J. Phys. Chem. A **2005,** *109,* 11629−11641

A General Nonlinear Expansion Form for Electronic Wave Functions[†]

Ron Shepard[‡]

Chemistry Division, Argonne National Laboratory, Argonne, Illinois 60439

Received: August 4, 2005

A new expansion form is presented for electronic wave functions. The wave function is a linear combination of product basis functions, and each product basis function in turn is formally equivalent to a linear combination of configuration state functions that comprise an underlying linear expansion space. The expansion coefficients that define the basis functions are nonlinear functions of a smaller number of variables. The expansion form is appropriate for both ground and excited states and to both closed and open shell molecules. The method is formulated in terms of spin-eigenfunctions using the graphical unitary group approach (GUGA), and consequently it does not suffer from spin contamination.

1. Introduction

The graphical unitary group approch (GUGA) of Shavitt[1−7] has been used to compute electronic wave functions for multiconfiguration self-consistent-field (MCSCF)[8,9] and configuration interaction[4,10] (CI) expansions. It forms the underlying basis of the COLUMBUS Program System,[10−12] of which the main emphasis is the accurate computation of global potential energy surfaces of ground and excited states. The graphical representation of the linear expansion space results in an intuitive approach, in efficient matrix element evaluation, in flexible wave function expansion spaces, and in efficient wave function optimization procedures. The entire procedure is based on expansions in terms of spin-adapted configuration state functions (CSFs), and consequently, the resulting wave functions do not suffer from spin contamination, and the optimization process is not plagued with artificial spin instabilities. The main disadvantage of these approaches is that large wave function expansions result in computationally demanding optimization procedures with large storage requirements. In particular, MCSCF expansions are limited typically to about 15 valence orbitals and benchmark full-CI expansions are limited to about 25 orbitals; the largest MR-SDCI expansions that can be computed are now about 10^9 CSFs, and the largest full-CI expansions are now about 10^{10} CSFs, the limits resulting from the complications associated with the storage and manipulation of such large data sets. It would be of great benefit if those practical limitations could be extended.

In the new approach presented herein, we attempt to keep all of the advantages of the GUGA approach while eliminating both the large data sets and the large computational effort of the current methods. Because the final wave function is represented as a linear combination of *product basis functions*, both ground and excited electronic states may be computed, and the Ritz variational bounds apply to all of the computed eigenvalues. The storage of the large sets of CSF expansion coefficients is eliminated and replaced instead with the storage of a much smaller number of variational parameters. The computational effort that scales as the CSF expansion length is eliminated and replaced with effort that scales only as the smaller number of variational parameters.

2. Method

We first summarize some of the relevant details of the GUGA method. In the unitary group approach the CSF expansion terms are represented as a sequence of integer triples (a_n, b_n, c_n), which form the rows of a Paldus array.[13−15] These integers are related to the number of orbitals n, the number of electrons N_n, and the spin quantum number S_n of that Paldus array row according to

$$N_n = 2a_n + b_n$$
$$S_n = b_n/2$$
$$n = a_n + b_n + c_n \tag{1}$$

In the GUGA approach, each integer triple corresponds to a *node* (or *vertex*, or *distinct row*) on a *Shavitt graph*. Each node thereby corresponds to an $\hat{S}^2$ spin-eigenfunction with eigenvalue $S_n(S_n + 1)$ and to a specific number of electrons N_n. The individual orbitals correspond to vertical levels in the graph. The Shavitt graph is a directed graph with a single *tail* (source) node located at a fictitious level 0 corresponding to the physical vacuum, and a single *head* (sink) at the highest level corresponding to the N and S of interest. The nodes at one level are connected with *arcs* (or *edges* or *steps*) to the nodes at the adjacent levels. There are four possible step numbers that connect the nodes, denoted by the integer $d = 0, ..., 3$. The changes of the various quantities associated with each of these steps are summarized in Table 1. Each node in the Shavitt graph is connected to between one and four nodes at the next higher level, and to one to four nodes at the next lower level (except for the tail, which has no lower arcs, and the head, which has no higher arcs).

