A near linear-scaling smooth local coupled cluster algorithm for electronic structure

Joseph E. Subotnik[a)]
Biophysics Program, University of California, Berkeley, California 94720

Alex Sodt[b)] and Martin Head-Gordon[c)]
Department of Chemistry, University of California, Berkeley, California 94720
and Chemical Sciences Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720

(Received 30 March 2006; accepted 19 July 2006; published online 21 August 2006)

We demonstrate near linear scaling of a new algorithm for computing smooth local coupled-cluster singles-doubles (LCCSD) correlation energies of quantum mechanical systems. The theory behind our approach has been described previously, [J. Subotnik and M. Head-Gordon, J. Chem. Phys. **123**, 064108 (2005)], and requires appropriately multiplying standard iterative amplitude equations by a bump function, creating local amplitude equations (which are smooth according to the implicit function theorem). Here, we provide an example that this theory works in practice: we show that our algorithm leads to smooth potential energy surfaces and yields large computational savings. As an example, we apply our LCCSD approach to measure the post-MP2 correction to the energetic gap between two different alanine tetrapeptide conformations. © *2006 American Institute of Physics*. [DOI: 10.1063/1.2336426]

I. INTRODUCTION

The correlation of electrons above and beyond the Hartree-Fock mean-field level is crucial towards understanding bond making and bond breaking in chemistry. Hartree-Fock theory correctly predicts stable equilibrium geometries; however, the correlation energy is crucial for determining relative energies and barrier heights accurately enough to be chemically useful. Unfortunately, correlation schemes are expensive almost by definition. The exact full configuration interaction (FCI) scales exponentially with the number of electrons. Widely used approximations such as singles and doubles coupled cluster theory[2–5] (CCSD) and its correction for triple excitations, CCSD(T),[6,7] still scale with the sixth and seventh powers of molecular size, respectively.

In order to make correlation energies computationally tractable, many local correlation schemes have been introduced over the last 20 years (mostly at the perturbative or MP2 level) beginning with Saebo and Pulay,[8–12] and later developed by Carter,[13] Friesner,[14] Schütz and Werner,[15–27] and many others. The basic premise behind local correlation calculations is to localize all canonical molecular orbitals, and then correlate only those localized molecular orbitals which are close to each other in physical, three-dimensional space ($\mathbb{R}^3$). We refer the reader to a previous publication[1] for a more comprehensive history of the development of local correlation theory, along with a list of the many contributors who pushed it forward.

In this paper, we focus on local coupled-cluster theory. Several, local coupled-cluster algorithms have been proposed in the past,[21,28–32] of which the Schütz-Werner[15,21,31,32] algorithm has arguably received the most attention. The Schütz-Werner local coupled-cluster algorithm was first implemented in 1996 and then made to scale linearly in 2001; its approach was based very closely on the earlier local MP2 models of Saebo and Pulay.[12] The central idea behind this approach is to solve the standard CCSD equations in a basis of orthonormal, localized occupied orbitals and redundant, nonorthogonal localized virtual orbitals. One first divides the fourfold groupings $(ijab)$ of orbitals into domains of strong, weak, very weak, and negligible sets based on separation in $\mathbb{R}^3$; then, one either performs CCSD, MP2, multipole expansion, or one simply ignores the corresponding amplitudes t_{ij}^{ab}, respectively. The Schütz-Werner method was a remarkable accomplishment in local correlation theory, as the authors produced a very fast, linear-scaling algorithm which was accurate at near equilibrium geometries. Furthermore, Schütz and Werner were able to calculate gradients with their method, which allowed the minimization of structures over small changes in geometry—a very useful property of their algorithm.

The essential shortcoming of the Pulay-Schütz -Werner approach, however, is that the potential energy surfaces are discontinuous.[33] As a consequence, the gradient calculated by their algorithm is defined almost everywhere, but not quite. The reason for this discontinuity is clear. By explicitly choosing which groupings of orbitals should be treated at a given level of theory, the algorithm makes a highly geometry-dependent choice. By definition, this algorithm is not a model chemistry as defined by Pople.[34] When the nuclei move around, domains of orbital groupings will often change and, consequentially, the correlation energy changes discontinuously. Though these discontinuities have been shown to be small (≈millihartree) for small systems, they may well be larger for big systems. In addition, these discon-

[a)]Electronic mail: subotnik@post.harvard.edu
[b)]Electronic mail: alexsodt@berkeley.edu
[c)]Electronic mail: mhg@chem.berkeley.edu

tinuities are not discrete points in configuration space, but rather surfaces of low dimension, which are not easily identified. If one wants to avoid these discontinuities, one may keep all domains fixed over the whole configuration space, but, in that case, one can get hysteresis in the correlation energy.

Two consequences arise from this deficiency. First, computational chemists cannot always optimize geometries with confidence according to the Pulay-Werner scheme. Geometries have certainly been successfully optimized when the domains are fixed and thus the potential energy surfaces are smooth;[35] but for large and/or subtle changes in geometry, where the best domains are not obvious and should not be held static, geometric optimization is not very practical. Second, the Pulay-Werner potential energy surface is not suitable at all for propagation of time dynamics. On the one hand, if one allows the domains to be chosen on the fly, and then one passes over a sharp discontinuity, one no longer conserves energy and the trajectory will be unstable. On the other hand, even if the domains are fixed, a large closed orbit in nuclear configuration space again may introduce hysteresis in the potential energy function. This is because the potential energy must be a symmetric function of the nuclear coordinates of identical particles in order to avoid hysteresis. For example, suppose there are two hydrogen atoms which are not symmetrically located in a molecule. And now, consider a trajectory wherein these two hydrogen atoms are slowly displaced in space until they have swapped positions (and all other atoms are unmoved as well). The new geometry is now identical to the old geometry, but the domains are different, and we will likely violate energy conservation, implying that the trajectory is unphysical. Thus, hydrogen exchange cannot be modeled. For these reasons, dynamics is impractial using the Pulay-Werner scheme.

In a previous article,[1] we proved mathematically that a viable approach towards constructing local correlation algorithms with smooth potential energy surfaces is to employ bump functions and invoke the implicit function theorem from differential geometry. In order to approximate any given level of correlation theory, one simply localizes all occupied and virtual orbitals, and then applies smooth bump functions where appropriate to make sparse the amplitude equations. In our previous work, we demonstrated that this approach worked for the case of N_2 separation treated at the coupled-cluster doubles (CCD) level. Furthermore, in that paper, we argued that such local approximations should generally be very accurate and allow significant speedups of the calculations for big system. In this paper, we demonstrate definitively that such an approach does, in fact, yield highly accurate calculations; furthermore, and most importantly, we show that such an approach leads to enormous savings in computational time. From the perspective of computational efficiency, our algorithm is similar to the Pulay-Werner scheme and should be capable of formal linear scaling. Indeed, for alkanes of increasing length in a cc-pVDZ basis, we can almost achieve linear scaling with not yet fully optimized code.

II. THEORY

We now briefly summarize our theoretical approach towards a smooth local coupled-cluster algorithm. First, we work in a basis of localized, orthonormal occupied and virtual orbitals (see Sec. II A). Second, we define a smooth one-dimensional bump function that interpolates smoothly between 0 and 1 (see Sec. II B). Third, like the Pulay-Saebo-Werner scheme, our algorithm picks a localization criteria by which we partition orbitals and amplitudes into strongly interacting, moderately interacting, weakly interacting, and negligibly interacting groupings. We combine these localization criteria with our one-dimensional bump function to create an orbital bump function that is nonzero only for quartets of orbitals that are near each other (see Sec. II C). Fourth, we multiply the coupled-cluster iterative equations appropriately so that the equations become local (see Sec. II D). Fifth, we solve the iterative equations only for the strongly interacting amplitudes (see Sec. II E). Our solution will automatically be a smooth function of the nuclear coordinates according to the implicit function theorem.

A. Choice of orbitals

In our local coupled-cluster singles-doubles (LCCSD) algorithm, we have used only orthonormal localized orbitals (both occupied and virtual). For our local correlation energy and amplitudes to be smooth functions of nuclear position, it is crucial that the orbitals vary smoothly with nuclear position. To that end, we used the prescription of Subotnik *et al.*[36] for orthonormal, localized extra-valence hard virtual orbitals.

