PCCP

Check for updates

Economical quasi-Newton unitary optimization of electronic orbitals†

Samuel A. Slattery, [ID] [a] Kshitijkumar A. Surjuse, [ID] [a] Charles C. Peterson,[b] Deborah A. Penchoff [ID] [c] and Edward F. Valeev [ID] *[a]

We present an efficient quasi-Newton orbital solver optimized to reduce the number of gradient evaluations and other computational steps of comparable cost. The solver optimizes orthogonal orbitals by sequences of unitary rotations generated by the (preconditioned) limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm equipped with trust-region step restriction. The low-rank structure of the L-BFGS inverse Hessian is exploited when solving the trust-region problem. The efficiency of the proposed "Quasi-Newton Unitary Optimization with Trust-Region" (QUOTR) solver is compared to that of the standard Roothaan–Hall approach accelerated by the Direct Inversion of Iterative Subspace (DIIS), and other exact and approximate Newton solvers for mean-field (Hartree–Fock and Kohn–Sham) problems.

1 Introduction

Orbital optimization is a fundamental ingredient of the electronic structure methods at all levels of approximation, from 1-body models (Hartree–Fock (HF), Kohn–Sham density functional theory (KS DFT), collectively known as the self-consistent field (SCF) method[1]‡), to many-body methods (*e.g.*, multiconfiguration self-consistent field (MCSCF)). Despite the long history of innovation,[2–29] development of improved orbital optimizers continues to this day.[30–35] Although the relevant functionals of the orbitals are nonconvex, and global and local nonconvex optimization is NP-hard,[36,37] it is known that many practical orbital optimization problems are easily solved using existing heuristics. For the crucial HF/KS SCF use case, the most popular solvers in the molecular context are based on the Roothaan–Hall (RH) iterative diagonalization of the Fock matrix[2,38] augmented by convergence accelerators such as Anderson mixing[39] or the closely related direct inversion in the iterative subspace (DIIS) method,[10,13,40,41] as well as others.[42] However, several issues plague the efficient RH/DIIS heuristics:

• for systems with complex electronic structure (such as molecules far from equilibrium, open-shell systems,[32,43] and

systems with small HOMO–LUMO gaps[28]) convergence will be slow,[30] erratic, or nonexistent,[31,44]

• the use of diagonalization produces canonical orbitals whose lack of localization makes them incompatible with fast algorithms for the Fock matrix construction (*e.g.*, using local or sparse density fitting[45–47]),

• applications to large systems and/or in non-LCAO representations can be bottlenecked by the $\mathcal{O}(N^3)$ cost of diagonalization,[17,18,26]

• locating non-Aufbau (*e.g.*, excited state) solutions is possible[48] but is not robust, and

• even in favorable cases the convergence rate is linear[49,50] (*i.e.*, the error is reduced by approximately the same factor each iteration) or perhaps slightly better when accelerated with DIIS;[51] this is slower than the quadratic convergence exhibited by, *e.g.*, the Newton method.[52]

The lack of convergence guarantees is probably the most severe of these in practice. Extensions of the standard RH/DIIS heuristics have been devised to improve the robustness[40,53,54] but for challenging cases the user is expected to control the many heuristic solver control parameters that help the convergence (level shift, damping, *etc.*).

Orbital optimizer solvers that rely on direct energy minimization can address some/all of these concerns and thus have a long history of development.[3,5,6,8,9,11,12,14–19,21–29,55] In the molecular mean-field context direct minimization SCF solvers have long been employed as the recommended alternative in the case of convergence problems, used in combination with RH/DIIS to gain superlinear convergence, and to enable reduced-scaling SCF approaches.[26,28] Nevertheless, RH/DIIS remains the default SCF solver, not due to its formal advantages, but due to its superior efficiency. This may be puzzling since

[a] Department of Chemistry, Virginia Tech, Blacksburg, VA 24061, USA.
E-mail: efv@vt.edu
[b] Office of Advanced Research Computing, University of California, Los Angeles, CA 90095, USA
[c] UT Innovative Computing Laboratory, University of Tennessee, Knoxville, TN 37996, USA

‡ Sometimes SCF is also used to denote a specific class of solvers to the Hartree–Fock/Kohn–Sham equations, which goes counter to the original use of the term.

direct minimization solvers are often demonstrated to converge in as few as (or fewer) *iterations* than RH/DIIS.[21,35] However, the number of iterations is a misleading figure since each update of the orbitals or density matrix may involve multiple energy/gradient evaluations or solving similarly expensive subproblems (such as multiplication of a trial orbital rotation by the orbital Hessian). In other words, the number of gradient evaluations (Fock build equivalents, N_F) in a direct minimization solver is typically significantly greater than the number of iterations (N_I), whereas in RH/DIIS they are equal. Thus the latter typically involves significantly fewer Fock matrix evaluations, which in most practical applications determines the overall cost.

The objective of this work is to design a quasi-Newton orbital optimizer that minimizes the number of gradient evaluations (and its equivalents) to be as competitive with RH/DIIS as possible, and as robust as possible without the need to adjust the control parameters. Our "Quasi-Newton Unitary Optimization with Trust-Region" (QUOTR) solver uses preconditioned limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm[53] step-restricted by trust-region (TR) and leverages the inverse L-BFGS Hessian's low-rank structure to efficiently solve the trust-region update problem.[56]

The rest of the manuscript is structured as follows. In Section 2 we briefly review the general classes of SCF solvers before describing the theoretical aspects of QUOTR. Next, the implementation of QUOTR is discussed in Section 3. In Section 4 we display solver performance statistics for a standard set of chemical systems and make a comparison to a method that uses information from the "exact" Hessian. Additionally, in Section 4 we illustrate the utility of QUOTR for several prototypical problems where RH/DIIS and other SCF solvers struggle, such as a system with vanishing HOMO–LUMO gap as well as select d- and f-element containing systems. In Section 5 we summarize our findings.

2 Formalism

2.1 Overview of SCF solver approaches

All SCF methods attempt to iteratively minimize the electronic energy $E(\mathbf{x})$ or its Lagrangian counterpart, where $\mathbf{x}$ is a set of independent parameters defining the particular method. In practice the minimum is determined by using the energy, its gradient $\mathbf{g}$, and optionally the Hessian $\mathbf{B}$. Starting with an initial (guess) set of parameters $\mathbf{x}^{(0)}$ SCF solvers construct improved parameter values using the current energy and its derivatives, (optionally) their values from previous iterations (histories), as well as any optional additional parameters and their histories:

$$\mathbf{x}^{(k+1)} = f(\{\mathbf{x}^{(k)}\}, \{E^{(k)}\}, \{\mathbf{g}^{(k)}\}, \ldots) \tag{1}$$

The SCF solvers differ in how they construct the update in eqn (1); unfortunately, it is not possible to systematically classify the solvers since in the vast majority of cases $f()$ is an *algorithm*, not a simple function. Thus here we only focus on essential common elements of all SCF solvers.

Most solvers split the update problem (1) into 2 subproblems by defining the parameter update,

$$\mathbf{s}^{(k)} \equiv \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} = \alpha^{(k)}\mathbf{p}^{(k)} \tag{2}$$

in terms of a search direction $\mathbf{p}^{(k)}$ and a step size $\alpha^{(k)}$, each of which has its own prescription similar to eqn (1)

$$\mathbf{p}^{(k)} = g(\{\mathbf{x}^{(k)}\}, \{E^{(k)}\}, \{\mathbf{g}^{(k)}\}, \ldots), \tag{3}$$

$$\alpha^{(k)} = h(\{\mathbf{x}^{(k)}\}, \{E^{(k)}\}, \{\mathbf{g}^{(k)}\}, \ldots). \tag{4}$$

The need to control the step size is common to all SCF solvers due to the fundamental nonlinearity of the energy function. Therefore even solvers that do not employ eqn (2), such as RH/DIIS, still introduce *ad hoc* ways to control the step size by level shifting, damping, and other means of step restriction.

The simplest "2-step" solver is the steepest descent (SD) method[3] in which the search direction $\mathbf{p}^{(k)}$ is opposite to the current gradient $\mathbf{g}^{(k)}$:

$$\mathbf{p}^{(k)} \stackrel{\text{SD}}{=} -\frac{\mathbf{g}^{(k)}}{\|\mathbf{g}^{(k)}\|} \tag{5}$$

Unfortunately, although the SD method is guaranteed to converge to a nearby minimum, the plain SD variant converges very slowly;[57,58] this can be rationalized by comparing it to the (exact) Newton step:

$$\mathbf{s}^{(k)\,\text{Newton}} = -\left(\mathbf{B}^{(k)}\right)^{-1}\mathbf{g}^{(k)}. \tag{6}$$

Hessian $\mathbf{B}$ is a diagonally-dominant matrix with a large (and growing with the basis set size) condition number. Luckily it is relatively simple to construct an effective approximation to the Hessian; a particularly popular way is to use only the 1-electron terms in the Hessian, $\mathbf{B}_{1e}$. Approximate Hessians can then be used for preconditioning SD (using the 1-electron Hessian for preconditioning is also known as the "energy weighted steepest descent" method[5,6,8]) by replacing $\mathbf{g}^{(k)}$ in eqn (5) with the preconditioned gradient:

$$\tilde{\mathbf{g}}^{(k)} \equiv (\mathbf{B}_{1e}^{(k)})^{-1}\mathbf{g}^{(k)}. \tag{7}$$

The RH method can be viewed as a simplified version of preconditioned SD, due to its step being exactly the negative of the gradient preconditioned by the 1-electron Hessian:[24,28]

$$\mathbf{s}^{(k)} \stackrel{\text{RH}}{=} -\tilde{\mathbf{g}}^{(k)}. \tag{8}$$

More sophisticated prescriptions for direction include the conjugate gradient (CG) method[18,20,22,27,29] in which history is limited to the information about the current and previous iteration. Of course, the use of preconditioning is mandatory with CG just as with SD. Unfortunately neither SD nor CG, even with an approximate preconditioner, lead to an optimal convergence rate near the minimum. Thus the most efficient solvers utilize exact or approximate Hessians near the minimum. The time-determining step of such models usually involves direct evaluation of the action of exact (or approximated) Hessian onto a trial step, at a cost similar to the cost of the gradient evaluation (*i.e.*, the Fock matrix evaluation in the mean-field case).[13,16,17] Although it is possible to apply the straightforward Newton method using the exact Hessian when sufficiently close to the minimum,[9] to be able to use the exact Hessian further away from the minimum requires some form of step restriction.

6558 | *Phys. Chem. Chem. Phys.*, 2024, **26**, 6557–6573

This journal is © the Owner Societies 2024

The popular augmented Hessian (AH)[31] method can be viewed as a Newton method with optimally restricted steps; it can also be viewed as a quasi-Newton method in which an approximate (level-shifted) Hessian is used. The diverse family of quasi-Newton methods each use approximate Hessians of some form, often generated from information contained in the gradients and steps of the previous iterations.[52] The quasi-Newton idea has been used in MCSCF for a long time,[59] and the most commonly employed approximation in SCF is some form of the BFGS algorithm.[14,15,19,21,29,33] The BFGS method has recently been used with success in the MCSCF context[60] and the selected configuration interaction (CI) context.[61]

Although some solvers compute the step length separately from the direction, more sophisticated approaches fuse step restriction deeper into the step computation. Indeed, when an underlying quadratic model of the energy exists, it is not natural to simply perform a line search toward the (unrestricted) minimum of the model, considering that the model is known to be locally accurate in all directions. The alternative concept of searching for the minimum of a model in all directions, but restricting the step size to some maximum value, is the key idea of the trust-region (TR) method.[23–26,28,31,55,62,63] Two important aspects of any TR method are: how the trust-region is updated between iterations, and how the trust-region problem is solved for the step. The update method that is commonly used is based on an algorithm developed by Fletcher,[64] and one of the first true TR applications in quantum chemistry used it in the context of MCSCF.[62] A common occurrence of the TR problem in quantum chemistry is within the framework of the AH method; due to the use of full (level-shifted) Hessian in AH the cost of the TR solve is similar to the cost of the unrestricted step.[31] Here we use the TR method in the context of the L-BFGS method which allows us to exploit the low-rank structure of the L-BFGS Hessian to essentially eliminate the extra cost of using the TR method.[56]

2.2 QUOTR: quasi-Newton unitary optimization with trust-region

Our direct minimization SCF solver is a preconditioned quasi-Newton (L-BFGS) solver with TR step restriction. Although its aspects are similar to prior SCF solvers, there are several novel elements:

Table 1 User-controllable parameters of the QUOTR solver

Description	Symbol	Value
Energy convergence threshold	t_{ce}	10^{-9}
Gradient convergence threshold	t_{cg}	10^{-5}
L-BFGS start threshold	t_b	0.1
Max history size	m	8
Regularizer threshold	t_r	0.25
History keep threshold	t_h	10^{-5}
Exponential tolerance	t_e	10^{-15}
Compare zero threshold	t_0	10^{-11}
Line search fitting range shrink factor	$\alpha_{fit,shrink}$	1/2
Minimum TR tolerance	t_t	10^{-10}
TR step accept threshold	τ_1	0
TR shrink threshold	τ_2	0.25
TR expand threshold	τ_3	0.75
TR shrink factor	η_1	0.25
TR shrink by step factor	η_2	0.5
TR expand check factor	η_3	0.8
TR expand by factor	η_4	2.0

● The optimization is parameterized with a consistent "reference" (epoch) MO basis allowing use of the exact gradient with minimal computation after the Fock matrix is constructed.