Each CSF expansion term corresponds to a walk from the graph tail to the graph head. This walk touches one node at each level, and it touches only the single arc at each level that connects the node below it to the node above it in that walk. A CSF can thereby be represented by denoting either the set of nodes in the corresponding walk or by denoting the sequence of steps in that walk. This latter choice is called the *step vector*

[†] Part of the special issue “Jack Simons Festschrift”.

[‡] E-mail: shepard@tcg.anl.gov.

10.1021/jp0543431 CCC: $30.25

TABLE 1: Characterization of Step Numbers

d	Δa_d	Δb_d	Δc_d	ΔN_d	ΔS_d
0	0	0	1	0	0
1	0	+1	0	1	$+^1/_2$
2	1	−1	1	1	$-^1/_2$
3	1	0	0	2	0

representation and is especially convenient because it is compact and it is independent of the (arbitrary) labeling and ordering of the nodes.

In a typical Shavitt graph an individual node may be touched by many walks, so it is convenient to organize the graph based on storage of the nodes; each node is indexed by an integer j and the storage of the connecting arcs, and other information discussed below, associated with that node is called a *distinct row table* (DRT). Each node (other than the root) is connected by at least one arc to nodes in the next lower level; consequently, the total number of arcs in the graph satisfies $(N_{\text{row}} - 1) \le N_{\text{arc}} < 4(N_{\text{row}} - 1)$. Each CSF may be assigned a contiguous integer index that may be computed as a summation of the integer *arc weights* that are associated with the arcs.

$$m = 1 + \sum_{p=0}^{n-1} y_{d_p j_p} = 1 + \sum_{p=0}^{n-1} y_{\mu(p)} \tag{2}$$

We adopt the convention that j_p is the node index of the bottom of the arc in the walk of interest at level p, and d_p is the step number associated with the arc. In this way the pair of indices (d,j) specify an arc. In the following, it is sometimes convenient to denote a (d,j) pair by a single arc index μ, and $\mu(p)$ in eq 2 is the arc at level p in the walk. These arc weights, along with the total number of walks, can be computed with a recursive procedure whose effort scales only as the number of nodes in the Shavitt graph and not as the (usually much larger) number of walks. From the information stored in the DRT, it is straightforward to construct the step vector from a given CSF index m, or to do the reverse and to determine the integer CSF index m from a given step vector; in both cases, the effort scales only as the number of orbitals n.

For the Shavitt graphs encountered with typical wave function expansions (e.g., MCSCF, MR-SDCI, full-CI), it is observed that the number of nodes is usually a small fraction of the total number of walks. For example, for full-CI expansions, the number of walks N_{csf} is given by[1]

$$N_{\text{csf}} = \frac{b+1}{n+1}\binom{n+1}{a}\binom{n+1}{c} \tag{3}$$

and the number of nodes (distinct rows) in the corresponding Shavitt graph is given by

$$N_{\text{row}} = (a+1)(c+1)\left(b + \frac{1}{2}d + 1\right) - \frac{1}{6}d(d+1)(d+2) \tag{4}$$

with $d = \text{Min}(a,c)$. Table 2 gives some selected values for full-CI singlet wave functions with $n = N$ up to 30 orbitals. For larger numbers of orbitals the Stirling approximation ($\ln(n!) \approx n\ln(n) - n + (1/2)\ln(2\pi n)$) may be used to estimate the binomial coefficients

$$N_{\text{row}} = (n+2)(n+3)(n+4)/24$$

$$N_{\text{csf}} \approx \left(\frac{8}{\pi}\right)\frac{4^n}{n^2} \tag{5}$$

TABLE 2: Comparison of N_{row} and N_{csf} for Singlet Full-CI Wave Function Expansions