For the occupied space and valence virtual space, we required well-defined, smoothly varying orbitals. Unfortunately, Boys orbitals can be degenerate. For instance, if we consider the molecule N_2 aligned along the z axis, there are two π-like Boys localized orbitals, π_x and π_y. Now, these may be mixed to form

$$\pi_x' = \cos(\alpha)\pi_x + \sin(\alpha)\pi_y,$$

$$\pi_y' = \sin(\alpha)\pi_x - \cos(\alpha)\pi_y,$$

which are also clearly valid Boys localized orbitals. In this example, the Boys orbitals are not unique because the cylindrical symmetry of the molecule yields a continuous degeneracy. For this reason, we localized the occupied and valence virtual orbitals with a Boys-type localization function that maximized not just the sum of the variances of the orbitals, but also effectively the sum of the squares of their second moments. Let η denote a localized molecular orbital. We called our localization function (which must be optimized) $\zeta_{\mathrm{BoysQuad}}$:

$$\zeta_{\mathrm{Boys}}(\eta_1, \ldots, \eta_m) = \sum_{i=1}^{m} \langle \eta_i|x|\eta_i\rangle^2 + \sum_{i=1}^{m} \langle \eta_i|y|\eta_i\rangle^2 + \sum_{i=1}^{m} \langle \eta_i|z|\eta_i\rangle^2, \quad (1)$$

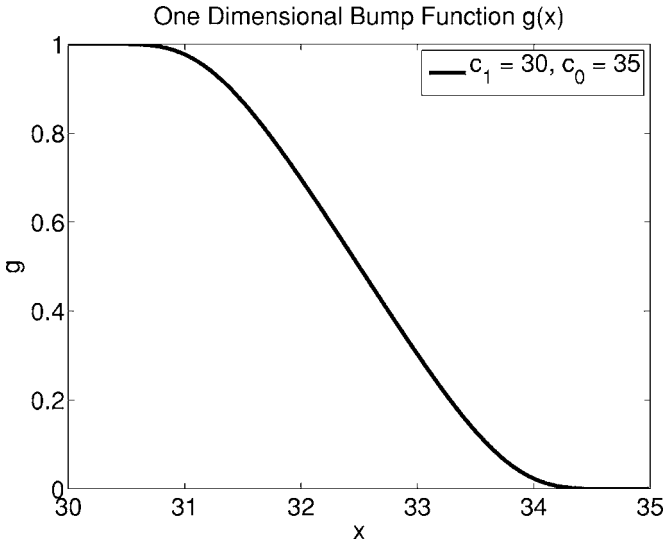

FIG. 1. The smooth bump function which interpolates smoothly between 0 and 1. See text for details.

$$\zeta_{\mathrm{BoysQuad}}(\eta_1,\ldots,\eta_m)=\zeta_{\mathrm{Boys}}+\frac{1}{\lambda_x}\sum_{i=1}^{m}\langle\eta_i|x^2|\eta_i\rangle^2+\frac{1}{\lambda_y}\sum_{i=1}^{m}\langle\eta_i|y^2|\eta_i\rangle^2+\frac{1}{\lambda_z}\sum_{i=1}^{m}\langle\eta_i|z^2|\eta_i\rangle^2, \tag{2}$$

$$\lambda_x=\sum_{i=1}^{m}\langle\phi_i^{\mathrm{can}}|x^2|\phi_i^{\mathrm{can}}\rangle \quad \text{similarly for } \lambda_y,\lambda_z. \tag{3}$$

This choice of $\zeta_{\mathrm{BoysQuad}}$ breaks continuous symmetries and produces unique, well-isolated, smooth, localized, orthonormal occupied and valence virtual orbitals. When combined with the smooth, hard-virtual space produced by the algorithm in Ref. 36, we obtained smoothly varying localized orthonormal orbitals that allowed our algorithm to proceed.

B. The one-dimensional bump function

The essential tool we need to construct smooth local-correlation equations is a bump function. For a one-dimensional bump function, one is free to choose any smooth function of one variable x, which interpolates smoothly from 1 to 0 over the range of $x\in[c_1,c_0]$. In our work, we define the one-dimensional bump function ζ as

$$\zeta(x)=1, \quad x<c_1,$$

$$\zeta(x)=\frac{1}{1+e^{-[(c_1-c_0)/(c_1-x)+(c_1-c_0)/(x-c_0)]}}, \quad x\in(c_1,c_0),$$

$$\zeta(x)=0, \quad x>c_0. \tag{4}$$

ζ is infinitely differentiable everywhere. ζ is plotted in Fig. 1 for $c_1=30$, and $c_0=35$.

C. Localization criteria and definition of amplitude classes

Similar to the choice of one-dimensional bump function, the criteria by which we bump a pair or quartet of localized orbitals is arbitrary. For instance, one can bump by Coulomb or exchange two-electron integrals, by atomic orbital (AO) contribution, or by distance between orbital centers. Certainly, bumping by integrals would be the best choice, as we have argued before,[1] and which will be discussed more in Sec. VI.

In the present paper, however, for maximal speed and ease of implementation, we have employed the last option, i.e., sorting pairs of orbitals by distances between their centers, which is very efficient with little extra work. Let η_i and η_j be localized molecular orbitals. We make the following definitions for our orbital bump functions:

$$\mathbf{r}_i=\langle\eta_i|\mathbf{r}|\eta_i\rangle, \tag{5}$$

$$g_{ij}=\zeta((\mathbf{r}_i-\mathbf{r}_j)^2), \tag{6}$$

$$g_{ijab}=g_{ij}g_{ab}g_{ib}g_{ja}g_{ia}g_{jb}. \tag{7}$$

Now, in our LCCSD algorithm, we want to have classes of strong and moderately strong t amplitudes. As such, we must construct two independent orbital bump functions corresponding to these two different types of amplitude classes. Thus, we make the following definitions. First, we define a strong bump function by

$$c_0^s=35a_0^2(\sqrt{c_0^s}\approx 3.13\ \text{Å}), \tag{8}$$

$$c_1^s=30a_0^2(\sqrt{c_1^s}\approx 2.90\ \text{Å}), \tag{9}$$

and, second, a moderate bump function by

$$c_0^m=70a_0^2(\sqrt{c_0^m}\approx 4.43\ \text{Å}), \tag{10}$$

$$c_1^m=60a_0^2(\sqrt{c_1^m}\approx 4.10\ \text{Å}). \tag{11}$$

We now have obvious definitions for amplitude classes. t_{ij}^{ab} is a *strong t* amplitude if $g_{ijab}^s>0$. t_{ij}^{ab} is a *moderate t* amplitude if $g_{ijab}^m>0$ but $g_{ijab}^s=0$. Note that $g_{ij}^s=0$ if the localized molecular orbitals η_i and η_j are separated by more than 3.13 Å. This fact will shortly be used to ensure the locality of our LCCSD equations.

For completeness, we should now define the remaining sets of t amplitudes, namely, the weakly and negligibly interacting t amplitudes. These classes include those t_{ij}^{ab} for which $g_{ijab}^s=g_{ijab}^m=0$. In that case, t_{ij}^{ab} is said to be weak if $|\langle\eta_i\eta_j||\eta_a\eta_b\rangle|>10^{-16}$ and negligible otherwise. This will be discussed more fully below.