● The preconditioner is updated only on some iterations, and it is regularized in a simple way to ensure a positive definite Hessian.

● The low-rank structure of the L-BFGS Hessian is exploited when solving for the quasi-Newton step on the TR boundary.

The QUOTR algorithm is described in Algorithm 1, and its user-controllable parameters are listed in Table 1. The parameters listed have been divided into three groups: free user choice, convergence tweaking, and expert-only controls. The two convergence criteria for the solver can be chosen however the user wishes, within reason. The next three parameters could be adjusted in cases that convergence is not as fast as desired. Finally, the remaining parameters are not recommended to be adjusted. Below we elaborate on each key aspect of the solver.

2.2.1 Parameterization. It is important to consider how the standard unconstrained quasi-Newton minimization scheme can be mapped to the constrained minimization of the single-

---

Algorithm 1 QUOTR SCF solver

1: **function** QUOTR $(\mathbf{C}^{(0)})$	
2: $k \leftarrow 0$, $S_{\text{taken}} \leftarrow$ false, $R_{\text{hist}} \leftarrow$ false, $\mathbf{U}_{\text{epoch}} \leftarrow \mathbf{1}$, $\{E^{(k)}, \mathbf{F}_{\text{AO}}\} \leftarrow \text{Fock}(\mathbf{C}^{(k)})$	
3: $\mathbf{g}^{(k)} \leftarrow \text{Grad}(\mathbf{F}_{\text{AO}}, \mathbf{C}^{(k)}, \mathbf{U}_{\text{epoch}})$	▷eqn (14)
4: $\mathbf{B}_0 \leftarrow \text{Preconditioner}(\mathbf{F}_{\text{AO}}, \mathbf{C}^{(k)})$	▷eqn (24)
5: **while** $\text{RMS}(\mathbf{g}^{(k)}) > t_{cg}$ AND $(k = 0$ OR $\Delta E^{(k)} > t_{ce})$ **do**	
6: **if** S_{taken} **then**	
7: $\mathbf{s}^{(k-1)} \leftarrow \mathbf{B}_0^{-1/2}\tilde{\mathbf{s}}^{(k-1)}$	
8: **if** $\mathbf{y}^{(k-1)} \cdot \mathbf{s}^{(k-1)} > t_h \|\mathbf{y}^{(k-1)}\| \|\mathbf{s}^{(k-1)}\|$ **then**	
9: $\tilde{\mathbf{y}}^{(k-1)} \leftarrow \mathbf{B}_0^{-1/2}\mathbf{y}^{(k)}$	
10: $\tilde{\mathbf{S}} \leftarrow \text{Append}(\tilde{\mathbf{S}}, \tilde{\mathbf{s}}^{(k-1)})$, $\tilde{\mathbf{S}} \leftarrow \text{Trim}(\tilde{\mathbf{S}}, m)$, $\tilde{\mathbf{Y}} \leftarrow \text{Append}(\tilde{\mathbf{Y}}, \tilde{\mathbf{y}}^{(k-1)})$, $\tilde{\mathbf{Y}} \leftarrow \text{Trim}(\tilde{\mathbf{Y}}, m)$ $\tilde{\mathbf{V}} \leftarrow \text{Concat}(\tilde{\mathbf{S}}, \tilde{\mathbf{Y}})$	▷update history
11: **end if**	
12: **end if**	
13: $R_{\text{hist}} \leftarrow \Delta^{(k)} < t_t$ OR $\|\mathbf{g}^{(k)}\|_\infty > t_b$	
14: **if** NOT R_{hist} **then**	
15: $\tilde{\mathbf{g}}^{(k)} \leftarrow \mathbf{B}_0^{-1/2}\mathbf{g}^{(k)}$	
16: **if** $k > 0$ AND $\tilde{\mathbf{V}}$ not empty **then**	▷BFGS
17: $\tilde{\mathbf{s}}^{(k)} \leftarrow \text{L-BFGS}(\tilde{\mathbf{g}}^{(k)}, \tilde{\mathbf{V}})$	▷eqn (27) and (29)

```
18:              if ‖š^(k)‖ > Δ^(k) then
19:                  š^(k) ← TRSTEP(Δ^(k), g̃^(k), š^(k), Ṽ)              ▷Algorithm 2
20:              end if
21:              q^(k) ← eqn (38), R_hist ← q^(k) > 0                     ▷energy increase predicted
22:          else
23:              š^(k) ← −g̃^(k)                                          ▷steepest descent
24:          end if
25:      end if
26:      if R_hist then                                                  ▷new epoch
27:          if not S_taken then
28:              {E^(k), F_AO} ← FOCK(C^(k))
29:          end if
30:          Ṽ ← {}, U_epoch ← 1, B_0 ← PRECONDITIONER(F_AO, C^(k)), g^(k) ← GRAD(F_AO, C^(k), U_epoch)   ▷steepest descent
31:          g̃^(k) ← B_0^{-1/2} g^(k), š^(k) ← −g̃^(k)
32:      end if
33:      if k = 0 OR Ṽ is empty then                                     ▷line search
34:          α^(k) ← LINESEARCH(š^(k))                                    ▷Section 2.2.4
35:          š^(k) ← α^(k) š^(k)/‖š^(k)‖, Δ^(k) ← α^(k)
36:      end if
37:      S_taken ← true, s^(k) ← B_0^{-1/2} š^(k)                         ▷attempt step
38:      U^(k) ← UNITARYSTEP (s^(k), U_epoch)                             ▷eqn (16) and (12)
39:      C^(k+1) ← C^(k) U^(k), {E^(k+1), F_AO} ← FOCK(C^(k+1)), ΔE^(k) ← E^(k+1) − E^(k)
40:      if k > 0 AND Ṽ not empty then
41:          ρ^(k) ← ΔE^(k)/q^(k)
42:          S_taken ← ρ^(k) ≥ τ_1 OR ΔE^(k) ≤ t_0
43:          if ρ^(k) < τ_2 then
44:              Δ^(k+1) ← MIN(η_1 Δ^(k), η_2 ‖š^(k)‖)
45:          else if (ρ^(k) > τ_3 AND ‖š^(k)‖ > η_3 Δ^(k)) then
46:              Δ^(k+1) ← η_4 Δ^(k)
47:          end if
48:      else
49:          S_taken ← ΔE^(k) ≤ t_0
50:          if S_taken then
51:              Δ^(k+1) ← ‖š^(k)‖
52:          end if
53:      end if
54:      if S_taken then
55:          k ← k + 1, U_epoch ← U_epoch U^(k−1), g^(k) ← GRAD(F_AO, C^(k), U_epoch), y^(k−1) ← g^(k) − g^(k−1)   ▷accept step
56:      else
57:          Δ^(k) ← Δ^(k+1)                                             ▷reject step, change TR
58:      end if
59:  end while
60:  return C^(k)
61: end function
```

determinant energy where the orbitals are required to be orthonormal. In the following, we assume the linear combination of atomic orbitals (LCAO) representation of the molecular orbitals (MOs), and thus the MOs are defined by the coefficient matrix $\mathbf{C}$.

We seek a unitary matrix, $\bar{\mathbf{U}}$, that transforms the initial (guess) set of orthonormal orbitals, $\mathbf{C}^{(0)}$, into the target solution, $\bar{\mathbf{C}}$.

$$\bar{\mathbf{C}} = \mathbf{C}^{(0)}\bar{\mathbf{U}} \tag{9}$$

The target unitary matrix is built as a sequence of unitary rotations,

$$\bar{\mathbf{U}} = \mathbf{U}^{(0)}\mathbf{U}^{(1)}\cdots \equiv \prod_k \mathbf{U}^{(k)}, \tag{10}$$

with each unitary rotation $\mathbf{U}^{(k)}$ obtained by solving a local subproblem. The orbitals at iteration $k > 0$ are given by the coefficient matrix obtained from the total rotation determined thus far.

$$\mathbf{C}^{(k)} = \mathbf{C}^{(0)}\prod_{i=0}^{k-1}\mathbf{U}^{(i)} = \mathbf{C}^{(k-1)}\mathbf{U}^{(k-1)} \tag{11}$$

We use the standard[9,65–67] exponential parameterization of unitary $\mathbf{U}^{(k)}$:

$$\mathbf{U}^{(k)} = \exp(\boldsymbol{\sigma}^{(k)}), \tag{12}$$

where $\boldsymbol{\sigma}^{(k)}$ is an antihermitian matrix encoding the unitary rotation of the orbitals. Formulating the optimization problem in terms of the antihermitian coordinate matrix $\boldsymbol{\sigma}$ rather than in terms of the density matrix allows us to avoid the need for diagonalization (which restores the idempotency of the density matrix in extrapolation/interpolation methods like DIIS[13,40]). The matrix exponentials are evaluated accurately (to finite precision t_e) as a Taylor series expansion using a simple scaling-and-squaring approach[31,68] with a fixed order of 2. In this technique, the matrix to be exponentiated is first divided by $2^2 = 4$. Next, the Taylor expansion for the exponential function is carried out on this scaled matrix, truncating when the norm of

6560 | Phys. Chem. Chem. Phys., 2024, 26, 6557–6573

This journal is © the Owner Societies 2024

the next term in the series drops below t_e. The resulting matrix is raised to the fourth power (by squaring twice) to obtain the exponential of the original matrix. Although some approaches evaluate the exponential approximately and return to the target manifold by additional orthogonalization,[9,19] accurate evaluation is important to be able to maintain the fidelity of the relationship between parameters and the objective function values in the context of extrapolation methods like BFGS. Evaluation of the matrix exponential could be improved further, *e.g.*, by leveraging the block-sparse antihermitian structure of $\boldsymbol{\sigma}$.[34]

It is important to express both the gradients and parameter updates for each iteration in the same coordinate frame (basis). In the context of the DIIS methods this is usually done by working in the AO basis (*e.g.*, as noted by Pulay in ref. 13 "In order to be useful for extrapolation purposes, [the Fock] matrices (one in each iteration step) must be transformed to a common basis, *e.g.*, to the original AO basis set."). Here the working frame is defined by the initial orthogonal MO basis for each *epoch* (namely, the sequence of iterations whose history is used to construct the current approximation to the Hessian). The gradient of the energy evaluated with orbitals $\mathbf{C}^{(k)}$ with respect to their arbitrary rotation $\boldsymbol{\sigma}^{(k)}$ has the familiar form when expressed in terms of $\mathbf{C}^{(k)}$:

$$(\mathbf{g}^{(k)})_{ia} \equiv \frac{\partial E}{\partial \sigma_{ai}^{(k)}} = 2n_i(\mathbf{F}^{(k)})_{ia}, \tag{13}$$

where a and i refer to the unoccupied and occupied MOs in $\mathbf{C}^{(k)}$, respectively, and n_i is the occupancy of ith orbital (2 for spin-restricted closed-shell SCF,1 for spin-unrestricted SCF). However, the gradient "at" arbitrary MOs can be expressed in an arbitrary (*e.g.*, epoch) basis. For example, the gradient at current orbitals $\mathbf{C}^{(k)}$ in epoch basis can be obtained by transforming eqn (13) to the epoch basis:

$$\mathbf{g}_{\text{epoch}}^{(k)} \equiv 2n_i[\mathbf{F}_{\text{epoch}}^{(k)}, \mathbf{P}_{\text{epoch}}^{(k)}]. \tag{14}$$

Here $\mathbf{F}_{\text{epoch}}^{(k)}$ is the Fock matrix evaluated with current orbitals $\mathbf{C}^{(k)}$ but represented in the epoch MO basis and $\mathbf{P}_{\text{epoch}}^{(k)}$ is the projector onto the occupied MOs in $\mathbf{C}^{(k)}$ expressed in the epoch basis. In practice, the Fock matrix in the epoch basis is evaluated in AO basis using the AO density matrix evaluated from $\mathbf{C}^{(k)}$ and then transformed to the epoch basis. The projection operator onto the occupied space at iteration k in the epoch basis is obtained as

$$\mathbf{P}_{\text{epoch}}^{(k)} = \mathbf{U}_{\text{epoch}}^{(k)}\mathbf{P}(\mathbf{U}_{\text{epoch}}^{(k)})^T, \tag{15}$$

where $\mathbf{P}$ is the diagonal matrix, with ones and zeroes on the diagonal for the occupied/unoccupied MOs, respectively. Not only does this formulation of the gradient allow us to have a consistent basis for forming the L-BFGS Hessian, but it also avoids evaluating the gradient at particular MOs using a non-truncating Taylor series around the reference/epoch basis, as in other solvers.[33]