$n = N$	N_{row}	N_{csf}
2	5	3
4	14	20
6	30	175
8	55	1,764
10	91	19,404
12	140	226,512
14	204	2,760,615
16	285	34,763,300
18	385	449,141,836
20	506	5,924,217,936
22	650	79,483,257,308
24	819	1,081,724,803,600
26	1015	14,901,311,070,000
28	1240	207,426,250,094,400
30	1496	2,913,690,606,794,775

giving, very roughly for large n

$$n \approx \text{Log}_4(N_{\text{csf}})$$

$$N_{\text{row}} \approx \frac{1}{24}\,\text{Log}_4(N_{\text{csf}})^3 \tag{6}$$

It is clear from the above approximation that $N_{\text{row}} \ll N_{\text{csf}}$ for large n. The $\text{Log}_4(N_{\text{csf}})$ quantity appears in these expressions because for these types of full-CI wave function expansions, almost all of the nodes are connected to 4 nodes at the next lower level (however, as we show below, this is not true for all wave function expansions). Our goal is to devise, as much as possible, a computational scheme that depends only on N_{row} rather than the usually much larger quantity N_{csf}.

To achieve this goal, we assign a numerical *arc factor* to each of the arcs in a given Shavitt graph. These arc factors will be denoted individually as α_{dj} where, analogous to the y_{dj} notation of the arc weights given above, j is the index of the node at the bottom of the arc and d is the step number of the arc. All of the upper arcs associated with node j will be denoted α_j, and the entire set of these arc factors will be denoted $\boldsymbol{\alpha}$. The CSF coefficient x_m associated with a particular walk m is defined to be the product of the arc factors in that walk. That is, in analogy to eq 2

$$x_m = \prod_{p=0}^{n-1} \alpha_{d_p j_p} = \prod_{p=0}^{n-1} \alpha_{\mu(p)} \tag{7}$$

Because one and only one arc factor is associated with each orbital level in this product, there are always exactly n arc factors that contribute to each of the CSF coefficients. The mapping of the set of arc factors to the vector of CSF coefficients will be denoted as $\mathbf{x} \equiv \mathscr{X}(\alpha)$. A product function, denoted $|M\rangle$, is then defined in terms of these CSF coefficients as

$$|M\rangle = \sum_{m=1}^{N_{\text{csf}}} x_m |\tilde{m}\rangle \tag{8}$$

A simple example of this relation is shown in Figure 1 for a 3-orbital, 3-electron, doublet full-CI expansion space. There are eight CSFs in the expansion, and given a set of arc factors α, the expansion coefficients for this

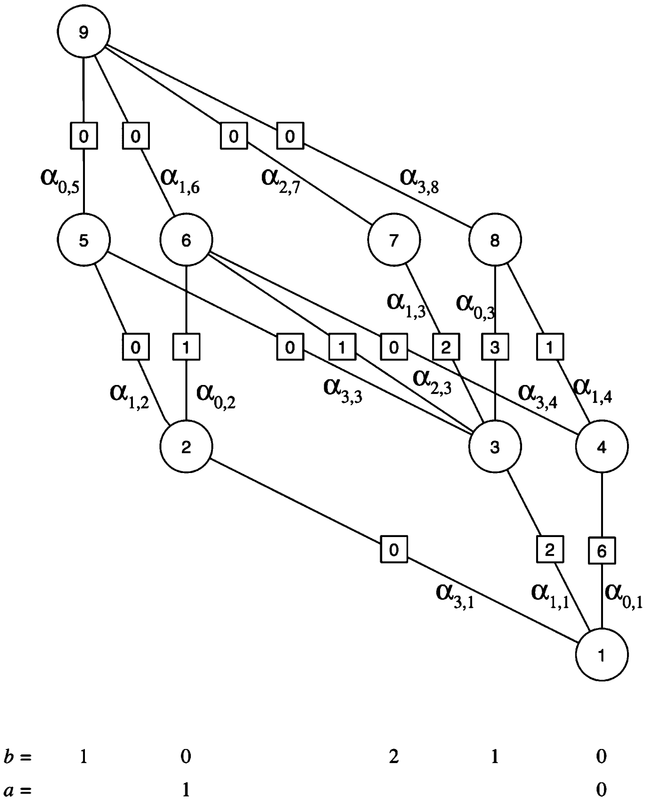

Figure 1. Shavitt graph for a three-electron, three-orbital, doublet full-CI expansion. The node index is denoted by the circled values, the arc weight y_{dj} is indicated by the square boxes, and the arc factor α_{dj} is written next to its corresponding arc.