D. The smoothly bumped LCCSD equations

Once we have chosen orbital bump functions, the "right" way to make the CCSD equations local and sparse is obvious. Essentially, the equations require bumping the t amplitudes on the right hand side of the iterative equations:

$$\tilde{t}_i^a=g_{ia}^s t_i^a, \tag{12}$$

$$\tilde{t}_{ij}^{ab} = g_{ijab}^{s} t_{ij}^{ab}, \tag{13}$$

and for efficiency we must sometimes bump the product of two T_1 amplitudes in order to achieve formal linear scaling (even though this is not a physically motivated ansatz):

$$t_i^a t_j^b \rightarrow g_{ijab} t_i^a t_j^b = g_{ij} g_{ab} g_{ib} g_{ja} \tilde{t}_i^a \tilde{t}_j^b. \tag{14}$$

Lastly, there are four contractions for which we must explicity bump the integral in order to enforce locality. The reasons for which this is necessary should be clear from Sec. III B. We note that the need to bump nonlocal pairs of single excitations and certain two-electron integrals was previously recognized by Werner and co-workers.[15,21]

For concreteness, we now write down the smoothly bumped local CCSD equations. Though perhaps not obvious, we emphasize that all t amplitudes on the right hand side of the equations below are bumped—either *explicitly* or *implicitly* (through the integrals). We follow very closely the formalism of Stanton *et al.*[37] Explicitly, the iterative equation for T_1 reads

$$\begin{aligned} D_i^a t_i^a = f_{ia} + g_{ia}^{s} \Bigg\{ & \sum_e \tilde{t}_i^e f_{ae}(1-\delta_{ae}) - \sum_m \tilde{t}_m^a f_{mi}(1-\delta_{mi}) \\ & + \sum_e \tilde{t}_i^e \mathcal{F}_{ae} - \sum_m \tilde{t}_m^a \mathcal{F}_{mi} + \sum_{me} \tilde{t}_{im}^{ae} \mathcal{F}_{me} \\ & - \sum_{nf} t_n^f \langle na||if\rangle g_{naif}^{s} - \frac{1}{2}\sum_{mef} \tilde{t}_{im}^{ef} \langle ma||ef\rangle \\ & - \frac{1}{2}\sum_{men} \tilde{t}_{mn}^{ae} \langle nm||ei\rangle \Bigg\}. \end{aligned} \tag{15}$$

For T_2,

$$\begin{aligned} D_{ij}^{ab} t_{ij}^{ab} = & \langle ij||ab\rangle + g_{ijab}^{m} \{ P_-(ab) \sum_e \tilde{t}_{ij}^{ae} f_{be}(1-\delta_{be}) - P_-(ij) \sum_m \tilde{t}_{im}^{ab} f_{mj}(1-\delta_{mj}) \} \\ & + g_{ijab}^{s} \Bigg\{ P_-(ab) \sum_e \tilde{t}_{ij}^{ae} \Bigg(\mathcal{F}_{be} - \frac{1}{2}\sum_m \tilde{t}_m^b \mathcal{F}_{me} \Bigg) \\ & - P_-(ij) \sum_m \tilde{t}_{im}^{ab} \Bigg(\mathcal{F}_{mj} + \frac{1}{2}\sum_e \tilde{t}_j^e \mathcal{F}_{me} \Bigg) + \frac{1}{2}\sum_{mn} \tau_{mn}^{ab} W_{mnij} + \frac{1}{2}\sum_{ef} \tau_{ij}^{ef} W_{abef} + P_-(ij) P_-(ab) \sum_{me} (\tilde{t}_{im}^{ae} W_{mbej} - \tilde{t}_i^e \tilde{t}_m^a \langle mb||ej\rangle) \\ & + P_-(ij) \sum_e \tilde{t}_i^e \langle ab||ej\rangle - P_-(ab) \sum_m \tilde{t}_m^a \langle mb||ij\rangle \Bigg\}. \end{aligned} \tag{16}$$

The intermediate 2-tensors above are defined as follows:

$$\begin{aligned} \mathcal{F}_{ae} = & -\frac{1}{2}\sum_m f_{me} \tilde{t}_m^a + \sum_{mf} g_{mafe}^{s} t_m^f \langle ma||fe\rangle \\ & - \frac{1}{2}\sum_{mnf} \hat{\tau}_{mn}^{af} \langle mn||ef\rangle, \end{aligned} \tag{17}$$

$$\mathcal{F}_{mi} = \frac{1}{2}\sum_e \tilde{t}_i^e f_{me} + \sum_{en} g_{mnie}^{s} t_n^e \langle mn||ie\rangle + \frac{1}{2}\sum_{nef} \hat{\tau}_{in}^{ef} \langle mn||ef\rangle, \tag{18}$$

$$\mathcal{F}_{me} = f_{me} + \sum_{nf} g_{mnef}^{s} t_n^f \langle mn||ef\rangle, \tag{19}$$

while the intermediate 4-tensors are defined as

$$W_{mnij} = \langle mn||ij\rangle + P_-(ij) \sum_e \tilde{t}_j^e \langle mn||ie\rangle + \frac{1}{4}\sum_{ef} \tau_{ij}^{ef} \langle mn||ef\rangle, \tag{20}$$

$$\begin{aligned} W_{abef} = & \langle ab||ef\rangle - P_-(ab) \sum_m \tilde{t}_m^b \langle am||ef\rangle \\ & + \frac{1}{4}\sum_{mn} \tau_{mn}^{ab} \langle mn||ef\rangle, \end{aligned} \tag{21}$$

$$\begin{aligned} W_{mbej} = & \langle mb||ej\rangle + \sum_f \tilde{t}_j^f \langle mb||ef\rangle - \sum_n \tilde{t}_n^b \langle mn||ej\rangle \\ & - \sum_{nf} \ddot{\tau}_{jn}^{fb} \langle mn||ef\rangle, \end{aligned} \tag{22}$$

$$\hat{\tau}_{ij}^{ab} = \tilde{t}_{ij}^{ab} + \frac{1}{2} g_{ijab}^{s} (t_i^a t_j^b - t_i^b t_j^a), \tag{23}$$

$$\tau_{ij}^{ab} = \tilde{t}_{ij}^{ab} + g_{ijab}^{s} (t_i^a t_j^b - t_i^b t_j^a), \tag{24}$$

$$\ddot{\tau}_{ij}^{ab} = \frac{1}{2}\tilde{t}_{ij}^{ab} + g_{ijab}^{s} t_i^a t_j^b. \tag{25}$$

Here, we have used the standard permutation operators,

$$P_{\pm}(pq) = 1 \pm \mathcal{P}(pq), \tag{26}$$

where $\mathcal{P}(p,q)$ permutes the indices p and q. Lastly, we have used D to denote the diagonal Fock matrix energy differences:

$$D_i^a = f_{ii} - f_{aa}, \tag{27}$$

$$D_{ij}^{ab} = f_{ii} + f_{jj} - f_{aa} - f_{bb}. \tag{28}$$

E. Solving the smoothly bumped LCCSD equations

1. Strong and moderate amplitudes

The equations above are identical to the standard CCSD equations if one sets $g^s_{ij}=g^m_{ij}=1\,\forall\, i,j$. However, because the t amplitudes always appear in bumped form $\tilde{t}^{ab}_{ij}$ on the right hand side of the iterative equations, all matrix multiplication touches only those t^{ab}_{ij} for which $g^s_{ijab}>0$. Hence, only the strong t amplitudes must be iteratively calculated. Furthermore, because we have inserted nontrivial bump functions (either g^s or g^m) around all product terms on the right hand side of Eqs. (15)–(25), only the strong and moderate t amplitudes have nontrivial answers. The moderate t amplitudes can be calculated with one iteration at the end of the calculation, so they are essentially free.

2. Weak and negligible amplitudes

Now, we recall that the weak amplitudes are defined as those t^{ab}_{ij} for which $g^s_{ijab}=g^m_{ijab}=0$. These amplitudes need not be calculated at all, for they have the unique solution

$$(t^{(\mathrm{weak})})^{ab}_{ij}=\frac{\langle \eta_i\eta_j||\eta_a\eta_b\rangle}{f_{ii}+f_{jj}-f_{aa}-f_{bb}}. \tag{29}$$

This is the Kapuy formula for the most basic second-order perturbative correction to the t amplitudes.[38–44] These amplitudes make independent contributions to the correlation energy and can be added quickly to form the KMP2 energy [see Eq. (35)].

Finally, we have defined all weak amplitudes t^{ab}_{ij} for which $|\langle \eta_i\eta_j||\eta_a\eta_b\rangle|<10^{-16}$ as negligibly interacting amplitudes. According to Eq. (29) above, it is clear that these amplitudes can be fully ignored with only slight numerical error on the order of machine precision. In this way, our LCCSD equations enforce locality and sparsity on the iterative CCSD equations.

F. The implicit function theorem

The smoothness of our LCCSD energy is guaranteed by the implicit function theorem of differential geometry. According to that theorem, so long as the first-derivative matrix of the coupled cluster equations (with respect to t amplitudes) is smooth and invertible, then the t amplitudes (and correlation energy) will be smooth functions of nuclear coordinates. For this reason, no bump function should ever be applied to the left hand side of the iterative LCCSD equations. Aside from that, one may apply bump functions liberally to transform nonlocal equations into local equations, while retaining differentiability of the solution. However, bump functions do distort the true, physical iterative amplitude equations, and thus, should be used sparingly and only where absolutely necessary.