Note that $\boldsymbol{\sigma}^{(k)}$ is a matrix, but only some of its elements can be varied independently. It is also traditional in applied mathematics literature to arrange the parameters of multivariate

functions into vectors. Thus it is appropriate to comment on the detailed relationship of $\boldsymbol{\sigma}^{(k)}$ and the corresponding *step* vector $s^{(k)}$. Due to the antihermiticity of $\boldsymbol{\sigma}$, $\sigma_{pq} = -\sigma_{qp}$, hence for real orbitals (which we assume here without any loss of generality) only the lower elements are independent. Although for 1-body SCF methods considered here elements of the gradient matrix are nonzero only between occupied and unoccupied orbitals, hence one would think that only $(\boldsymbol{\sigma})_{ia}$ elements need to be independently varied, this is only true for $\boldsymbol{\sigma}$ expressed in the current MO basis (*i.e.*, the basis defining the density). To be able to work in an arbitrary (*e.g.*, epoch) basis all lower-triangular elements of $\boldsymbol{\sigma}$ thus must be considered independent. For an MO basis with n_o orbitals, this means that the number of independent parameters is $n \equiv n_o(n_o - 1)/2$, assuming no additional symmetries are taken into account. In the spin-unrestricted case, the sigma elements for the separate alpha and beta spin MOs are simply concatenated into one vector. Notice that the gradient matrix in eqn (14) is also antihermitian, with the same structure as $\boldsymbol{\sigma}$. Thus, we can map the matrix elements of the gradient to a vector in exactly the same way as for the matrix elements of $\boldsymbol{\sigma}$, using only the lower (or upper) triangle. We use the symbol $s^{(k)}$ for the vector version of $\boldsymbol{\sigma}^{(k)}$; for the gradient henceforth only its vector form, $\mathbf{g}^{(k)}$, is used.

The steps and gradient differences are the ingredients for the L-BFGS update to the Hessian and its inverse. All quantities throughout each epoch (see below) are kept in the epoch "reference" basis to make application of L-BFGS and TR consistent. However, when a new step $s^{(k)}$ is proposed, to convert it to the unitary rotation *via* eqn (12) it must be transformed to the current MO basis, *via*

$$\boldsymbol{\sigma}^{(k)} = \mathbf{U}_{\text{epoch}}^{(k-1)T}\boldsymbol{\sigma}_{\text{epoch}}^{(k)}\mathbf{U}_{\text{epoch}}^{(k-1)}. \tag{16}$$

Our convergence criteria require both the energy change between iterations ($\Delta E^{(k)} \equiv E^{(k+1)} - E^{(k)}$) and the root mean square (RMS) of the unique gradient elements to be small. Generally, we require energy change to be below $1 \times 10^{-9}E_h$ and the RMS gradient (in epoch basis) to be smaller than 1×10^{-5}. However, for some comparisons with other solvers, we use the 2-norm of the gradient instead of the RMS version.

2.2.2 Quasi-Newton method. The L-BFGS algorithm is used to approximate the Hessian, $\mathbf{B}_{\text{BFGS}}^{(k)} \approx \mathbf{B}^{(k)}$, and its inverse, $\mathbf{H}_{\text{BFGS}}^{(k)} \equiv (\mathbf{B}_{\text{BFGS}}^{(k)})^{-1} \approx (\mathbf{B}^{(k)})^{-1}$, using the history vectors from (at most) m previous iterations. In the following, we will assume that the history size is equal to m for simplicity. This approximate Hessian and its inverse can be represented in low-rank form as follows:[56]

$$\mathbf{B}_{\text{BFGS}}^{(k)} = \mathbf{B}_0 - \mathbf{V}_{\text{B}}^{(k)}(\mathbf{W}^{(k)})^{-1}\mathbf{V}_{\text{B}}^{(k)T} \tag{17}$$

$$\mathbf{H}_{\text{BFGS}}^{(k)} = \mathbf{H}_0 + \mathbf{V}_{\text{H}}^{(k)}\mathbf{M}^{(k)}\mathbf{V}_{\text{H}}^{(k)T} \tag{18}$$

Here the matrix $\mathbf{B}_0$ is an initial (often diagonal) approximation to the Hessian chosen at the beginning of the current epoch, which in principle could be any positive definite matrix.[52] The matrix $\mathbf{H}_0$ is the corresponding initial inverse Hessian approximation: $\mathbf{H}_0 \equiv \mathbf{B}_0^{-1}$. More will be said about these critical components later. At iteration k with the BFGS history containing m {step, gradient difference} pairs, matrix $\mathbf{V}_{\text{B}}^{(k)}$ has $2m$ columns,

This journal is © the Owner Societies 2024

Phys. Chem. Chem. Phys., 2024, **26**, 6557–6573 | **6561**

with the first m being history step vectors multiplied by the initial Hessian, $\mathbf{B}_0 \mathbf{s}^{(k)}$, followed by the m matching gradient differences, $\mathbf{y}^{(k)} \equiv \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}$. $\mathbf{V}_H^{(k)}$ is obtained from $\mathbf{V}_B^{(k)}$ as $\mathbf{V}_H^{(k)} \equiv \mathbf{H}_0 \mathbf{V}_B^{(k)}$. The square matrices $\mathbf{W}^{(k)}$ and $\mathbf{M}^{(k)}$ have dimension $2m$, and the need for the inverse of $\mathbf{W}^{(k)}$ is not a problem since m is typically small (we have used $m = 8$). The formation of $\mathbf{W}^{(k)}$ and $\mathbf{M}^{(k)}$ is described in the literature,[56,69] but essentially they are composed of various dot products involving the history vectors, (requiring an inverse of one of the $m \times m$ subblock).

$$\mathbf{W}^{(k)} = \begin{pmatrix} ((\mathbf{S}^{(k)})^T \mathbf{B}_0 \mathbf{S}^{(k)}) & | \mathbf{L}^{(k)} \\ (\mathbf{L}^{(k)})^T & | -\mathbf{E}^{(k)} \end{pmatrix} \qquad (19)$$

$$\mathbf{M}^{(k)} = \begin{pmatrix} ((\mathbf{L}^{(k)})^{-1}(\mathbf{E}^{(k)} + (\mathbf{Y}^{(k)})^T \mathbf{H}_0 \mathbf{Y}^{(k)})(\mathbf{L}^{(k)})^{-T}) & | -(\mathbf{L}^{(k)})^{-1} \\ -(\mathbf{L}^{(k)})^{-T} & | \mathbf{0} \end{pmatrix} \qquad (20)$$

In eqn (19) and (20), $\mathbf{S}^{(k)}$ is an $n \times m$ matrix containing the history column vectors, $\mathbf{s}^{(k)}$, and similarly $\mathbf{Y}^{(k)}$ contains the gradient differences, $\mathbf{y}^{(k)}$. The smaller $m \times m$ submatrices $\mathbf{L}^{(k)}$ and $\mathbf{E}^{(k)}$ are simply constructed as below.

$$\mathbf{L}_{ij}^{(k)} = \begin{cases} \mathbf{s}^{(k-m-1+i)} \cdot \mathbf{y}^{(k-m-1+j)}, & \text{if } i > j \\ 0 & \text{otherwise} \end{cases} \qquad (21)$$

$$\mathbf{E}_{ij}^{(k)} = \begin{cases} \mathbf{s}^{(k-m-1+i)} \cdot \mathbf{y}^{(k-m-1+j)}, & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \qquad (22)$$

One of the advantages of the L-BFGS Hessian approximation, apart from not requiring calculation of second derivatives, is that it can be stored in this factorized form by simply keeping the relatively small matrices $\mathbf{B}_0$, $\mathbf{V}_B^{(k)}$, and $\mathbf{W}^{(k)}$. Considering that $\mathbf{B}_0$ is a diagonal matrix, we only need to store $(2m + 1) n + 4m^2$ elements, which is typically much smaller than the full Hessian which requires n^2 elements. From the development up to this point, it would seem that we also need to store the information for the inverse L-BFGS Hessian, specifically $\mathbf{V}_H^{(k)}$, but this will be dealt with soon.

The quasi-Newton step, $\mathbf{s}^{(k)} = -\mathbf{H}_{BFGS}^{(k)} \mathbf{g}^{(k)}$, is calculated by multiplying the inverse L-BFGS Hessian of eqn (18) with the negative of the gradient. Thus, the factorized form of eqn (18) makes the task of calculating the quasi-Newton step simply a matter of a few matrix-vector multiplications. Note that although we only need the inverse Hessian to compute the quasi-Newton step, the Hessian is used to compute the energy decrease predicted by the quadratic model. This is needed for determining how the TR is to be updated between iterations as described in Section 2.2.3.

As is well known, due to the large (and increasing with the basis) condition number of the Hessian it is important to use a preconditioner to achieve competitive convergence.[18,19,21] Since the orbital Hessian is often diagonally dominant and its 1-electron (Fock) contributions are cheap to evaluate, we define $\mathbf{B}_0$ in terms of the diagonal elements of the 1-electron

component of the exact Hessian, $\mathbf{B}_{1e}$, whose unique nonzero elements in the current MO basis are

$$(\mathbf{B}_{1e})_{(ia)(jb)} = 2n_i(F_{ab}\delta_{ij} - F_{ij}\delta_{ab}). \qquad (23)$$

Since we compute the preconditioner once per epoch (*i.e.*, infrequently), for each epoch we choose the basis to make the Hessian as diagonally dominant as possible by choosing the "pseudocanonical" basis,[5] which makes the occupied–occupied and unoccupied–unoccupied blocks of the Fock matrix diagonal (the off-diagonal blocks are non-zero until convergence). In such choice of the epoch basis $\mathbf{B}_0$ is defined as the regularized nonzero unique elements on the diagonal of $\mathbf{B}_{1e}$:

$$(\mathbf{B}_0)_{(ia)(ia)} = 2n_i \, r(F_{aa} - F_{ii}); \qquad (24)$$

the rest of the unique diagonal elements are set to 1 to ensure the finite condition number of the Hessian and existence of the inverse Hessian in an arbitrary basis. Regularizer $r(x)$ in eqn (24) is defined as

$$r(x) = \begin{cases} x, & \text{if } x > t_r, \\ t_r & \text{otherwise.} \end{cases} \qquad (25)$$

with the regularizer threshold t_r defining the minimum acceptable HOMO–LUMO gap at the beginning of the epoch. Unlike some other quasi-Newton solvers,[21] we do not update the diagonal part of the approximate Hessian every iteration. In principle, this could lead to slower convergence, since the approximation becomes less accurate as the orbitals are changed from the point where the diagonal Hessian was calculated.[21] Indeed, we found that in the early iterations, it is imperative to use an updated preconditioner, and thus we do an approximate line search along the preconditioned steepest descent direction until the max element of the gradient drops below a threshold (we generally use 0.1 which is smaller than 0.25, which has literature precedent[19]). During this early phase of the solver, the orbitals are made "pseudocanonical" in each iteration, and the preconditioner is rebuilt. Essentially, the epochs are only 1 iteration long. However, near the solution, we have found that it is not necessary to update the preconditioner every iteration, and because we work in the epoch MO basis it would be difficult to update the preconditioner. We have found that with a good initial guess, only a median of 3 iterations of this line search are required to drop the max gradient element below 0.1 and trigger L-BFGS starting for simple systems (see Section 4.1). If the gradient gets large again, the history is reset, and preconditioned steepest descent is again carried out with an updated preconditioner. Every time the history is reset the epoch basis is also reset to the current orbitals.

