Authorization Interoperability Across Grid Middleware in OSG and EGEE. *J Grid Comput* **2009,** *7*, 297-307.

26. (a) Becke, A. D., Density-Functional Exchange-Energy Approximation with Correct Asymptotic-Behavior. *Phys. Rev. A* **1988,** *38*, 3098-3100; (b) Lee, C. T.; Yang, W. T.; Parr, R. G., Development of the Colle-Salvetti Correlation-Energy Formula into a Functional of the Electron-Density. *Phys. Rev. B* **1988,** *37*, 785-789.

27. Franke, R.; Van Wullen, C., First-order relativistic corrections to MP2 energy from standard gradient codes: Comparison with results from density functional theory. *J. Comput. Chem.* **1998,** *19*, 1596-1603.

28. (a) Pantazis, D. A.; Chen, X. Y.; Landis, C. R.; Neese, F., All-electron scalar relativistic basis sets for third-row transition metal atoms. *J. Chem. Theory Comput.* **2008,** *4*, 908-919; (b) Weigend, F.; Ahlrichs, R., Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. *Phys. Chem. Chem. Phys.* **2005,** *7*, 3297-3305.

29. (a) Eichkorn, K.; Weigend, F.; Treutler, O.; Ahlrichs, R., Auxiliary basis sets for main row atoms and transition metals and their use to approximate Coulomb potentials. *Theor. Chem. Acc.* **1997,** *97*, 119-124; (b) Weigend, F.; Haser, M.; Patzelt, H.; Ahlrichs, R., RI-MP2: optimized auxiliary basis sets and demonstration of efficiency. *Chem. Phys. Lett.* **1998,** *294*, 143-152.

30. Grimme, S., Semiempirical hybrid density functional with perturbative second-order correlation. *J. Chem. Phys.* **2006,** *124*.

31. Neese, F., Software update: the ORCA program system, version 4.0. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2017**, e1327-n/a.

32. Neese, F., The ORCA program system. *Wiley Interdiscip Rev Comput Mol Sci* **2012,** *2*, 73-78.

33. Thain, D.; Tannenbaum, T.; Livny, M., Distributed computing in practice: the Condor experience. *Concurr Comp-Pract E* **2005,** *17*, 323-356.

34. Couvares, P. K., T.; Roy, A.; Weber, J.; Wenger, K., *Workflow in Condor*. Springer Press: 2007.

35. Varela-Alvarez, A.; Yang, T. H.; Jennings, H.; Kornecki, K. P.; Macmillan, S. N.; Lancaster, K. M.; Mack, J. B. C.; Du Bois, J.; Berry, J. F.; Musaev, D. G., Rh_2(II,III) Catalysts with Chelating Carboxylate and Carboxamidate Supports: Electronic Structure and Nitrene Transfer Reactivity. *J. Am. Chem. Soc.* **2016,** *138*, 2327-2341.

36. Kawamura, T.; Katayama, H.; Nishikawa, H.; Yamabe, T., Ligand Dependence of Electronic Configuration of the Rh-Rh Bond in Rh_2^{5+} Complexes as Studied by Electron-Spin Resonance and Electrochemistry. *J. Am. Chem. Soc.* **1989,** *111*, 8156-8160.

37. Pulay, P., Convergence Acceleration of Iterative Sequences - the Case of Scf Iteration. *Chem. Phys. Lett.* **1980,** *73*, 393-398.

38. Schlegel, H. B.; Binkley, J. S.; Pople, J. A., First and second derivatives of two electron integrals over Cartesian Gaussians using Rys polynomials. *J. Chem. Phys.* **1984,** *80*, 1976-1981.

39. Pople, J. A.; Krishnan, R.; Schlegel, H. B.; Binkley, J. S., Derivative Studies in Hartree-Fock and Moller-Plesset Theories. *Int. J. Quantum Chem.* **1979,** *16*, 225-241.

40. Goerigk, L.; Grimme, S., Double-hybrid density functionals. *Wiley Interdiscip Rev Comput Mol Sci* **2014,** *4*, 576-600.

41. Ziolkowski, M.; Jansik, B.; Kjaergaard, T.; Jorgensen, P., Linear scaling coupled cluster method with correlation energy based error control. *J. Chem. Phys.* **2010,** *133*.

42. Neese, F.; Schwabe, T.; Kossmann, S.; Schirmer, B.; Grimme, S., Assessment of Orbital-Optimized, Spin-Component Scaled Second-Order Many-Body Perturbation Theory for Thermochemistry and Kinetics. *J. Chem. Theory Comput.* **2009,** *5*, 3060-3073.

43. Dolan, N. S.; Scamp, R. J.; Yang, T.; Berry, J. F.; Schomaker, J. M., Catalyst-Controlled and Tunable, Chemoselective Silver-Catalyzed Intermolecular Nitrene Transfer: Experimental and Computational Studies. *J. Am. Chem. Soc.* **2016,** *138*, 14658-14667.