III. ALGORITHMIC DETAILS AND IMPLEMENTATION

Our local-correlation energy is defined by the iterative solution to the bumped LCCSD equations above. Below we will outline how we have solved those equations in a near linear-scaling fashion; but first, we strenuously emphasize that our locally correlated wave functions and correlation energies are completely independent of our method of solution. This invariance is the strength of our approach, and is necessary for our locally correlated energies to be smooth functions of nuclear coordinates.

A. Orbital blocking

For standard matrix multiplication, blocking of matrices is essential in order to best utilize cache memory and maximize performance. There is a natural trade-off, however, when applying blocking to systems with sparsity. On the one hand, if a block is large and full, matrix multiplication is much faster. On the other hand, if a block is large and has only one nonzero element, many floating-point operations (FLOPS) will be wasted.

In our algorithm, at the beginning of each calculation, we pick a fixed number of blocks for the occupied and virtual, alpha and beta spaces. This number is chosen so that the average number of orbitals per block is 4 or 5, which has proven to be a good compromise between cache utility and sparsity. Closeby orbitals are blocked together using simulated annealing as follows.

Let $M_{o/v,\alpha/\beta}$ be the number of occupied/virtual alpha/beta blocks of molecular orbitals. Let $M=M_{o,\alpha}+M_{o,\beta}+M_{v,\alpha}+M_{v,\beta}$ be the total number of blocks. We denote the set of all blocks by $(A_1,\ldots,A_M)$, where we order them, say, by $\{o\alpha,o\beta,v\alpha,v\beta\}$. We perform a simulated annealing calculation, maximizing the energy function

$$E(A_1,\ldots,A_M)=\sum_{i,j=1}^{M}|r_i-r_j|^2|A_i||A_j|, \tag{30}$$

where r_i is the average position in three-dimensional space $(\mathbb{R}^3)$ of the block of orbitals A_i, and $|A_i|$ is the number of orbitals in the block A_i. We have found this function to be a quick and efficient way to block closeby orbitals together.

B. Integrals

By far, the most numerous 4-tensors to be computed and manipulated are the two-electron integrals, $\langle pq\|rs\rangle$. We have chosen to compute these integrals according to the resolution of identity (RI) method,[45] fitting pairs of molecular orbitals (pq) to fitting functions K: $(pq|K)$. This initial step is fast. Thereafter, the matrix elements are contracted to form the two-electron integrals: $(pq|rs)=\Sigma_K(pq|K)(K|rs)$, which is formally an expensive step.

But, of course, not all integrals are necessary. The early papers of Werner and co-workers (e.g., Ref. 21) describe clearly how only a linear number of integrals are necessary in a local algorithm. Similarly, in our very different algorithm, by applying bump functions to the t amplitudes on the right hand side of the iterative CCSD equations, one can show that almost always, only a linear number of integrals are neccesary in the iterative calculation. Let us consider, for example, the all-virtual two-electron integrals, $\langle ab\|ef\rangle$, which are typically the biggest block of integrals necessary in a CCSD calculation.

The integral $\langle ab\|ef\rangle$ enters the iterative equations above only through the term

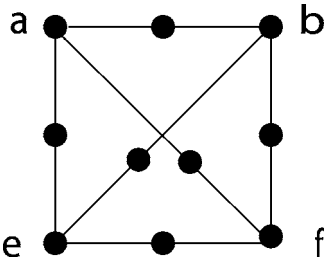

FIG. 2. A schematic diagram of all necessary two-electron virtual-only integrals. Here, a dot represents any molecular orbital (occupied or virtual), and a line connecting orbitals x and y means that $g_{xy}>0$. Originates from term $D_{ij}^{ab}t_{ij}^{ab}=\Sigma_{ef}\langle ab||ef\rangle t_{ij}^{ef}g_{ijef}$. Similar diagrams can be drawn for the other sets of necessary two-electron integrals.

$$D_{ij}^{ab}t_{ij}^{ab}+=g_{ijab}^{s}\frac{1}{2}\sum_{ef}\tau_{ij}^{ef}\langle ab||ef\rangle, \tag{31}$$

where we repeat the definition above:

$$\tau_{ij}^{ef}=g_{ijef}^{s}t_{ij}^{ef}+g_{ijef}^{s}(t_{i}^{e}t_{j}^{f}-t_{i}^{f}t_{j}^{e}). \tag{32}$$

Because $g_{ijab}^{s}>0$ and $g_{ijef}^{s}>0$ for any nontrivial iteration, there are thus clear guidelines for which we must compute $\langle ab||ef\rangle$. After all, the orbitals ij must "connect" the orbitals ab and ef. Pictorially, if we represent an orbital p by a black dot and then we draw a line between all orbitals (pq) for which $g_{pq}^{s}>0$, we show diagrammatically in Fig. 2 how few all-virtual integrals must be calculated. Similar diagrams can be drawn for the other sets of four-electron integrals.

As mentioned previously, however, there are four specific contractions for which there are more than a linear number of nonzero terms, and thus one must explicitly bump the two-electron integrals in order to enforce only a linearity of terms. These terms occur in Eqs. (15) and (17)–(19) and always arise from contractions of the form

$$D_{q}^{p}t_{q}^{p}+=\sum_{rs}\langle pr||qs\rangle t_{s}^{r}. \tag{33}$$

Here, even if we replace t_r^s by $\tilde{t}_r^s$ on the right hand side of the equation, the sparsity of $\tilde{t}$ is still not enough to limit the size of the contraction. Thus, we must explicitly bump the two-electron integrals in order to achiever linear scaling. In fact, whenever the naked, unbumped t amplitude appears on the right hand side of Eqs. (15)–(19), it is always because we have already bumped the relevant integral (and, thus, implicitly the t amplitude).

Ignoring these four exceptions, one must never include more than three degrees of separation between any two orbitals in a relevant two-electron integral. Furthermore, one is suprised to learn that there are actually more $\langle ia||bc\rangle$ integrals to be calculated and stored than there are $\langle ab||ef\rangle$ integrals according to our LCCSD algorithm. Overall, it is clear why only a linear, albeit a large linear, number of integrals are required by our algorithm.

C. Disk I/O storage

The disk requirements of our algorithm are dominated by the need to compute and store the 4-tensors, most importantly the two-electron integrals, that arise in any calculation. Though this number becomes linear in the limit of a large system, it is crucial to store these large tensors in an effectively sparse fashion.

In our implementation, all 2-tensors are kept in core throughout the calculation. 4-tensors are separated into chunks of memory and written individually to disk. Just as for blocks of orbitals, chunks of blocks of orbitals should be as local as possible. Hence, again, one groups the localized blocks of orbitals together into large chunks of memory using simulated annealing. The same energy function from above is used, where now C stands for chunks of blocks of orbitals, P is for the number of chunks, and r_i is the average position of chunk C_i:

$$E(C_1,\ldots,C_P)=\sum_{i,j=1}^{P}|r_i-r_j|^2|C_i||C_j|. \tag{34}$$

One point should now be made. The optimal number of chunks is very unclear. Certainly, if all data can be kept in core, then the code is optimal. However, for large systems, smaller chunk sizes may be better than larger chunk sizes when seek times are not the bottleneck, and where asynchronous input-output ports (I/O) can save an order of magnitude in the wall time. We have not experimented with this parameter, and as such, our I/O time is certainly larger than it should be. We have thus far aimed only for the fewest number of chunks. Future work will focus on optimizing the I/O bottleneck.

D. Data structures in memory

Our algorithm has been designed to need as little memory as possible. By breaking up the 4-tensors into chunks that are written to disk, we do not require large chunks of memory. Formally, our amplitude iterations require only a quadratic amount of memory—though, of couse, more memory translates into fewer seeks and less I/O time.

When in memory, chunks of both 2-tensors and 4-tensors are stored sparsely using a blocked compressed row-storage (BCRS) scheme.[46] For the 2-tensors, this sparse storage scheme is common in scientific computing circles. One requires only two integer arrays *row_ptr* and *col_ind*, and one double array *val*, which contains the actual data.