An alternative and perhaps more conventional view of the preconditioner is that it is a basis transformation that makes the diagonal part of the L-BFGS Hessian or its inverse closer to an identity matrix. To see how this view relates to the diagonal Hessian, consider the following transformation of the quasi-Newton equation: $\mathbf{s} = -\mathbf{H}\mathbf{g}$ (omitting iteration index). Multiply both sides of the equation by $\mathbf{B}_0^{1/2}$ and insert

6562 | *Phys. Chem. Chem. Phys.*, 2024, **26**, 6557–6573

This journal is © the Owner Societies 2024

$1 = \mathbf{B}_0^{1/2}\mathbf{B}_0^{-1/2}$ (which is clearly acceptable because the diagonal matrix $\mathbf{B}_0$ is guaranteed to be positive definite due to the regularizer) between the inverse Hessian and the gradient, to get an equivalent equation.

$$\mathbf{B}_0^{1/2}\mathbf{s} = -(\mathbf{B}_0^{1/2}\mathbf{H}\mathbf{B}_0^{1/2})\mathbf{B}_0^{-1/2}\mathbf{g}. \tag{26}$$

With $\tilde{\mathbf{s}} \equiv \mathbf{B}_0^{1/2}\mathbf{s}$, $\tilde{\mathbf{g}} \equiv \mathbf{B}_0^{-1/2}\mathbf{g}$, and $\tilde{\mathbf{H}} \equiv \mathbf{B}_0^{1/2}\mathbf{H}\mathbf{B}_0^{1/2}$ identified as the step, gradient, and inverse Hessian, respectively, in the "preconditioned" basis (henceforth denoted by the tilde), the Newton step (eqn (26)) becomes

$$\tilde{\mathbf{s}} = -\tilde{\mathbf{H}}\tilde{\mathbf{g}}. \tag{27}$$

The L-BFGS Hessian and inverse Hessian in the preconditioned basis simplify to

$$\tilde{\mathbf{B}}_{\text{BFGS}} \equiv \mathbf{B}_0^{-1/2}\mathbf{B}_{\text{BFGS}}\mathbf{B}_0^{-1/2} = \mathbf{1} - \tilde{\mathbf{V}}(\mathbf{W})^{-1}\tilde{\mathbf{V}}^T, \tag{28}$$

$$\tilde{\mathbf{H}}_{\text{BFGS}} \equiv \mathbf{B}_0^{1/2}\mathbf{H}_{\text{BFGS}}\mathbf{B}_0^{1/2} = \mathbf{1} + \tilde{\mathbf{V}}\mathbf{M}\tilde{\mathbf{V}}^T, \tag{29}$$

where

$$\tilde{\mathbf{V}} \equiv \mathbf{B}_0^{1/2}\mathbf{V}_H = \mathbf{B}_0^{-1/2}\mathbf{V}_B. \tag{30}$$

The $\tilde{\mathbf{V}}^{(k)}$ matrix is computed straightforwardly from the history vectors in the preconditioned basis. Note that some steps of the algorithm require quantities in the original basis, such as the sanity checks and computing the orbital rotation matrices *via* eqn (12), thus it is not possible to work exclusively in the preconditioned basis. Transforming back to the original basis is straightforward, *e.g.*, $\mathbf{s}^{(k)} = \mathbf{B}_0^{-1/2}\tilde{\mathbf{s}}^{(k)}$.

Here is probably a good place to summarize the steps to obtain the unitary rotation at iteration k, since there are now quite a few layers.

$$\tilde{\mathbf{s}}^{(k)} \rightarrow \mathbf{s}^{(k)} \rightarrow \sigma_{\text{epoch}}^{(k)} \rightarrow \sigma^{(k)} \rightarrow \mathbf{U}^{(k)} \tag{31}$$

To keep the L-BFGS Hessian positive definite between iterations, we require that[57]

$$\mathbf{s}^{(k)} \cdot \mathbf{y}^{(k)} > t_h \|\mathbf{s}^{(k)}\| \|\mathbf{y}^{(k)}\|. \tag{32}$$

When this requirement is not met vectors $\{\tilde{\mathbf{s}}^{(k)}, \tilde{\mathbf{y}}^{(k)}\}$ are not added to the history. Here we used $t_h = 10^{-5}$.

2.2.3 Trust-region step restriction. Since the quasi-Newton methods use a quadratic approximation to the objective function, when optimizing a nonlinear function every proposed quasi-Newton step must be tested for sanity to ensure that the quadratic model is a faithful approximation. First, we expect each step to lower the energy, hence each proposed step should point downhill. Second, steps in downhill directions should not be too large, due to the increasing likelihood that the quadratic model becomes poor. QUOTR uses the trust-region method for step restriction. In the TR method the maximum step size is limited by the trust-radius that is dynamically updated by comparing the quadratic model predictions with the actual objective function values encountered during the optimization. Namely step $\tilde{\mathbf{s}}^{(k)}$ is TR-acceptable if

$$\|\tilde{\mathbf{s}}^{(k)}\| \leq \Delta^{(k)}, \tag{33}$$

Table 2 User-controllable parameters of the TR solver

Description	Symbol	Value
Initial guess σ	σ_{init}	0.0
Convergence criterion 1	T_1	10^{-4}
Convergence criterion 2	T_2	10^{-7}
Maximum iterations	i_{max}	500
Sigma modify factor	F_0	1.1

where trust-radius $\Delta^{(k)}$ is updated using Fletcher's algorithm.[64] By comparing the quadratic model prediction for the energy with the actual energy value every iteration the trust-radius can be expanded, contracted, or left unchanged (see Algorithm 1). Fletcher's algorithm parameters (τ_i/η_i) were borrowed from a recent study.[56] The initial value of the trust-radius is set to the most recent successful line search step size since this should be of the correct order of magnitude for the next step. Also, we always perform line search when there are no history data available so the most recent step size from line search is always a known quantity when quasi-Newton steps are attempted.

When the quasi-Newton step does not satisfy eqn (33), we solve for the optimal step that is within the TR, which must be on the TR boundary: $\|\tilde{\mathbf{s}}^{(k)}\| = \Delta^{(k)}$. This is done by finding an optimal level-shift, σ, which satisfies the two conditions:[56]

$$(\tilde{\mathbf{B}}_{\text{BFGS}} + \sigma\mathbf{1})\tilde{\mathbf{s}}^{(k)} = -\tilde{\mathbf{g}}^{(k)} \tag{34a}$$

$$\|\tilde{\mathbf{s}}^{(k)}\| = \Delta^{(k)}. \tag{34b}$$

Level-shift σ of the L-BFGS Hessian is updated iteratively until eqn (34) is satisfied to the desired precision controlled by parameters $T_{1,2}$ (see Table 2 and Algorithm 2).

The TR solver in QUOTR is based on the solver described by Burdakov *et al.*[56] that leverages the low-rank structure of the L-BFGS Hessian. The advantage of this approach is that the cost of each TR solver iteration is trivial compared to the conventional TR formulation with exact Hessian in which each iteration has a cost similar to that of the gradient evaluation. The TR solver algorithm is outlined in Algorithm 2 and its user-controllable parameters are given in Table 2. Although the parameters for the TR solver could be adjusted, we recommend keeping them as listed. The initial guess for σ is best kept at zero, as this will help convergence in cases where the optimal level shift is a very small value. Changing parameters $T_{1,2}$ only impacts when the TR problem is considered solved, and tightening the values given is not expected to have a significant impact on QUOTR overall.

Algorithm 2. Trust-region step update

1: **function** TRSTEP $(\Delta, \tilde{\mathbf{g}}, \tilde{\mathbf{s}}, \tilde{\mathbf{V}})$
2:      $\sigma \leftarrow \sigma_{\text{init}}, C \leftarrow \text{false}, F \leftarrow F_0$ ▷initialize
3:      $\mathbf{G} \leftarrow \tilde{\mathbf{V}}^T\tilde{\mathbf{V}}$
4:      $\{\mathbf{R}, \mathbf{R}^{-1}\} \leftarrow \text{ORTHOGONALIZE}(\mathbf{G})$ ▷eqn (37)
5:      $\mathbf{R}^{-1}\mathbf{W}^{-1}(\mathbf{R}^{-1})^T = \mathbf{U}\Lambda\mathbf{U}^T$ ▷diagonalize
6:      $\mathbf{P}_\parallel = \tilde{\mathbf{V}}\mathbf{R}\mathbf{U}$ ▷eqn (12)[57]
7:      $\tilde{\mathbf{g}}_\parallel = (\mathbf{P}_\parallel)^T\tilde{\mathbf{g}}$
8:      $r_{\text{max}} = \dim(\tilde{\mathbf{g}}_\parallel)$

This journal is © the Owner Societies 2024

Phys. Chem. Chem. Phys., 2024, **26**, 6557–6573 | **6563**

procedure has recently been used,[33] and a nearly identical method for the step determination also has precedent.[49]

Choosing α_{fit} is crucial for the success of the line search. Due to the fact that the exponential parametrization of the unitary is periodic, it is straightforward to estimate the shortest period of oscillation by finding the largest (magnitude) eigenvalue ω_{max} of the σ matrix (or matrices, for the unrestricted case) corresponding to the SD direction $-\tilde{\mathbf{g}}^{(k)}/\|\tilde{\mathbf{g}}^{(k)}\|$.[71] This results in

$$\alpha_{\text{fit}} = \frac{2\pi}{q|\omega_{\text{max}}|} \qquad (41)$$

Here q is set to 4 due to the quartic dependence of the Hartree–Fock energy on the orbitals without the orthonormality constraint.

The minimum of $E_{\text{fit}}(\alpha)$ is found by solving the quadratic equation $dE_{\text{fit}}/d\alpha = 0$. Each of its two solutions, α_{min}, is checked for sanity in turn; it is expected to be real, positive, resulting in a decrease of energy (*i.e.*, $E_{\text{fit}}(\alpha_{\text{min}}) < E_{\text{fit}}(0)$), and the second derivative of E_{fit} at α_{min} should be positive. Since a 3rd-order polynomial can have at most one local minimum, the solution with the smallest positive α is checked whether it satisfies all of these criteria. The failure to meet these conditions indicates a poor quality of the fit, and in such case the polynomial fit is recomputed with α_{fit} scaled by $\alpha_{\text{fit,shrink}} = 1/2$. Note that the energy decrease criterion is checked only after building the new Fock matrix, while the other conditions can be checked immediately after solving the quadratic equation.

2.2.5 Orbital guess. Starting (guess) orbitals are another critical component for rapid SCF convergence. We generally use an extended Hückel initial guess,[72] which is constructed in a minimal basis and then projected onto the full orbital basis. The standard Wolfsberg–Helmholtz formula for the off-diagonal elements of the extended Hückel Hamiltonian is used:[72] $H_{ij} = K'S_{ij}(H_{ii} + H_{jj})/2$, but with the updated formula for the value of K'.[73] Instead of experimental ionization potentials for the diagonal elements, we follow a suggestion by Lehtola[74] (earlier by Norman[75]) and use numerical Hartree–Fock orbital energies[76] for each shell. Although the guess orbitals are populated according to the Aufbau principle using the extended Hückel energies and in the minimal basis, after projection to the orbital basis the populations may not be qualitatively correct. When this situation occurs, and the symmetry of the orbitals is such that there is no gradient between incorrectly occupied and incorrectly unoccupied orbitals, QUOTR will not be able to correct the populations. Therefore, we have added an option to perturb the guess orbitals to allow the solver to rotate the incorrectly occupied orbitals and find the lower energy solution. The orbitals are perturbed by $\exp(\boldsymbol{\sigma})$ with unique elements of $\boldsymbol{\sigma}$ filled with uniformly-distributed random numbers in $[-0.05, 0.05]$. The "strength" of this random perturbation can be changed, which simply scales all elements of $\boldsymbol{\sigma}$ such that the maximum absolute value is something other than 0.05 (default). Additionally, we can choose to either perturb "all" or just the "valence" orbitals. Pseudorandom number generator is used with user-controlled seed to ensure deterministic perturbation.

2.2.6 Additional heuristics. Unfortunately, the sole use of preconditioned L-BFGS with TR step restriction is not sufficient when dealing with problems with complex optimization landscapes that arise for open-shell and, especially, metal-containing systems. This occurs due to poor quality of the quadratic model when far from convergence. Thus, as is typical with second-order solvers[19,21] QUOTR uses SD steps until the $\|\mathbf{g}\|_{\infty}$ drops below the L-BFGS start threshold, t_{b}.

Here's a brief recap of all situations that cause history reset:
- The gradient is too large ($\|\mathbf{g}^{(k)}\|_{\infty} > t_{\text{b}}$)
- The TR is too small ($\varDelta^{(k)} < t_{\text{t}}$)
- The quadratic model predicts energy increase ($q^{(k)} > 0$)

The first 2 of these situations are detected before the L-BFGS step is constructed, allowing the solver to skip this step to go directly to re-building the preconditioner and on to perform the line search. The last situation is only determined after the L-BFGS step is calculated.