product function are given by

$$\begin{aligned}|M\rangle &= \sum_{m=1}^{8} x_m|\tilde{m}\rangle \\ &= (\alpha_{3,1}\alpha_{1,2}\alpha_{0,5})|310\rangle + (\alpha_{3,1}\alpha_{0,2}\alpha_{1,6})|301\rangle + \\ &\quad (\alpha_{1,1}\alpha_{3,3}\alpha_{0,5})|130\rangle + (\alpha_{1,1}\alpha_{2,3}\alpha_{1,6})|121\rangle + \\ &\quad (\alpha_{1,1}\alpha_{1,3}\alpha_{2,7})|112\rangle + (\alpha_{1,1}\alpha_{0,3}\alpha_{3,8})|103\rangle + \\ &\quad (\alpha_{0,1}\alpha_{3,4}\alpha_{1,6})|031\rangle + (\alpha_{0,1}\alpha_{1,4}\alpha_{3,8})|013\rangle \end{aligned} \tag{9}$$

in which the CSFs are denoted with step-vectors. In this small 8-CSF expansion, there are 15 arc factors and 9 nodes. For larger expansions of this type, as discussed above, there are typically many more CSFs than nodes in the graph. However, this observation leads to the important topics of uniqueness, redundancy, normalization, and interpretation that we now address.

Given a set of arc factors, which define a product function $|M\rangle$, an expectation value may be written, for example, for the Hamiltonian operator, as

$$E = \frac{\langle M|\hat{H}|M\rangle}{\langle M|M\rangle} \tag{10}$$

Even for the normal situation, in which the x_m are considered as the variational parameters, there is a redundancy in the above expression because the coefficients may be scaled by an arbitrary nonzero factor β without changing the value of E. Even if the coefficients are scaled such that the denominator is unity, there is still the arbitrary choice of sign factor $\beta = \pm 1$. When $|M\rangle$ is a product function, there is even more ambiguity in the choices for the arc factors α. For example, if all the arc factors at a particular level are scaled by a factor β, then this affects only the overall norm and does not change any expectation value. Different levels could, in principle, be scaled by different scale factors, and only the overall norm would be affected. Furthermore, consider a situation such as node $j = 7$ in Figure 1. This node is connected by a single arc with $d = 1$ to the level below it, and it is connected by a single arc with $d = 2$ to the level above it. This pair of arc factors, $\alpha_{1,3}$ and $\alpha_{2,7}$, always occur together, and any scaling of this pair of arc factors of the form $\alpha_{1,3} \leftarrow \alpha_{1,3}/\beta$ and $\alpha_{2,7} \leftarrow \alpha_{2,7}\beta$ would leave the product $(\alpha_{1,3}\alpha_{2,7})$ unchanged and, therefore, would leave also $|M\rangle$ unchanged. With this kind of arbitrariness, given two different product functions $|M\rangle$ and $|N\rangle$, defined in terms of two sets of arc factors α^M and α^N, respectively, is it possible to determine if $|M\rangle = \beta|N\rangle$ simply by examining the individual arc factors? Finally, is there a way to attach a simple physical or mathematical meaning to the individual arc factors?

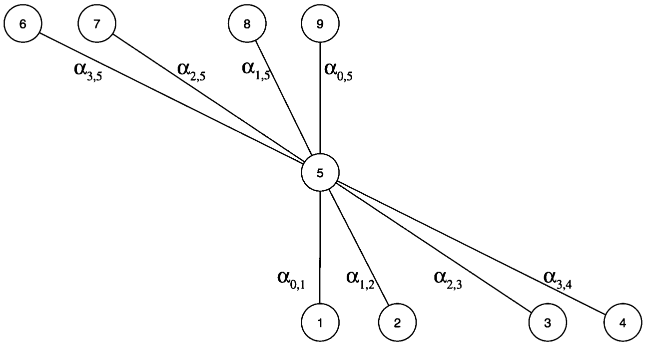

Figure 2. Arbitrary interior node of a Shavitt graph, labeled $j = 5$, shown with its connections to the nodes at the next lower and the next higher level. An arbitrary scaling of the lower arc factors by a factor $1/\beta$, and a simultaneous scaling of the upper arc factors by β, leave every product $(\alpha_{\text{low}}\alpha_{\text{high}})$ unchanged.