For 4-tensors, the same exact storage scheme works, provided that one regards a 4-tensor as a glorified 2-tensor. In other words, one may regard the 4-tensor E_{i_1,i_2,i_3,i_4}, where $i_2<N_2$ and $i_4<N_4$ as a simple 2-tensor $\tilde{E}_{i_1N_2+i_2,i_3N_4+i_4}$. The BCRS scheme is then applicable, at a cost of not very much extra overhead.

E. A sketch of our algorithm

For completeness, we now sketch how our complete algorithm works.

(1) Do a Hartree-Fock calculation. Subsequently, localize all occupied and virtual orbitals (η_i,η_a).
(2) Solve for the weakly interacting t amplitudes by computing the KMP2 sum:

$$E_{\mathrm{KMP2}} = \sum_{ijab} \frac{\langle \eta_i \eta_j || \eta_a \eta_b \rangle}{f_{ii} + f_{jj} - f_{aa} - f_{bb}}, \tag{35}$$

see Sec. II E 2. Though this needs to be done only for weakly interacting t amplitudes (i.e., one may ignore negligibly interacting t amplitudes), we have not yet screened our orbital quartets and, thus, we have not yet efficiently implemented this summation.

(3) Perform a simulated annealing run to best group pairs of closeby orbitals into blocks. Perform a second simulated annealing run to best group nearby blocks of orbitals into chunks, which will be stored contiguously on disk.
(4) Calculate distances between orbitals and construct relevant 2- and 4-tensors after cutting off unnecessary pairs and quartets.
(5) Iterate the LCCSD equations until self-consistent. This requires contracting and updating only the strong t amplitudes until they are converged.
(6) Use the converged values for the strong t amplitudes to compute the moderate t amplitudes. According to Eq. (16), the moderate t amplitudes are computed just as normal MP2 amplitudes, except that they feel effective forces only from the strong t amplitudes, and hence can be solved in one iteration (i.e., there is no self-consistency condition to be attained).
(7) Using the converged values for the strong and moderate t amplitudes, compute the final LCCSD energy.

IV. NUMERICAL RESULTS

The algorithm described above has been implemented and interfaced with the Q-CHEM quantum chemistry package.[47] All correlated calculations were carried out on top of an unrestricted Hartree-Fock (UHF) ground state. For restricted Hartree-Fock (RHF) calculations, our algorithm would be faster and the solution would still be properly spin adapted (just like regular CCSD).

A. Accuracy

In order to demonstrate the accuracy of our LCCSD method, we provide LCCSD and CCSD energies in Table I for carbon chains of length 10 (in a cc-pVDZ basis). We do this for the alkane, the polyene, and the poly-yne. In all three cases, note that we attain over 99% of the correlation energy, suggesting that our algorithm is not very sensitive to the gap between the highest occupied molecular orbital (HOMO) and lowest unoccupied molecular orbital (LUMO).

B. Scaling

In order to demonstrate the scaling of our LCCSD method, we have computed LCCSD energies for alkanes of increasing size in a cc-pVDZ basis. Results are in Tables I–III.

1. Disk

In Table II, we list the number of T_2 amplitudes and the number of $(ia|bc)$ integrals that must be stored on disk as a function of alkane size. The number of *exact* T_2 amplitudes is the number of amplitudes for which the corresponding strong bump function is nonzero and which will contribute to the LCCSD calculation; i.e., this is the number of t_{ij}^{ab} for which $g_{ijab}^{s} > 0$. The number of *effective* T_2 amplitudes is the number of T_2 amplitudes we must carry through our calculation. We must necessarily carry more amplitudes than exactly necessary because we block our orbitals (which, in turn, is done for the computational savings gained while doing matrix multiplication in cache). Thus, if i is blocked with i' and if $g_{ijab}^{s} > 0$, then we must store on disk both t_{ij}^{ab} and $t_{i'j}^{ab}$, even if $g_{i'jab}^{s} = 0$.

In practice, we are usually required to carry between two and three times more T_2 amplitudes than what is exactly required: $N_{\mathrm{eff}}/N_{\mathrm{exact}} \approx 2.5$. Of course, the effective number of amplitudes defines the speed of our algorithm, and thus the effective number needs to be minimized for the greatest speed up. Our desire to limit the number of effective amplitudes leads us to perform a simulated annealing experiment at the beginning of a LCCSD calculation, as discussed earlier.

From the data in Table II, one can get a sense of just how much disk space is required for large LCCSD calculations. Most importantly, we note that the number of amplitudes and integrals becomes linear as the size of the molecule increases. This allows our algorithm to run in a near-linear amount of time for large systems.

TABLE I. Energetics of our LCCSD algorithm for growing alkane chains, one polyene, and one poly-yne. Note that our LCCSD energies are close to linear functions of alkanes. This is the correct answer, since alkane chemistry is size extensive. Furthermore, note that our LCCSD energy is similarly accurate for alkanes, polyenes, and poly-ynes, suggesting that our local algorithm is not very sensitive to the HOMO-LUMO gap.

Type	n	LCCSD energy (hartree)	Full CCSD energy	Percent full CCSD energy (%)
Alkane	10	−1.608 57	−1.615 97	99.5
	20	−3.188 10	N/A	N/A
	30	−4.767 63	N/A	N/A
	40	−6.347 15	N/A	N/A
Alkene	10	−1.428 87	−1.435 68	99.5
Alkyne	10	−1.265 81	−1.270 71	99.6

2. Timings and Memory

In Table III, we list a number of relevant times as a function of alkane size for our sample calculations. We ran all of our calculations on a 2 GHz IBM PowerPC 970 (G5) processor using 8 Gbytes of random access memory (RAM).

In the fourth and third columns of Table III, we list the total time required for our LCCSD iterations and the total contraction time. The iteration time includes the contraction time; their difference is essentially permutation time and overhead time for building sparse matrices. This difference will be minimized in future implementations. One can see that our algorithm begins to scale nearly linearly over the

TABLE II. Sparsity of our LCCSD algorithm for growing alkane chains. Note that the number of T_2 amplitudes approaches a linear number asymptotically as the system size grows. Because the two-electron integrals dominate the amount of disk space required by our algorithm, this table can be used to estimate the disk requirements of our algorithm.

n	Size of basis	Exact number of T_2 amplitudes	Effective number of T_2 amplitudes	Effective percent of T_2 amplitudes %	Effective number of $(ia\|bc)$ needed	Effective percent of $(ia\|bc)$ needed (%)
10	250	7 853 919	15 156 501	7	813 915 014	54
20	490	16 948 935	30 424 054	0.9	2 281 289 463	10
30	730	26 051 499	43 828 015	0.3	3 711 518 998	3
40	970	35 151 159	52 981 261	0.1	5 084 956 005	1

regime of less than 500 basis functions. In the asymptotic limit, the timings for iterations will certainly be linear. For comparison, we note that the time for a full CCSD for decane is 134 089 s, while the total time for a LCCSD calculation is 126 188 s. The speedup in wall time for LCCSD over full CCSD will be remarkable for alkanes bigger than $C_{10}H_{22}$. Specifically, it will be approximately a factor of 20 faster for the C20 chain, and over 1000 times faster for the C40 chain. Of course, one-dimensional systems are the most favorable case, because the speedup depends critically on the number of strong amplitudes that are coupled to a given amplitude—a quantity that increases strongly with dimensionality.

In the interest of completeness, in the sixth and fifth columns of Table III, we list the wall time for generating the necessary two-electron integrals in the iterative CCSD equations and the wall time needed for computing the perturbative term $\Sigma_{ijab}\langle \eta_i \eta_j || \eta_a \eta_b \rangle / (\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b)$, which is formally non-linear without any local approximations. These terms are not rate limiting in the above calculations (with fewer than 970 basis functions), as they are computed only once. As such, we have not yet optimized our algorithm to compute them in the fastest, best-scaling method.

For larger molecules, however, our data suggest that these terms will become very expensive. Indeed, in the final column of Table III, we report the total wall time for the calculation (including the time for the Hartree-Fock calculation). From these data, it is clear that our code will not be linear scaling until we implement efficient code for treating the perturbative and integral subroutines (which will be done in the future). In fact, in the case of $C_{80}H_{162}$, our naive implementation is not yet capable of generating the necessary integrals and perturbative terms in a reasonable amount of time. Still, our implementation thus far has focused only on minimizing the most difficult part, the time for the CC iterations; in the future, we believe we can implement a fast-scaling code for the perturbative and integral subroutines. This should, in turn, allow LCCSD calculations on molecules with far more than 1000 basis functions.