3 Technical details

The QUOTR solver was implemented in a developmental version of the Massively Parallel Quantum Chemistry (MPQC) version 4 program package.[77] The default values for parameters in Tables 1 and 2 were used throughout, unless noted. The orbital bases sets used were 6-31G*,[78–84] 6-31G**,[80] 6-311++G**,[85–88] def2-TZVPP,[89] cc-pVTZ-DK,[90] and cc-pVTZ-X2C.[91] Density fitting, where noted, used the def2-universal-J basis.[92] The extended Hückel initial guess was constructed in the Huzinaga MINI basis,[93] then projected onto the orbital basis. Calculations on the f-element containing system in Section 4.2.3 did not use the extended Hückel guess to avoid the uncertainties about its quality in such heavy systems. Instead, we use a superposition of minimal atomic basis guess densities to construct the initial Fock matrix in the orbital basis (without projection), followed by diagonalization. The minimal AO basis used the corresponding subset of the ANO-DK3 basis[94] on Fm atom and the MINI AO basis on the other atoms. The same minimal bases were used to compute the atomic charges in Section 4.2.3, using the pseudoinverse method described in ref. 95. The orbital bases used in the relativistic calculations employed cc-pVTZ-X2C on the Fm atom, and cc-pVTZ-DK on all other atoms.

Hartree–Fock calculations were performed in Section 4.1 for the G2 set,[96] the geometries for which were obtained from the Gaussian output files on the NIST website[97] with the exception of four systems that were not available with the correct method (MP2 = FULL/6-31G*). For the four systems that were not available from NIST (acetamide, furan, SiH_2-triplet and 2-butyne) Gaussian 09[98] was used to obtain the geometry. The G2-1 set consists of 55 systems and is a subset of G2-2, which consists of a total of 148 systems.[96]

Henceforth RH/DIIS will be denoted simply by DIIS. Unless explicitly mentioned, DIIS results were obtained with its implementation in MPQC using the default parameters: keeping the 5 most recent pairs of Fock matrix and error vectors for the extrapolation, and no damping applied.

This journal is © the Owner Societies 2024

Phys. Chem. Chem. Phys., 2024, **26**, 6557–6573 | **6565**

The KS DFT implementation in MPQC uses GauXC[99] (which uses LibXC[100]) for calculation of the exchange–correlation potentials and energies. The integration grid used for the 1PLW calculations in Section 4.2.1 was the "ultrafine" grid (99 radial Mura–Knowles[101] points, 590 angular Lebedev–Laikov[102] points). All other KS DFT calculations used the "superfine" grid which has 250 radial points and 974 angular points for all atoms except hydrogen, which has 175 radial points. The particular parameterization we use for LDA is Slater Exchange[103] with VWN RPA.[104] For the B3LYP calculations on the Cr systems in Section 4.2.2, we use VWN3 for the local correlation functional[104] to match PySCF. The structure of the neuropeptide, 1PLW,[105] was obtained from the Protein Data Bank (PDB).[106]

Calculations using KDIIS[53] for SCF acceleration on the CrC and Cr_2 systems in Section 4.2.2 were performed with the Orca program system, version 5.0.4.[107] Additionally, the DIIS implementation from Orca was also used for these systems instead of the MPQC version. The bond length used for both of these diatomic systems is 2 angstrom, as has been used in previous studies.[20,31] Orbitals were plotted for Cr_2 with Jmol; due to its inability to read in Molden files with $l = 4$ (g) AOs, the calculations (only for the visualizations) were performed with def2-TZVPP with g-type AOs removed.

Full (2-component) and spin-free (1-component) 1-electron X2C Hamiltonians were implemented in MPQC using the standard formalism.[108,109] No empirical scaling was utilized to emulate the mean-field effects on the Dirac Hamiltonian. For the sake of comparison with the results of ref. 110 only the spin-free X2C Hamiltonian was used here.

4 Results and discussion

4.1 Easy testset: G2 data set

Performance of QUOTR was first assessed for converging Hartree–Fock wave functions (RHF and UHF for closed- and open-shell systems, respectively) and compared to DIIS, as well as published literature data for three second-order solvers, GDM,[21] ETDM,[33] and CIAH.[30] The computations where convergence was not achieved in 256 iterations (333 for ETDM, 50 macroiterations for CIAH) were removed from the statistical

values and aggregated in the "no convergence" row. The number of "local minima" for each solver was determined by comparing converged energies to the lowest energy that we obtained, except for the GDM and ETDM results (which are the numbers reported by the respective publications). All calculations with QUOTR and DIIS used the extended Hückel guess with perturbed valence orbitals. Although we attempted to compare solvers as faithfully as possible, due to lack of direct access to the source code and/or implementation of GDM and ETDM this was not always possible (see below).

Table 3 reports the number of Fock matrix evaluations N_F ("Fock builds") and the number of solver iterations N_I. Due to the different performance statistics reported in the literature for GDM and ETDM, three different sets of G2 calculations were performed.

4.1.1 G2-1/6-311++G**. Similar to QUOTR, the geometric direct minimization (GDM) solver is a BFGS-based solver introduced by Head-Gordon and Van Voorhis in 2002.[21] A key difference between GDM and QUOTR is the use of the TR by QUOTR. Additionally, GDM updates the preconditioner every iteration (rather than once per epoch in QUOTR), with regularization applied by adding a diagonal shift to the Hessian equal to the energy change in the most recent iteration. Therefore, comparison to GDM is appropriate as a way to evaluate the effectiveness of the TR and the appropriateness of the preconditioner. Due to the lack of access to the commercial implementation of GDM, we restricted our comparison to the data published in ref. 21.

To make the comparison with GDM as faithful as possible, we used the same orbital basis set and convergence criteria ($1 \times 10^{-10} E_h$ for the energy difference between iterations and 1×10^{-7} for the RMS of the unique gradient elements). Our initial guess orbitals were likely similar; however, we did apply a random unitary perturbation to the valence orbitals, which was not done by GDM. The average number of iterations taken by QUOTR is about 4 more than GDM, and it has a higher max at 107 for NO followed by P_2 at 66 iterations. Notice, though, that GDM found 5 local minima relative to the lowest energy that they could obtain in any of their calculations. Thus, while the convergence with QUOTR takes more iterations, QUOTR appears to be more robust than GDM. Unfortunately, the number of Fock builds used by GDM was not reported in ref. 21.

Table 3 Performance comparison of QUOTR to other SCF solvers for standard G2 set

	G2-1/6-311++G**			G2-2/6-31G**			G2-2/6-31G*		
	DIIS	QUOTR	GDM[a]	DIIS	QUOTR	ETDM[b]	DIIS	QUOTR	CIAH[c]
N_F: mean	15.2	26.8	—	13.6	20.5	(17)	15.3	19.4	34.5
N_F: median	14	20	—	12	17	(17)	12	16	30
N_F: max	40	136	—	64	107	72	234	69	77
N_I: mean	15.2	20.5	16.3	13.6	15.3	—	15.3	14.2	2.9
N_I: median	14	15	—	12	12	—	12	12	3
N_I: max	40	107	42	64	96	—	234	55	5
Local minima	3	0	5	5	0	—	8	0	4
No convergence	2	0	0	2	0	0	1	0	1

[a] Geometric direct minimization.[21] [b] Exponential transformation direct minimization.[33] [c] Co-iterative augmented Hessian.[30]

6566 | *Phys. Chem. Chem. Phys.*, 2024, **26**, 6557–6573

This journal is © the Owner Societies 2024

The computational cost of QUOTR is fairly competitive with DIIS, with a median number of Fock builds being 20 and 14, respectively; note that the latter is close to the reported performance of DIIS in the GDM paper.[21] The only systems that did not converge for DIIS in 256 iterations were HCO and Si_2. There were no local minima found by QUOTR (relative to the DIIS solution) but for 3 systems (CN, O_2 and CH singlet) QUOTR got a significantly lower energy than DIIS. Also note that for 5 systems GDM landed on local minima too.[21] Thus for the G2-1/6-311++G** test set QUOTR was found to be more robust than DIIS and GDM, albeit with a slightly worse performance.

4.1.2 G2-2/6-31G.** Another similar, and more recent, direct minimization SCF solver is the exponential transformation direct minimization (ETDM) solver.[33] Again, the lack of TR is one of the main differences compared to QUOTR, but ETDM also approximates the gradient as it does not work in the epoch formulation. The fact that ETDM does not use TR may be compensated by the stronger criteria used in the line search. Due to the lack of access to the implementation of ETDM we restricted our comparison to the data published in ref. 33.

Comparison to ETDM was less precise for a few reasons. The data presented in Table 3 for ETDM used KS DFT (PBE) and a different basis (double-zeta polarized numerical atomic orbital basis equipped with projector augmented wave (PAW) for the inner region). Here we performed all-electron SCF in the 6-31G** Gaussian AO basis, because it is a double-zeta basis with polarization functions on all atoms, so should be similar to the basis used by ETDM. Also, the open-shell KS DFT implementation in MPQC is not yet finalized, hence we are comparing QUOTR HF SCF to ETDM PBE SCF. Lastly, QUOTR set $m = 3$ to make the comparison with ETDM as faithful as possible. The average number of iterations (unclear if it is mean or median) reported for ETDM was 17, which compares well with the median of QUOTR at 17. Therefore, we conclude that QUOTR is roughly equivalent in computational cost to ETDM. Notice also that 5 of the DIIS solutions are local minima relative to QUOTR.

4.1.3 G2-2/6-31G*. The last batch of comparisons pits QUOTR and DIIS against CIAH, a second-order (augmented Hessian) solver that uses exact Hessian.[30] The augmented Hessian (AH) approach can be viewed as a variant of the Newton method with step restriction induced by a spectral shift of the Hessian tuned at each step to ensure that the predicted step results in energy decrease (this idea is sufficiently general to be applicable in combination with RH/DIIS[28]). The spectral shift can be viewed as an optimal regularizer for the Hessian; since it vanishes automatically in the vicinity of the minimum the augmented Hessian approach approaches the quadratic convergence rate of the unmodified Newton method. Thus the augmented Hessian methods are potentially superior to RH/DIIS or quasi-Newton methods (like QUOTR) that have slower convergence rates. Indeed, for the SCF problem the AH methods are known[30,31] to converge in substantially fewer iterations than the RH/DIIS heuristics, and are more robust. Thus the AH-based SCF methods can viewed as the benchmark to beat for QUOTR.

With access to the implementation of CIAH in PySCF[111] we were able to perform a direct comparison with QUOTR (Table 3). Note that the number of iterations is sometimes used[35] to compare the cost of the AH-based methods to that of RH/DIIS and quasi-Newton approaches; such comparison is misleading. Each iteration of the AH approach, in addition to the evaluation of the gradient, involves iteratively solving an eigenproblem defining the optimal shift; thus the cost of each iteration is determined by the cost of multiplying a trial step vector by the (exact) Hessian times the number of iterations of the eigensolver. The cost of applying exact Hessian to the trial step is comparable to the cost of the Fock matrix evaluation (in fact many programs will use the same machinery for both). Thus the performance assessment of CIAH and other AH-based methods will report the number of *Fock build equivalents*, N_F. For CIAH the total number of Fock build equivalents is the sum of the key frames (KF) and coulomb/exchange (JK) calls, with the former accounting for the cost of the *exact* evaluation of the gradient and the latter the cost of Hessian-step products.[30]

The convergence statistics in Table 3 indicate that the median N_F for QUOTR is roughly half of the median N_F for CIAH. QUOTR was also more robust: for one system (HCl) CIAH did not converge as it could not reduce the gradient norm below 1.2×10^{-6} and failed to make progress until the maximum number of iterations was reached (50). In four other systems (CH, O_2, NO_2 and Si_2) CIAH landed on a local minimum, as indicated by the substantially lower energies obtained with QUOTR. For the rest of the systems, CIAH and QUOTR agreed within $1 \times 10^{-9} E_h$. The systems that took the most Fock build equivalents to converge with QUOTR and CIAH were Si_2 and NF_3, respectively, requiring 69 and 77.

As expected, RH/DIIS is on average slightly faster than QUOTR, but is far less robust, with 8 local minima and 1 system where converged solution could not be obtained. This demonstrated proliferation of incorrect solutions found with DIIS for even such "easy" chemical systems as those in the G2 test set has significant implications. How could high-throughput screening be performed with confidence using such a solver? We expect that other programs that default to using the RH solver will also have similar issues. And, as demonstrated in ref. 21, second-order solvers are not a panacea.