We address all of these issues by introducing a *standard form* for the arc factors. Given an arbitrary set of arc factors, it is possible to transform them into this standard form with the only significant change to $|M\rangle$ being an overall scaling. This standard form consists of scaling the individual terms in such a way to achieve overall normalization $\langle M|M\rangle = 1$, and it allows different sets of arc factors to be compared directly. To achieve this goal, we consider an arbitrary interior node of a Shavitt graph and consider all of the arcs that connect this node to the nodes at the adjacent levels above and below. Node $j = 5$ in Figure 2 is a representative example. It is clear that an arbitrary scaling of the lower arc factors by a factor $1/\beta$, and a simultaneous scaling of the upper arc factors by β, leave every possible product $(\alpha_{\text{low}}\alpha_{\text{high}})$ unchanged. That is, every walk that passes through that node will have its CSF coefficient unchanged by that scaling. The other walks in the expansion space that do not pass through that node would also be unchanged, the final result being that $|M\rangle$ itself is unchanged by such a scaling. This allows us to choose a particular scaling factor β for each node to enforce, in principle, any scaling convention that we choose.

The convention that is proposed here is based on the idea of normalization of lower walk *partial product functions*. Each node of the Shavitt graph is associated with such a partial product function, and the construction is defined recursively in the following manner. Referring to Figure 2, assume that normalized partial product functions have been computed for the nodes $j = 1...4$, and let these functions be denoted $|M_j\rangle$ with normalization $\langle M_j|M_j\rangle = 1$. We then define the partial product

```
do $p = 1, n$  ! loop over levels
    do $j = First(p), Last(p)$  ! loop over nodes in level $p$
        compute $\beta = \pm\left(\sum_{d=0}^{3} \alpha_{d,k_{d,j}}^2\right)^{\frac{1}{2}}$  where $k_{d,j} \neq 0$
        set $\alpha_{d,k_{d,j}} \leftarrow \alpha_{d,k_{d,j}} / \beta$  for $d=0\ldots3$ where $k_{d,j} \neq 0$
        set $\alpha_{d,j} \leftarrow \alpha_{d,j}\beta$  for $d=0\ldots3$ where $\ell_{d,j} \neq 0$
    enddo $j$
enddo $p$
```

Figure 3. Outline of the procedure to convert an arbitrary set of arc factors α to standard form.

function at node $j = 5$ as

$$|M_5\rangle = \alpha_{0,1}|M_1 \otimes 0\rangle + \alpha_{1,2}|M_2 \otimes 1\rangle + \alpha_{2,3}|M_3 \otimes 2\rangle + \alpha_{3,4}|M_4 \otimes 3\rangle \tag{11}$$

The notation $|M_j \otimes d\rangle$ means that a new orbital with step number d is appended to each of the step vectors in the expanded representation of the partial product function $|M_j\rangle$. We impose normalization on $|M_5\rangle$ to give

$$\begin{aligned} 1 &= \langle M_5|M_5\rangle \\ &= \alpha_{0,1}^2\langle M_1 \otimes 0|M_1 \otimes 0\rangle + \alpha_{1,2}^2\langle M_2 \otimes 1|M_2 \otimes 1\rangle + \\ &\quad \alpha_{2,3}^2\langle M_3 \otimes 2|M_3 \otimes 2\rangle + \alpha_{3,4}^2\langle M_4 \otimes 3|M_4 \otimes 3\rangle \\ &= \alpha_{0,1}^2 + \alpha_{1,2}^2 + \alpha_{2,3}^2 + \alpha_{3,4}^2 \end{aligned} \tag{12}$$