C. Smoothness

In order to demonstrate the smoothness of our LCCSD algorithm, we present in Fig. 3(a) the potential energy surfaces for the homolytic dissociation of ethane (CH_3CH_3) along the C–C bond; and in Fig. 4(a), the same is presented for the heterolytic dissociation of ketene (H_2CCO) also along the C–C bond. These were two model examples investigated previously by Russ and Crawford,[33] the only difference being that their calculations were restricted and ours are unrestricted.

In both cases, we show the full CCSD curve, the smoothly bumped LCCSD curve ($c_0^s = 35$ Å^2 and $c_1^s = 30$ Å^2), and the unsmoothed, nonbumped LCCSD curve ($c_0^s = c_1^s = 35$ Å^2). One can immediately see that our LCCSD algorithm closely approximates the full CCSD energy. Further-

TABLE III. Scaling of computational cost for our LCCSD algorithm for growing alkane chains. See text for details. So far, we have employed sparsity only for the iterative piece of the LCCSD calculation, which is nearly linear (or, certainly subquadratic) when the alkane chain is already large. The total wall time is not linear scaling, because the Hartree-Fock, perturbative, and integral timings are not linear; their code has not yet been optimized using any sparsity. For a comparison of time saved, we note that the total wall time for the full CCSD calculations is 134 089 s for $C_{10}H_{22}$, and this full CCSD timing scales as the sixth power.

n	Number of iterations until convergence	LCCSD contraction wall time (s)	LCCSD interation wall time (s)	Perturbation wall time (s)	Build $(pq\|rs)$ wall time (s)	Total wall time (s)
10	27	69 841	119 186	118	2 916	126 188
20	28	200 737	369 280	2 358	15 510	400 947
30	28	302 678	554 015	14 061	37 125	639 922
40	28	436 355	793 365	53 425	75 799	966 073

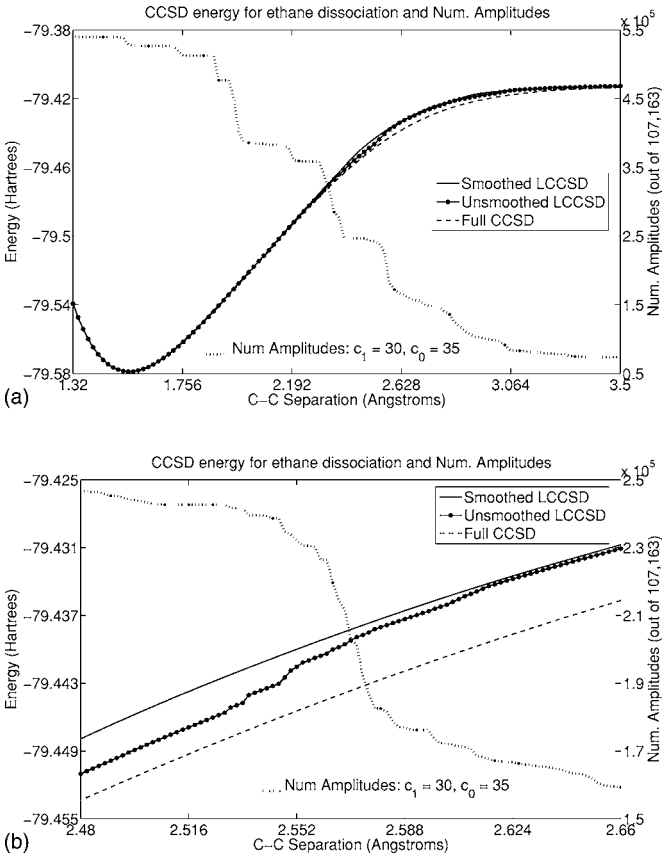

FIG. 3. (a) Here we show the CCSD and LCCSD (smoothed and unsmoothed) potential energy curves for ethane dissociation, when the C–C distance is between 1.3 and 3.5 Å. We use parameters $c_0=35$ Å^2 and $c_1=30$ Å^2. In general, the error $|\text{LCCSD}-\text{CCSD}|<1$ mhartree, but in the dissociation range (2.4–3 Å), we have $|\text{LCCSD}-\text{CCSD}|\approx 5-6$ mhartree. The calculation is unrestricted in a cc-pVDZ basis. (b) A blow-up of (a) when the C–C distance is between 2.48 and 2.66 Å. Note that the discontinuities of the unsmoothed curve are replaced by a fully differentiable smoothed curve. The calculation is unrestricted in a cc-pVDZ basis.

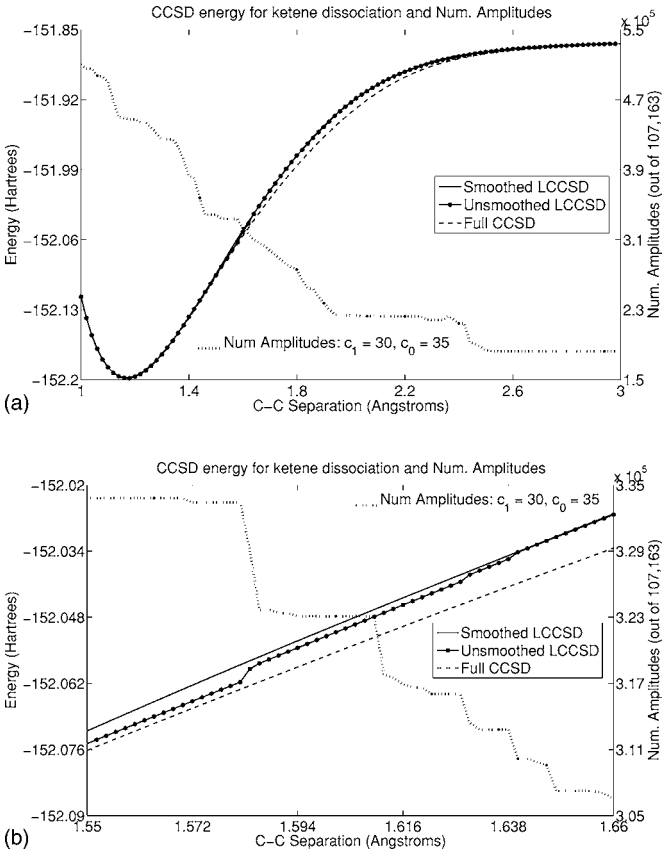

FIG. 4. (a) Here we show the CCSD and LCCSD (smoothed and unsmoothed) potential energy curves for ketene dissociation, when the C–C distance is between 1 and 3 Å. We use parameters $c_0=30$ Å^2 and $c_1=35$ Å^2. In general, the error $|\text{LCCSD}-\text{CCSD}|<1$ mhartree, but in the dissociation range (1.8–2.2 Å), we have $|\text{LCCSD}-\text{CCSD}|\approx 10$ mhartree. The calculation is unrestricted in a cc-pVDZ basis. (b) A blow-up of (a) when the C–C distance is between 1.55 and 1.66 Å. Note that the discontinuities of the unsmoothed curve are replaced by a fully differentiable smoothed curve. The calculation is unrestricted in a cc-pVDZ basis.

more, and not surprisingly, the unsmoothed curve is a better approximation than the smoothed curve, i.e., we sacrifice some accuracy for differentiability.

In Figs. 3(b) and 4(b), we zoom in on the curves from Figs. 3(a) and 4(a) over smaller domains with more resolution. Here, the nonbumped curves show the discontinuities we had anticipated in their potential energy surface. The figures show, however, that in both cases our smoothed LCCSD algorithm does fix this problem, producing fully differentiable potential energy curves with our choice of parameters.

V. CHEMICAL EXAMPLE

As an application of our LCCSD method, we have computed the energetic gap between an extended and a globular conformation of the alanine tetrapeptide. Basic hydrogen bonding provides intuition that the globular structure should be more energetically stable than the extended structure. Still, a quantitative estimate for the relative stabilities of these different conformations is an important prototype for understanding the energetics of protein folding; furthermore, the energetic gap between these two conformations should be reproducible by the force fields used in molecular mechanics calculations. Building on the pioneering local MP2 calculations of Beachy *et al.*,[48] DiStasio *et al.* have recently done extensive calculations on the energetics of these conformations at the RI-MP2 level of correlation, and they report an extrapolated cc-pVTZ/cc-pVQZ energetic gap of 4.8 kcal/mol.[49,50]

We have now done a very simple calculation to investigate how this energetic gap might be affected by higher level correlations. Using the (RI-MP2/cc-pVDZ)-optimized geometry of DiStasio *et al.*, we calculated the RI-MP2, RI-LMP2, and LCCSD energies of these two tetrapeptide conformations in a cc-pVDZ basis. As a double check of the accuracy of our method, we have also calculated the same quantities for two different dipeptide geometries, where the molecule is small enough that we can also calculate full CCSD energies. All data are given in Tables IV and V.