Even with direct access to the CIAH implementation it was difficult to compare methods fairly. Some differences were minor, such as the convergence tests. Both solvers use magnitudes of energy change and gradient for convergence monitoring. For the former threshold was set to $1 \times 10^{-9} E_h$, but the CIAH gradient criterion (1×10^{-6}) is defined in terms of the gradient norm rather than the RMS value used by QUOTR (1×10^{-5}). Some differences were more significant, like the choice of the guess orbitals. The QUOTR data in Table 3 was obtained with its default guess (perturbed extended Hückel) that differs from the default minimal AO guess used for CIAH calculations. To elucidate the impact of the guess differences we performed additional tests with (unperturbed) core Hamiltonian guess implemented identically in MPQC and PySCF to ensure that the initial energies matched to better than 9 digits between the

This journal is © the Owner Societies 2024

Phys. Chem. Chem. Phys., 2024, **26**, 6557–6573 | **6567**

Table 4 Performance comparison of CIAH and QUOTR for a subset of G2-2/6-31G* with core Hamiltonian guess

| System | CIAH | | | QUOTR |
	KF	JK	N_F	N_F
CH_4	9	35	44	14
CO	17	70	87	22
F_2	7	24	31	12
H_2	3	7	10	5
H_2O	11	43	54	14
HF	12	48	60	13
Li_2	6	18	24	10
LiH	5	15	20	10
N_2	9	34	43	13
NH_3	12	53	65	19
Median	9	34.5	43.5	13
Mean	9.1	34.7	43.8	13.2
Max	17	70	87	22

two programs. The convergence criteria for QUOTR were changed to match the criteria of CIAH, using 1×10^{-6} norm of the unique elements of the gradient, and the criterion on energy change was kept at $1 \times 10^{-9} E_h$. The results for RHF computations of 10 small molecules in the G2 set are displayed in Table 4. In all cases, CIAH and QUOTR found the same solution but the former required on average 3 times more Fock build equivalents.

A deeper breakdown of QUOTR's convergence statistics for the G2-2/6-31G* set is presented in Table 5, where "Before L-BFGS" refers to the line search iterations *before* the first use of a quasi-Newton step, and "Line Search" refers to *all* instances of line search (including those occurring later in the SCF process *e.g.* due to the gradient becoming too large again). On average 7 Fock builds are needed before the gradient is sufficiently reduced to start quasi-Newton steps. This is consistent with the average of 3 line search iterations before stating L-BFGS because each line search takes two Fock builds and we need one initial Fock build. These results are mostly an indication of the quality of the initial guess and the relative simplicity of the electronic structure in this test set. The data in Table 5 also illustrates the efficiency of the L-BFGS/TR combination used in QUOTR since almost half (8.8) of the total number of Fock builds (19.4) are spent in performing steps that ultimately use line searches. The negligible difference between "Line Search" and "Before L-BFGS" statistics indicates that in most cases there is no need for line search after starting L-BFGS/TR steps.

Although the G2-2 test set is composed of systems with relatively simple electronic structure, there is substantial

Table 5 QUOTR convergence statistics breakdown for G2-2/6-31G*

		Median	Max	Mean
N_F	Cumulative	16	69	19.4
	Before L-BFGS	7	21	7.8
	Line search	7	27	8.8
N_I	Cumulative	12	55	14.2
	Before L-BFGS	3	10	3.4
	Line search	3	13	3.9
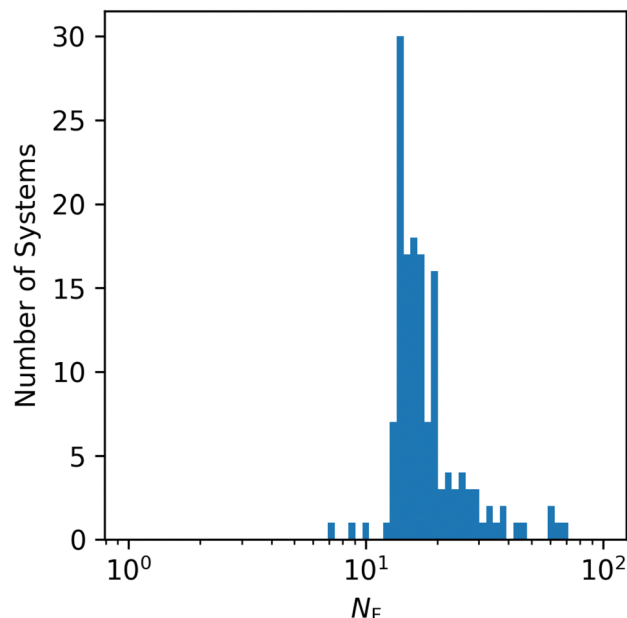
Fig. 1 Histogram of QUOTR's N_F for G2-2/6-31G*.

variance within the set, as illustrated in Fig. 1. Although for most systems convergence is achieved in fewer than 30 Fock builds, there are more than 10 systems for which more Fock builds were required.

Quality of the initial guess unfortunately matters even with very robust solvers. Specifically, the need for random perturbation of the initial guess orbitals was found to be crucial for some systems with geometric symmetry. In particular, we found for $AlCl_3$ that without breaking the symmetry of the extended Hückel orbitals, a local minimum at $-1619.598631 E_h$ was obtained by QUOTR. However, when the perturbation was
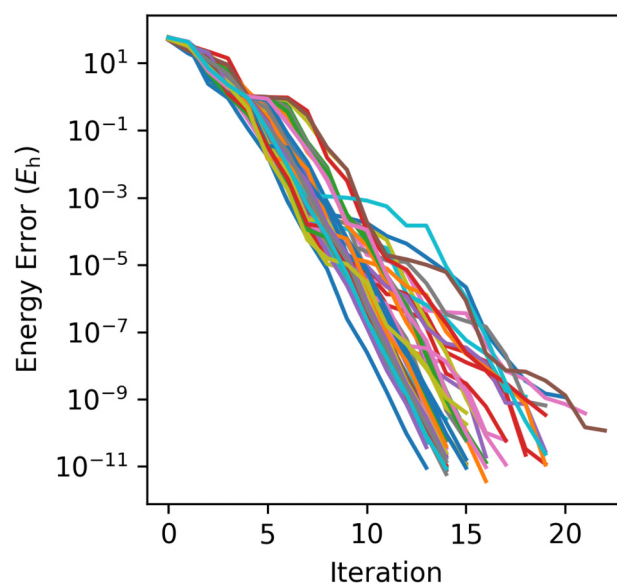
Fig. 2 Convergence of $AlCl_3$ using QUOTR with random perturbation to extended Hückel guess orbitals.

6568 | *Phys. Chem. Chem. Phys.*, 2024, **26**, 6557–6573

This journal is © the Owner Societies 2024

applied, a solution at $-1620.576010 E_h$ was consistently found. To examine how the random perturbations to the initial guess impact convergence, we ran $AlCl_3$ with 50 different seeds for the random number generator. The plot in Fig. 2 shows that when the minimum solution is accessible by symmetry, then QUOTR is robust in converging to the solution.

4.2 Challenging tests

We have now shown that QUOTR is competitive with standard RH/DIIS and competitive or superior to the representative quasi-Newton SCF solvers for the relatively easy test problems (G2 test set). To test the robustness of QUOTR we considered several prototypes of systems where the standard (RH/DIIS) usually fails outright, and even representative quasi-Newton heuristics struggle. We selected 3 types of problems where convergence difficulties often occur: (a) systems with small or vanishing HOMO–LUMO gap, (b) transition metal-containing systems, and (c) f-element containing systems.

4.2.1 System with vanishing HOMO–LUMO gap. To demonstrate the performance of QUOTR for a more challenging problem, we considered a small neuropeptide (1PLW[105]) among several identified by Rudberg et al.,[44] for which the semilocal KS DFT SCF solutions could not be obtained using an RH-based solver. We used QUOTR to obtain converged KS determinants with hybrid (PBE0 and B3LYP), semilocal (PBE), and local (LDA) functionals. The converged energies for the HOMO and LUMO along with the gap are displayed in Table 6 using the 6-31G** orbital basis, with the def2-universal-J basis used for density fitting. For the KS DFT calculations, QUOTR found somewhat lower energy solutions when the initial guess was not perturbed (about 0.36 mE_h for LDA, 0.31 mE_h for PBE, and 0.17 mE_h for B3LYP). Thus, the data in Table 6 is for the unperturbed initial guess. As this system is significantly larger than the G2 tests, it is appropriate to comment on the orthogonality of the final orbitals. We repeated the calculations for 1PLW and computed the orthogonality error, $\|\mathbf{C}^\dagger \mathbf{S} \mathbf{C} - \mathbf{1}\|$, for the converged coefficient matrix. In all cases, this measure was on the order of 1×10^{-12}, indicating that the final solution does not deviate significantly from orthogonality.

The sizeable 7.24 eV gap found for HF nearly vanished with the hybrid DFT functionals (PBE0, B3LYP), with all values within 0.01 eV of the values found in ref. 44. For the LDA and PBE functionals, for which solutions could not be located in ref. 44, QUOTR produced converged solutions with a nearly zero HOMO–LUMO gap! The origin of the vanishing gap in this and other similar biopolymers will be elaborated elsewhere, but we emphasize that the vanishing gap solution is the unphysical but "correct" solution, and QUOTR successfully located it.

Table 6 Frontier orbital energies (eV) and the HOMO–LUMO gap for HF and KS DFT models of the 1PLW popyleptide (see text)

	HF	LDA	PBE	PBE0	B3LYP
HOMO	−6.38	−3.26	−2.53	−2.70	−2.75
LUMO	0.86	−3.25	−2.52	−2.34	−2.47
Gap	7.24	0.01	0.01	0.36	0.28
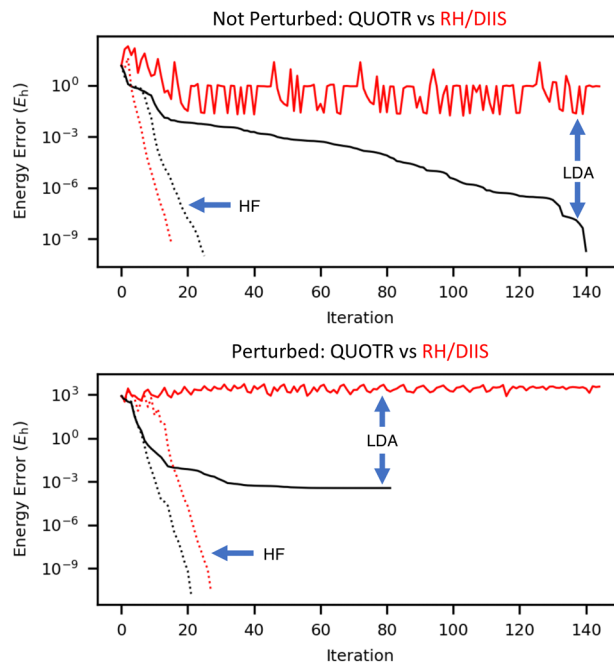
Fig. 3 Convergence of HF and KS DFT for the 1PLW polypeptide (displayed energy error relative to the lowest energy obtained for each method).

A key motivation for the development of QUOTR was the need to understand the origin of such unphysical solutions.

Fig. 3 illustrates how the HF and KS LDA energies converge with QUOTR and RH/DIIS solvers. The two panels in the figure illustrate the impact of the starting orbitals on the solver convergence. While the ultimate outcome—RH/DIIS did not converge for LDA, the rest of the combinations converged—did not depend on whether the starting orbitals were perturbed or not, it took twice as many iterations for QUOTR to converge to the LDA solution without perturbation than with. Note that the perturbation in these cases is applied to all orbitals, not just valence. Both solvers converge at a similar rate for the HF case where the HOMO–LUMO gap is large, with approximately 20 or fewer iterations sufficient for microhartree accuracy. With unperturbed guess the number of Fock builds for RH/DIIS and QUOTR are 17 and 43, respectively; with perturbed guess the corresponding counts are 28 and 34.

While QUOTR manages to locate the LDA solution correctly, its rate of convergence can be relatively slow. We identify regularization of the preconditioner as the likely culprit. Since at the converged KS LDA solution the HOMO–LUMO gap is zero (see Table 6) and the condition number of the exact Hessian is large and grows with the system size, the exact Hessian (hence, the preconditioner) can vary significantly as the solution is approached. The last iteration where the preconditioner was recomputed was 51 (with non-perturbed guess) and 32 (with perturbed guess).

4.2.2 Transition metal-containing molecules. We considered two small systems that are well-known to be challenges to mean-field solvers: Cr_2 and CrC in their lowest-energy singlet states.[20,31,35]

This journal is © the Owner Societies 2024

Phys. Chem. Chem. Phys., 2024, **26**, 6557–6573 | **6569**

Table 7 Performance of various SCF solvers for HF and KS DFT singlet ground states of CrC and Cr$_2$^{a}

|	QUOTR	TRAH	DIIS^b	KDIIS^b	QUOTR^c	CIAH^c
CrC						
RHF	162	377	44^d	47^d	205	164^d
LDA	148	202	—	320^d	107	208
B3LYP	129	300	25^d	440^d	91	240
Cr$_2$						
RHF	249	295	26^d	—	472	160^d
LDA	208	233	20	113^d	144	478^d
B3LYP	123	267	18^d	201^d	169	330

^{a} N_F are reported for each solver. The core Hamiltonian eigenstates were used as the initial guess, unless noted. The def2-TZVPP basis used throughout. ^{b} As implemented in ORCA. ^{c} Initial guess: hcore + 1 Fock build and diagonalize. ^{d} Local minimum.