One can see from Table IV that we do expect our method to be accurate enough to make energetic comparisons—though the very high accuracy in this case is likely by chance or a consequence of the relatively small size of these systems. From Table V, however, we find the energetic gap between globular and extended conformations of alanine tetrapeptide to be 8.17 kcal/mol at the RI-MP2 level and 2.81 kcal/mol at the RI-LMP2 level. This is not inconsistent

TABLE IV. Comparison of RI-MP2, RI-LMP2, CCSD, and LCCSD energies for extended and globular conformations of alanine dipeptide. In this case, the local approximation is spectacular—certainly far better than can be expected in general. The error introduced by the local approximation is less than 0.30 kcal/mol at both MP2 and CCSD levels of correlation.

Method	Extended geometry hartree	Globular geometry hartree	Difference (globular−extended) hartree	Difference (globular−extended) (kcal/mol)
RI-MP2	−494.355 302 7	−494.437 728 3	−0.0824	−51.72
RI-LMP2	−494.335 873 0	−494.418 727 2	−0.0829	−51.99
CCSD	−494.425 2021	−494.513 198 8	−0.0880	−55.22
LCCSD	−494.417 830 6	−494.505 707 6	−0.0879	−55.14

with the differences seen between local MP2 (Ref. 48) and full MP2 (Refs. 49 and 50) with basis sets of this size. This error is introduced by the local approximation, and arises partly from the neglect of physically significant terms, and partly from the neglect of physically insignificant terms associated with basis set superposition error (BSSE). In the future, we must make sure to recheck that our choice of parameters is sufficiently robust. For the present, we note that the energy difference is 3.51 kcal/mol at the local LCCSD level of correlation, which is very close to the local MP2 (LMP2) result. To ensure that the beyond-MP2 correction is computed in a balanced manner that allows errors due to local modeling to cancel, the beyond-MP2 correction should be defined as the difference between LMP2 and LCCSD. If one then makes the (fairly reasonable) presumption that beyond-MP2 correlation effects are reasonably described at the cc-pVDZ level, this suggests that such effects increase the energy gap between extended and globular conformations by roughly 0.70 kcal/mol, yielding a corrected result of 5.50 kcal/mol.

In the future, further calculations must be done using a larger basis in order to verify the significance of this preliminary result. Still, because the LCCSD and LMP2 energies match up so well, our preliminary data suggest that the contribution of coupled-cluster doubles is not substantial in understanding the relative energies of different tetrapeptide configurations. At the same time, our data validate the use of perturbation theory as a reliable tool for calculating the properties of small chains of amino acids. Given how much faster MP2 is than CCSD and the enormous sizes of proteins, it is encouraging that, in at least one relevant test case, MP2 achieves enough accuracy for a small biological system that one can virtually ignore the need for higher-order corrections. Another interesting future application of LCCSD, however, might be the stacking of DNA bases, a chemical example where MP2 is known to considerably overstimate the stacking energy.

VI. DISCUSSION

Two conclusions can be drawn immediately from the data we have presented in this paper. First, this paper confirms the results of Werner *et al.*, demonstrating that local-correlation theory is remarkably good at capturing the correlation of electrons in large alkanes with nondegenerate ground states. After all, we have been able to capture more than 99% of the correlation energy simply by explicitly correlating electrons no father than 3.13 Å apart, (i.e., quartets of *ijab* orbitals in spheres of radius 1.57 Å). More distant electrons are coupled either perturbatively (in the KMP2 term) or indirectly (through the interaction of two strong couplings[51]). This bodes very well for future two- and three-dimensional systems. Second, this paper shows definitively the success of our algorithm both at handling large molecules at the level of LCCSD correlation theory with near linear scaling, and at constructing smooth potential energy surfaces. These were our two initial goals at the outset of this LCCSD project.

Regarding the theoretical advances behind this paper, on the one hand, one can argue that our approach of applying bump functions is basic and not very noteworthy. After all, analogs of bump functions have been used for years in molecular mechanics, where potentials are tapered off to zero. On the other hand, however, we would respond that finding smooth local-correlation energies has eluded many chemists for a long time, and led to the development of entirely new local models.[50,52–54] Until now, it was not obvious how to rigorously smooth out local amplitude equations, when the

TABLE V. Comparison of RI-MP2, RI-LMP2, and LCCSD energies for extended and globular conformations of alanine tetrapeptide. In this case, the local approximation introduces a significant error (≈5.4 kcal/mol) at the MP2 level. We then approximate that $\Delta_{CCSD}=\Delta_{LCCSD}+(\Delta_{RI\text{-}MP2}-\Delta_{RI\text{-}LMP2})$. See text for a more detailed discussion.

Method	Extended geometry (hartree)	Globular geometry (hartree)	Difference (globular−extended) (hartree)	Difference (globular−extended) (kcal/mol)
RI-MP2	−987.730 945 2	−987.743 971 7	−0.013	−8.17
RI-LMP2	−987.689 713 3	−987.694 184 3	−0.0045	−2.81
LCCSD	−987.855 372 4	−987.860 960 5	−0.0056	−3.51

unknown variables, rather than known variables (e.g., integrals), must be cut. The application of the implicit function theorem to guide us was an important step. Second, we regard the application of molecular mechanics packages to large proteins as a major achievement for theoretical chemistry. So, even if our solution to the local correlation problem is not terribly elegant, we will be very happy if our algorithm allows one to rigorously explore accurate potential energy surfaces from first principles for even larger molecules.

From a development perspective, in the future, our most important goal is to choose the best criteria to bump pairs and quadruplets of orbitals. In this paper, we bumped pairs of orbitals by distances between centers because this was the easiest criteria to implement efficiently. However, as we have argued previously,[1] bumping pairs of orbitals by two-electron integrals is far more physically meaningful and should be investigated. The reasons are twofold. First, two-electron integrals take location *and* shape of orbitals into account. Second, we expect the KMP2 guess for t_{ij}^{ab} [Eq. (29)] to be a reasonable approximation of how important t_{ij}^{ab} is. Thus, ideally, we would choose to bump t_{ij}^{ab} only when the value $\langle \eta_i \eta_j || \eta_a \eta_b \rangle / (\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b)$ is above a certain threshold. And, if we use the Cauchy-Schwarz theorem, we can certainly bound the integral $(ia|jb)$ by products of two-electron integrals $[(ii|jj),(ia|ia),(jb|jb),(aa|bb)]$. This justifies bumping pairs of orbitals by two-electron integrals.

We note that bumping the quadruplet of orbitals $(ijab)$ directly by the KMP2 term would be hard to implement efficiently because the long decay of the Coulomb tail of $(ia|jb)$ means effectively a quadratic number of significant terms; we aspire to a linear number of significant terms. However, after bumping pairs of orbitals, we could bump quadruplets of orbitals by the KMP2 term in order to attain even more sparsity (in particular, from the energy denominators). This should also be investigated. We note that Auer and Nooijen[55] have recently published a very detailed paper where they did similar work: they screened for important t amplitudes according to their KMP2 term, and then computed iterative coupled-cluster values only for those important t amplitudes.

After finding and implementing the optimal bump criteria, we can anticipate four more future projects. First, some optimization of our algorithm in general remains to be done. We expect that a large speedup of our algorithm will be possible through better management of I/O and by exploiting asynchronous I/O; this has not yet been explored at all. Second, in the future, one must implement a program to calculate the gradient of the LCCSD energy, which should allow optimization of energies at a high level of electron correlation. Third, we may now explore response theory in a local orbital framework as a way of calculating correlated energies for excited states. Fourth, we should calculate the perturbative triples correction [CCSD(T)] if we want the most faithful correlation energy possible (in a reasonable period of time).

Many interesting chemical applications for LCCSD should now present themselves. The relative stabilities of different conformations of large molecules is crucial in chemistry. The tetrapeptide example presented in this paper shows how useful our LCCSD algorithm can be, if only to validate less accurate methods; and for open-shell systems, one is far less hopeful that a perturbative solution will be sufficiently accurate. Furthermore, barriers of chemical reactions and large molecule optimized geometries are also important, and these energies should be accessible by our approach in the future. Lastly, accurate correlation energies are crucial for almost all examples of bond making and bond breaking, so LCCSD should be quite useful in modeling such processes. In fact, it is hard to exaggerate the importance of fast and accurate, smooth local-correlation energy functions in theoretical chemistry.

VII. CONCLUSIONS

In summary, this paper shows that a fast, accurate, smooth local CCSD algorithm is achievable, using the theoretical model we have proposed over the past year. In the future, we expect this algorithm to find use in a wide range of chemistry applications, where accurate energetics are crucial. Furthermore, because the theory behind our method is so very simple, we expect, henceforward, that a variety of new local-correlation methods will evolve, each using the underlying theory we have developed here. One large aim for modern quantum chemistry is the construction of an optimally fast, smooth, accurate local-correlation energy function, and we believe that our LCCSD algorithm represents a useful step forward in this direction.

ACKNOWLEDGMENTS

The authors thank Anthony Dutoi and Greg Beran for many, many interesting discussions. They also thank Robert A. DiStasio, Jr. for providing them with dipeptide and tetrapeptide geometries. This article is dedicated to the memory of Jeremy K. Burdett. One of the authors (J.E.S.) was supported by the Fannie and John Hertz Foundation. This work was further supported by a grant from the Computational Nanoscience initiative of the U.S. Department of Energy.

[1] J. Subotnik and M. Head-Gordon, J. Chem. Phys. **123**, 064108 (2005).
[2] J. Cízek, J. Chem. Phys. **45**, 4256 (1966).
[3] J. Cízek, Adv. Chem. Phys. **14**, 35 (1969).
[4] J. Cízek and J. Paldus, Int. J. Quantum Chem. **5**, 359 (1971).
[5] T. Crawford and H. Schaefer III, Rev. Comput. Chem. **14**, 33 (2000).
[6] G. Purvis and R. Bartlett, J. Chem. Phys. **76**, 1910 (1982).
[7] K. Raghavachari, G. Trucks, and J. A. Pople, Chem. Phys. Lett. **157**, 479 (1989).
[8] P. Pulay, Chem. Phys. Lett. **100**, 151 (1983).
[9] S. Saebo and P. Pulay, Chem. Phys. Lett. **113**, 13 (1985).
[10] P. Pulay and S. Saebo, Theor. Chim. Acta **69**, 357 (1986).
[11] S. Saebo and P. Pulay, J. Chem. Phys. **86**, 914 (1987).
[12] S. Saebo and P. Pulay, Annu. Rev. Phys. Chem. **44**, 213 (1993).
[13] G. Reynolds, T. Martinez, and E. Carter, J. Chem. Phys. **105**, 6455 (1996).
[14] R. Murphy, M. Beachy, and R. Friesner, J. Chem. Phys. **103**, 1481 (1995).
[15] C. Hampel and H. Werner, J. Chem. Phys. **104**, 6286 (1996).
[16] A. E. Azhary, G. Rauhut, P. Pulay, and H. Werner, J. Chem. Phys. **108**, 5185 (1998).
[17] G. Rauhut, A. E. Azhary, F. Eckert, U. Schumann, and H. Werner, Spectrochim. Acta, Part A **55**, 647 (1999).
[18] M. Schütz, G. Hetzer, and H. Werner, J. Chem. Phys. **111**, 5691 (1999).
[19] M. Schütz and H. Werner, Chem. Phys. Lett. **318**, 370 (2000).
[20] M. Schütz, J. Chem. Phys. **113**, 9986 (2000).

[21] M. Schütz and H. Werner, J. Chem. Phys. **114**, 661 (2001).
[22] G. Rauhut and H. Werner, Phys. Chem. Chem. Phys. **3**, 4853 (2001).
[23] M. Schütz, J. Chem. Phys. **116**, 8772 (2002).
[24] G. Rauhut and H. Werner, Phys. Chem. Chem. Phys. **5**, 2001 (2003).
[25] H. Werner, F. Manby, and P. Knowles, J. Chem. Phys. **118**, 8149 (2003).
[26] M. Schütz, H. Werner, R. Lindh, and F. Manby, J. Chem. Phys. **121**, 737 (2004).
[27] T. Hrenar, G. Rauhut, and H. Werner, J. Phys. Chem. A **110**, 2060 (2006).
[28] G. Scuseria and P. Ayala, J. Chem. Phys. **111**, 8330 (1999).
[29] S. Li, J. Ma, and Y. Jiang, J. Comput. Chem. **23**, 237 (2002).
[30] N. Flocke and R. Bartlett, J. Chem. Phys. **121**, 10935 (2004).
[31] M. Schütz, Phys. Chem. Chem. Phys. **4**, 3941 (2002).
[32] M. Schütz, Phys. Chem. Chem. Phys. **5**, 3349 (2003).
[33] N. Russ and T. Crawford, J. Chem. Phys. **121**, (2004).
[34] J. Pople, in *Energy, Structure and Reactivity*, edited by D. Smith and W. McRae (Wiley, New York, 1973), p. 51.
[35] M. Schütz, G. Rauhut, and H. Werner, J. Phys. Chem. A **102**, 5997 (1998).
[36] J. Subotnik, A. Dutoi, and M. Head-Gordon, J. Chem. Phys. **123**, 114108 (2005).
[37] J. F. Stanton, J. Gauss, J. Watts, and R. Bartlett, J. Chem. Phys. **94**, 064334 (1991).
[38] E. Kapuy, Z. Csepes, and C. Kozmutza, Int. J. Quantum Chem. **23**, 981 (1983).
[39] E. Kapuy, F. Bartha, F. Bogar, Z. Csepes, and C. Kozmutza, Int. J. Quantum Chem. **38**, 139 (1990).
[40] E. Kapuy, F. Bogar, and C. Kozmutza, J. Mol. Struct. **297**, 365 (1993).
[41] E. Kapuy, F. Bogar, and E. Tfirst, Int. J. Quantum Chem. **52**, 127 (1994).
[42] C. Kozmutza, E. Kapuy, E. Evleth, and E. Kassab, J. Mol. Struct. **332**, 141 (1995).
[43] C. Kozmutza, E. Kapuy, E. Evleth, J. Pipek, and L. Trezl, Int. J. Quantum Chem. **57**, 775 (1996).
[44] J. Subotnik and M. Head-Gordon, J. Chem. Phys. **122**, 034109 (2005).
[45] G. F. M. Feyereisen and A. Komornicki, Chem. Phys. Lett. **208**, 359 (1993).
[46] J. Dongarra, in *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, edited by Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst (SIAM, Philadelphia, 2000), p. 315.
[47] J. Kong, M. S. Lee, A. M. Lee *et al.*, J. Comput. Chem. **21**, 1532 (2000).
[48] M. Beachy, D. Chasman, R. Murpy, T. Halgren, and R. Friesner, J. Am. Chem. Soc. **119**, 5908 (1997).
[49] R. DiStasio, Jr., R. Steele, Y. Rhee, Y. Shao, and M. Head-Gordon, J. Comput. Chem. (accepted).
[50] R. DiStasio, Jr., Y. Jung, and M. Head-Gordon, J. Chem. Theory Comput. **1**, 862 (2005).
[51] For instance, in the LCCSD equations, we note that the resolvent $t_{ij}^{ab}D_{ij}^{ab}$ has a component $\Sigma_{klcd}t_{ik}^{ac}\langle kc||ld\rangle t_{jl}^{bd}$. Hence, even if orbitals i and j are more than 3.35 Å apart, they will still feel each other indirectly through orbitals $klcd$.
[52] P. Maslen and M. Head-Gordon, Chem. Phys. Lett. **283**, 102 (1998).
[53] P. Maslen and M. Head-Gordon, J. Chem. Phys. **109**, 7093 (1998).
[54] M. Lee, P. Maslen, and M. Head-Gordon, J. Chem. Phys. **112**, 3592 (2000).
[55] A. A. Auer and M. Nooijen, J. Chem. Phys. **125**, 024104 (2006).