Table 7 reports the number of Fock builds necessary to converge HF and KS DFT using a variety of solvers. The first set of comparisons juxtaposes QUOTR (implemented in MPQC) against a quasi-Newton TRAH solver (implemented in the Orca program) and 2 variants of DIIS (both implemented in Orca). These computations used the core Hamiltonian initial guess throughout and the default convergence criteria of Orca: 5×10^{-5} for the gradient norm, $1 \times 10^{-6} E_h$ for the energy difference between iterations. Although the core Hamiltonian initial guess is known to be poor, it was chosen to make sure that the same initial orbital set was used to bootstrap computations in MPQC and Orca. The choice of such a poor starting point makes the job of the orbital optimizer even more difficult. It should be noted that TRAH uses a random number in one of the Davidson diagonalization start vectors which helps break symmetry, while for QUOTR we apply a small random unitary rotation to the initial guess (all orbitals, not just valence) with a maximum σ element of 0.01 for these systems. Thus, the initial guess for QUOTR differs from the others by this perturbation, and the QUOTR initial guess is usually higher in energy (approx. 1–2E_h for CrC and 4–4.5E_h for Cr$_2$).

Comparing QUOTR to TRAH, we see that in all cases QUOTR requires fewer Fock builds. The largest error in converged energies for QUOTR was for CrC with B3LYP, which was higher than TRAH by $1.1 \times 10^{-6} E_h$. This error is reasonable since the energies were only converged to $1 \times 10^{-6} E_h$.

The results for DIIS and KDIIS look promising according to the number of Fock builds; however, local minima are very common, so the rapid convergence is deceiving. Only in the LDA case for Cr$_2$ was DIIS able to find a solution that is not a local minimum relative to QUOTR's solution.

The second batch of comparisons juxtaposes QUOTR (in MPQC) against the CIAH solver (in PySCF) using the custom variant of core Hamiltonian (hcore) guess in PySCF, namely the standard hcore guess followed up by a single RH iteration. The first thing to notice is that QUOTR takes more Fock builds than CIAH for RHF, but fewer Fock builds for the other two methods. However, CIAH converges to a local minimum for both systems when using RHF, indicating that QUOTR is more robust and/or faster than CIAH in both cases.
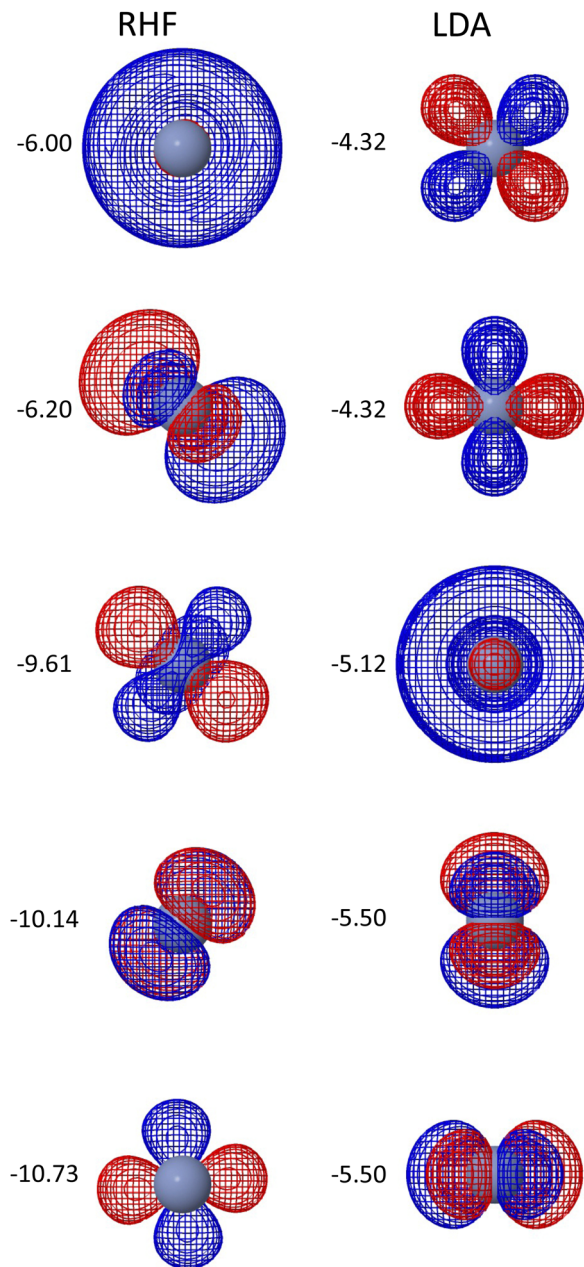
Fig. 4 The valence orbitals viewed along the bond axis and the matching orbital energies (eV) for RHF and LDA singlet ground states of Cr$_2$.

The very large number of Fock builds for Cr$_2$ with RHF merits further investigation. While QUOTR does converge to the lowest energy solution that we could find, it does so at a cost of 249 Fock builds (or 472 for the comparison with PySCF hcore guess). However, this difficulty is not unique to QUOTR, as TRAH also takes nearly 300 Fock builds to achieve the same solution. This is in contrast to the LDA solution, which seems to be generally easier to converge. Fig. 4 provides some insight into why RHF for Cr$_2$ is a difficult case. Namely, the RHF solution located by QUOTR lacks cylindrical symmetry, in contrast to LDA.

To summarize: for CrC and Cr$_2$ QUOTR was able to locate the lowest-energy solution (unlike DIIS and CIAH) and was faster than TRAH.

6570 | *Phys. Chem. Chem. Phys.*, 2024, **26**, 6557–6573

This journal is © the Owner Societies 2024

Table 8 Convergence statistics (N_I, N_F), energy error in kcal mol^{-1} (ΔE), and atomic charges on the fermium atom (Q) for the UHF ground state of [Fm(NO$_3$)]$^{2+}$ obtained by QUOTR starting from a series of quasirandomly-perturbed minimal atomic guesses

	Valence perturbed				All perturbed			
Seed	N_I	N_F	ΔE	Q	N_I	N_F	ΔE	Q
1	124	189	40.19	2.70	131	201	0.58	1.81
2	132	199	40.13	2.70	102	168	0	1.81
3	203	274	0.76	1.81	205	295	0.54	1.81
4	136	191	40.16	2.70	225	307	0.04	1.81
5	189	262	40.16	2.70	209	290	0.25	1.81
6	167	229	40.34	2.70	144	215	0.34	1.81
7	126	170	0.92	1.82	155	243	0.35	1.81

4.2.3 Actinide-containing molecule. For the ultimate challenge, we considered the problem of converging the all-electron UHF orbitals in fermium mononitrate dication ([Fm(NO$_3$)]$^{2+}$), which is a known challenge for the SCF solver.[110] Namely, Penchoff *et al.*[110] located 2 solutions, one with the expected +3 formal charge on Fm but located ∼100 kcal mol^{-1} *above* the correct ground state characterized by a +2 formal charge on Fm.

We used QUOTR with superposition of atomic densities guess orbitals, constructed as described in Section 3. Due to the complex optimization landscape in this system, it was necessary to explore the landscape of solutions by varying the initial guess. Thus the entire set of guess orbitals, or just their valence subset, was perturbed by pseudorandom unitaries generated using seven different integers (between 123 and 129) as the random engine seed. The results are displayed in Table 8 and Fig. 5. The column labeled "energy error" is relative to the lowest energy

solution that we obtained, which was for seed 2 with all orbitals perturbed (total energy −35 045.703224E_h). As can be seen, two types of solutions were found; the lower energy one has a formal +2 charge on Fm, and the other, roughly 40 kcal mol^{-1} higher in energy, has a +3 charge on Fm. The ground state energy agreed quite well with the value located by Penchoff *et al.* in ref. 110 using Molpro's RH/DIIS SCF solver using complicated guess obtained by merging converged fragment MOs (in fact, the ground state energy located by QUOTR is slightly lower in energy). Unfortunately, it was only possible to obtain 10 significant digits of precision in the energy, due to the impact of roundoff errors and the non-determinism of the Fock matrix construction in MPQC. As we did for 1PLW, we ran separate calculations to check the orthogonality of the converged orbitals. Again, we found errors on the order of 1×10^{-12} in all cases. Clearly, all electron computations in heavy element systems with Gaussian AO bases that have high condition numbers will be increasingly untenable in double precision.

There are clearly many outstanding challenges suggested by the computational experiments on this actinide-containing complex. In particular, the sensitivity of the final solution to the initial guess suggests that various global (*e.g.*, stochastic[35]) approaches to the orbital optimization should be considered.

5 Summary

We have presented a state-of-the-art solver for quasi-Newton unitary optimization that combines the preconditioned L-BFGS orbital update with the trust-region step restriction method. The exploitation of the low-rank structure of the L-BFGS Hessian, including in solving the trust-region (sub) problem, makes the QUOTR solver remarkably efficient, approaching the efficiency of the mainstream RH/DIIS heuristics when applied to problems with easy optimization landscapes (like the standard G2 test set). When applied to problems with complex optimization landscapes (problems with vanishing HOMO–LUMO gaps, d- and f-element containing molecules) QUOTR matches or exceeds the robustness of representative quasi-Newton solvers, all at a significantly lower computational cost due to avoiding the exact Hessian evaluation.

While QUOTR guarantees convergence to a local stationary point, it is not able to guarantee global convergence due to the nonconvexity of the energy. However, its efficiency makes it a robust building block for even sophisticated solvers that combine efficient local minimum search with global (*e.g.*, stochastic) landscape traversal.

Fig. 5 Convergence of X2C-UHF for [Fm(NO$_3$)]$^{2+}$ starting from a series of quasirandomly-perturbed minimal atomic guess orbitals; for each panel the energy error is defined relative to the lowest energy obtained in that panel's subset.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This journal is © the Owner Societies 2024

Phys. Chem. Chem. Phys., 2024, **26**, 6557–6573 | **6571**

Notes and references

1 D. R. Hartree, *Rep. Prog. Phys.*, 1947, **11**, 113–143.

2 C. C. J. Roothaan, *Rev. Mod. Phys.*, 1951, **23**, 69–89.

3 R. McWeeny, *Proc. R. Soc. Lond. A*, 1956, **235**, 496–509.

4 B. Levy and G. Berthier, *Int. J. Quantum Chem.*, 1968, **2**, 307–319.

5 I. H. Hillier and V. R. Saunders, *Proc. R. Soc. Lond. A*, 1970, **320**, 161–173.

6 I. H. Hillier and V. R. Saunders, *Int. J. Quantum Chem.*, 1970, **4**, 503–518.

7 B. Levy, *Chem. Phys. Lett.*, 1973, **18**, 59–62.

8 R. Seeger and J. A. Pople, *J. Chem. Phys.*, 1976, **65**, 265–271.

9 J. Douady, Y. Ellinger, R. Subra and B. Levy, *J. Chem. Phys.*, 1980, **72**, 1452–1462.

10 P. Pulay, *Chem. Phys. Lett.*, 1980, **73**, 393–398.

11 G. B. Bacskay, *Chem. Phys.*, 1981, **61**, 385–404.

12 G. B. Bacskay, *Chem. Phys.*, 1982, **65**, 383–396.

13 P. Pulay, *J. Comput. Chem.*, 1982, **3**, 556–560.

14 M. Head-Gordon and J. A. Pople, *J. Phys. Chem.*, 1988, **92**, 3063–3069.

15 T. H. Fischer and J. Almlof, *J. Phys. Chem.*, 1992, **96**, 9768–9774.

16 R. Shepard, *Theor. Chim. Acta*, 1993, **84**, 343–351.

17 A. P. Rendell, *Chem. Phys. Lett.*, 1994, **229**, 204–210.

18 A. T. Wong and R. J. Harrison, *J. Comput. Chem.*, 1995, **16**, 1291–1300.

19 G. Chaban, M. W. Schmidt and M. S. Gordon, *Theor. Chim. Acta*, 1997, **97**, 88–95.

20 A. D. Daniels and G. E. Scuseria, *Phys. Chem. Chem. Phys.*, 2000, **2**, 2173–2176.

21 T. Van Voorhis and M. Head-Gordon, *Mol. Phys.*, 2002, **100**, 1713–1721.

22 J. VandeVondele and J. Hutter, *J. Chem. Phys.*, 2003, **118**, 4365–4369.

23 L. Thøgersen, J. Olsen, D. Yeager, P. Jørgensen, P. Sałek and T. Helgaker, *J. Chem. Phys.*, 2004, **121**, 16.

24 L. Thøgersen, J. Olsen, A. Köhn, P. Jørgensen, P. Sałek and T. Helgaker, *J. Chem. Phys.*, 2005, **123**, 074103.

25 C. Yang, J. C. Meza and L.-W. Wang, *SIAM J. Sci. Comput.*, 2007, **29**, 1854–1875.

26 P. Sałek, S. Høst, L. Thøgersen, P. Jørgensen, P. Manninen, J. Olsen, B. Jansík, S. Reine, F. Pawłowski, E. Tellgren, T. Helgaker and S. Coriani, *J. Chem. Phys.*, 2007, **126**, 114110.

27 V. Weber, J. Vande Vondele, J. Hutter and A. M. N. Niklasson, *J. Chem. Phys.*, 2008, **128**, 084113.

28 S. Høst, J. Olsen, B. Jansík, L. Thøgersen, P. Jørgensen and T. Helgaker, *J. Chem. Phys.*, 2008, **129**, 124106.

29 K. Baarman and J. VandeVondele, *J. Chem. Phys.*, 2011, **134**, 244104.

30 Q. Sun, *arXiv*, 2017, preprint, arXiv:161008423 Phys, DOI: **10.48550/arXiv.1610.08423**.

31 B. Helmich-Paris, *J. Chem. Phys.*, 2021, **154**, 164104.

32 T. Nottoli, J. Gauss and F. Lipparini, *Mol. Phys.*, 2021, **119**, e1974590.

33 A. V. Ivanov, E. Ö. Jónsson, T. Vegge and H. Jónsson, *Comput. Phys. Commun.*, 2021, **267**, 108047.

34 C. Seidl and G. M. J. Barca, *J. Chem. Theory Comput.*, 2022, **18**, 4164–4176.

35 L. B. Dittmer and A. Dreuw, *J. Chem. Phys.*, 2023, **159**, 134104.

36 S. Vavasis, *Nonlinear Optimization: Complexity Issues*, Oxford University Press, 1991.

37 N. Boumal, P.-A. Absil and C. Cartis, *IMA J. Numer. Anal.*, 2019, **39**, 1–33.

38 G. G. Hall, *Proc. R. Soc. Lond. A*, 1951, **205**, 541–552.

39 D. G. Anderson, *J. ACM*, 1965, **12**, 547–560.

40 K. N. Kudin, G. E. Scuseria and E. Cancès, *J. Chem. Phys.*, 2002, **116**, 8255–8261.

41 A. J. Garza and G. E. Scuseria, *J. Chem. Phys.*, 2012, **137**, 054110.

42 R. J. Harrison, *J. Comput. Chem.*, 2004, **25**, 328–334.

43 C. Kollmar, *J. Chem. Phys.*, 1996, **105**, 8204–8212.

44 E. Rudberg, *J. Phys.: Condens. Matter*, 2012, **24**, 072202.

45 C. Köppl and H.-J. Werner, *J. Chem. Theory Comput.*, 2016, **12**, 3122–3134.

46 C. A. Lewis, J. A. Calvin and E. F. Valeev, *J. Chem. Theory Comput.*, 2016, **12**, 5868–5880.

47 X. Wang, C. A. Lewis and E. F. Valeev, *J. Chem. Phys.*, 2020, **153**, 124116.

48 A. T. B. Gilbert, N. A. Besley and P. M. W. Gill, *J. Phys. Chem. A*, 2008, **112**, 13164–13171.

49 R. E. Stanton, *J. Chem. Phys.*, 1981, **75**, 3426–3432.

50 R. Fletcher, *Mol. Phys.*, 1970, **19**, 55–63.

51 M. Chupin, M.-S. Dupuy, G. Legendre and É. Séré, *ESAIM: Math. Modell. Numer. Anal.*, 2021, **55**, 2785–2825.

52 J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 2nd edn, 2006.

53 C. Kollmar, *Int. J. Quantum Chem.*, 1997, **62**, 617–637.

54 X. Hu and W. Yang, *J. Chem. Phys.*, 2010, **132**, 054109.

55 J. B. Francisco, J. M. Martínez and L. Martínez, *J. Chem. Phys.*, 2004, **121**, 10863.

56 O. Burdakov, L. Gong, S. Zikrin and Y.-x Yuan, *Math. Prog. Comp.*, 2017, **9**, 101–134.

57 T. A. Claxton and N. A. Smith, *Theor. Chim. Acta*, 1971, **22**, 399–402.

58 D. H. Sleeman, *Theor. Chim. Acta*, 1968, **11**, 135–144.

59 J. Olsen and P. Jørgensen, *J. Chem. Phys.*, 1982, **77**, 6109–6130.

60 D. A. Kreplin, P. J. Knowles and H.-J. Werner, *J. Chem. Phys.*, 2019, **150**, 194106.

61 Y. Yao and C. J. Umrigar, *J. Chem. Theory Comput.*, 2021, **17**, 4183–4194.

62 P. Jo/rgensen and J. Simons, *J. Chem. Phys.*, 1983, **79**, 334–357.

63 H.-J. A. Jensen and P. Jørgensen, *J. Chem. Phys.*, 1984, **80**, 1204–1214.

64 R. Fletcher, *Practical Methods of Optimization*, Wiley, New York, 1980, vol. 1.

65 D. Thouless, *Nucl. Phys.*, 1960, **21**, 225–232.

66 B. Levy, *Chem. Phys. Lett.*, 1969, **4**, 17–19.

67 T. Helgaker, P. Jørgensen and J. Olsen, *Molecular Electronic-Structure Theory*, John Wiley & Sons, Ltd, Chichester, UK, 1st edn, 2000.

68 C. Moler and C. Van Loan, *SIAM Rev.*, 1978, **20**, 801–836.

69 R. H. Byrd, J. Nocedal and R. B. Schnabel, *Math. Program.*, 1994, **63**, 129–156.

70 P.-O. Löwdin, *Adv. Phys.*, 1956, **5**, 1–171.

71 T. Abrudan, J. Eriksson and V. Koivunen, *Signal Process.*, 2009, **89**, 1704–1714.

72 R. Hoffmann, *J. Chem. Phys.*, 1963, **39**, 1397–1412.

73 J. H. Ammeter, H. B. Buergi, J. C. Thibeault and R. Hoffmann, *J. Am. Chem. Soc.*, 1978, **100**, 3686–3692.

74 S. Lehtola, *J. Chem. Theory Comput.*, 2019, **15**, 1593–1604.

75 P. Norman and H. J. A. Jensen, *Chem. Phys. Lett.*, 2012, **531**, 229–235.

76 C. F. Fischer, *At. Data Nucl. Data Tables*, 1972, **4**, 301–399.

77 C. Peng, C. A. Lewis, X. Wang, M. C. Clement, K. Pierce, V. Rishi, F. Pavošević, S. Slattery, J. Zhang, N. Teke, A. Kumar, C. Masteran, A. Asadchev, J. A. Calvin and E. F. Valeev, *J. Chem. Phys.*, 2020, **153**, 044120.

78 R. Ditchfield, W. J. Hehre and J. A. Pople, *J. Chem. Phys.*, 1971, **54**, 724–728.

79 W. J. Hehre, R. Ditchfield and J. A. Pople, *J. Chem. Phys.*, 1972, **56**, 2257–2261.

80 P. C. Hariharan and J. A. Pople, *Theor. Chim. Acta*, 1973, **28**, 213–222.

81 J. D. Dill and J. A. Pople, *J. Chem. Phys.*, 1975, **62**, 2921–2923.

82 J. S. Binkley and J. A. Pople, *J. Chem. Phys.*, 1977, **66**, 879–880.

83 M. S. Gordon, J. S. Binkley, J. A. Pople, W. J. Pietro and W. J. Hehre, *J. Am. Chem. Soc.*, 1982, **104**, 2797–2803.

84 M. M. Francl, W. J. Pietro, W. J. Hehre, J. S. Binkley, M. S. Gordon, D. J. DeFrees and J. A. Pople, *J. Chem. Phys.*, 1982, **77**, 3654–3665.

85 R. Krishnan, J. S. Binkley, R. Seeger and J. A. Pople, *J. Chem. Phys.*, 1980, **72**, 650–654.

86 A. D. McLean and G. S. Chandler, *J. Chem. Phys.*, 1980, **72**, 5639–5648.

87 T. Clark, J. Chandrasekhar, G. W. Spitznagel and P. V. R. Schleyer, *J. Comput. Chem.*, 1983, **4**, 294–301.

88 G. W. Spitznagel, T. Clark, P. von Ragué Schleyer and W. J. Hehre, *J. Comput. Chem.*, 1987, **8**, 1109–1116.

89 F. Weigend and R. Ahlrichs, *Phys. Chem. Chem. Phys.*, 2005, **7**, 3297.

90 W. A. de Jong, R. J. Harrison and D. A. Dixon, *J. Chem. Phys.*, 2001, **114**, 48.

91 R. Feng and K. A. Peterson, *J. Chem. Phys.*, 2017, **147**, 084108.

92 F. Weigend, *Phys. Chem. Chem. Phys.*, 2006, **8**, 1057.

93 K. L. Schuchardt, B. T. Didier, T. Elsethagen, L. Sun, V. Gurumoorthi, J. Chase, J. Li and T. L. Windus, *J. Chem. Inf. Model.*, 2007, **47**, 1045–1052.

94 T. Tsuchiya, M. Abe, T. Nakajima and K. Hirao, *J. Chem. Phys.*, 2001, **115**, 4463–4472.

95 M. C. Clement, X. Wang and E. F. Valeev, *J. Chem. Theory Comput.*, 2021, **17**, 7406–7415.

96 L. A. Curtiss, K. Raghavachari, P. C. Redfern and J. A. Pople, *J. Chem. Phys.*, 1997, **106**, 1063–1079.

97 R. D. Johnson, Computational Chemistry Comparison and Benchmark Database, NIST Standard Reference Database 101, 2002.

98 M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. A. Montgomery Jr., J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, T. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. M. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, O. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski and D. J. Fox, *Gaussian 09*, Gaussian, Inc., 2013.

99 A. Petrone, D. B. Williams-Young, S. Sun, T. F. Stetina and X. Li, *Eur. Phys. J. B*, 2018, **91**, 169.

100 S. Lehtola, C. Steigemann, M. J. Oliveira and M. A. Marques, *SoftwareX*, 2018, **7**, 1–5.

101 M. E. Mura and P. J. Knowles, *J. Chem. Phys.*, 1996, **104**, 9848–9858.

102 V. I. Lebedev and D. N. Laikov, *Dokl. Math.*, 1999, **59**, 477–481.

103 P. A. M. Dirac, *Math. Proc. Cambridge Philos. Soc.*, 1930, **26**, 376–385.

104 S. H. Vosko, L. Wilk and M. Nusair, *Can. J. Phys.*, 1980, **58**, 1200–1211.

105 I. Marcotte, F. Separovic, M. Auger and S. M. Gagné, *Biophys. J.*, 2004, **86**, 1587–1600.

106 H. M. Berman, *Nucleic Acids Res.*, 2000, **28**, 235–242.

107 F. Neese, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2022, **12**, e1606.

108 W. Liu and D. Peng, *J. Chem. Phys*, 2009, **131**, 031104.

109 D. Peng, N. Middendorf, F. Weigend and M. Reiher, *J. Chem. Phys*, 2013, **138**, 184105.

110 D. A. Penchoff, C. C. Peterson, M. S. Quint, J. D. Auxier, G. K. Schweitzer, D. M. Jenkins, R. J. Harrison and H. L. Hall, *ACS Omega*, 2018, **3**, 14127–14143.

111 Q. Sun, X. Zhang, S. Banerjee, P. Bao, M. Barbry, N. S. Blunt, N. A. Bogdanov, G. H. Booth, J. Chen, Z.-H. Cui, J. J. Eriksen, Y. Gao, S. Guo, J. Hermann, M. R. Hermes, K. Koh, P. Koval, S. Lehtola, Z. Li, J. Liu, N. Mardirossian, J. D. McClain, M. Motta, B. Mussard, H. Q. Pham, A. Pulkin, W. Purwanto, P. J. Robinson, E. Ronca, E. R. Sayfutyarova, M. Scheurer, H. F. Schurkus, J. E. T. Smith, C. Sun, S.-N. Sun, S. Upadhyay, L. K. Wagner, X. Wang, A. White, J. D. Whitfield, M. J. Williamson, S. Wouters, J. Yang, J. M. Yu, T. Zhu, T. C. Berkelbach, S. Sharma, A. Y. Sokolov and G. K.-L. Chan, *J. Chem. Phys.*, 2020, **153**, 024109.

This journal is © the Owner Societies 2024

Phys. Chem. Chem. Phys., 2024, **26**, 6557–6573 | **6573**