Given an initial set of arc factors, the scale factor β may be chosen to satisfy this normalization condition. The partial product function of the graph tail at level $p = 0$, corresponding to the physical vacuum, is normalized by convention. The conversion of an arbitrary set of arc factors α to standard form is outlined in Figure 3. In Figure 3, $k_{d,j}$ are the downward chaining indices, and $\ell_{d,j}$ are the upward chaining indices[1] that give the connecting node indices at the lower and higher levels, respectively. The sign of the β factor is chosen such that the lower arc factor with largest magnitude, after scaling, is positive. The quantity $\langle M|M\rangle$ is not changed during the above process for $p < n$; it is only at the last step, for the head node at $p = n$, that the normalization is changed to satisfy the desired $\langle M|M\rangle = \langle M_{\text{head}}|M_{\text{head}}\rangle = 1$ normalization condition. In the special case that $\beta = 0$ for some node, the lower arc factors may be set to an arbitrary set of values and the scaling of the upper arc factors by $\beta = 0$ will ensure that the product function $|M\rangle$ remains unchanged; by convention, in the standard form the arc factor of the lowest step number is set to +1 and any remaining arc factors are set to zero.

After a set of arc factors have been transformed into standard form, an individual arc factor α_{dj} is seen to have a simple physical and mathematical interpretation. Namely, it is the expansion coefficient used to construct the partial wave function at the node $\ell_{d,j}$ according to eq 11. As such, it is a measure of the relative importance of the partial wave function $|M_j\rangle$ within the partial wave function $|M_{\ell_{d,j}}\rangle$, and thereby, indirectly through the higher arc factors, to all of the higher partial wave functions including ultimately $|M\rangle$. If a particular arc factor is large in magnitude, then it means that that particular orbital occupation and spin coupling combination are important in forming that partial wave function, and if that arc factor is small, then it means that particular combination is not important. This allows, for example, physical interpretation of the relative importance of different arc factors within the graph, and of particular combinations of arc factors within the graph in the same general way that, for example, an electron density is a measure of the importance of a particular orbital.

The above arc factor normalization shows that if a node j has η_j lower arcs (with $1 \leq \eta_j \leq 4$), then there are only $(\eta_j - 1)$ independent degrees of freedom among those arc factors. This is equivalent to the constrained movement on the surface of a unit η-sphere with $(\eta - 1)$ essential variables. When considering the dependence on the arc factors of expectation values and other properties, it is often beneficial to cast the formulation in terms of a minimal number of essential variables. There are many ways to parametrize the constrained movement on the surface of a η-sphere. Our choice is based on our experience with the parametrization of normalized orbital and CSF coefficients.[8,9,16] The essential variables φ_{qj} for $q = 1, ..., (\eta-1)$ are associated with node j. The corresponding arc factors for that particular node are defined according to

$$\begin{gathered} r = |\varphi_j| \\ \alpha_{\mu(1,j)} = \cos(r) \\ \alpha_{\mu(q+1,j)} = \sin(r)\varphi_{q,j}/r \quad \text{for } q = 1, ..., (\eta_j - 1) \end{gathered} \tag{13}$$

where $\mu(:,j)$ are the η_j lower arcs associated with node j. In other words, this parametrization is equivalent to starting with an arbitrary unit vector $\hat{e}_\varphi = \varphi_j/r$ that is orthogonal to $\hat{e}_1$, and the magnitude of φ_j defines an arbitrary rotation of a unit vector away from $\hat{e}_1$ within the $(\hat{e}_1,\hat{e}_\varphi)$ plane. The result is constrained movement on the surface of the unit η-sphere with coordinates given by α_j. This relationship between α_j and φ_j is local to node j of the graph. The relation can also be inverted; given an α_j in standard form, a corresponding φ_j can be determined, although due to the cyclic nature of the trigonometric functions, the inverse mapping from α_j to φ_j is not unique. For $\eta_j = 1$ the normalization condition requires $\alpha_{\mu(1,j)} = \pm 1$, and, by convention, we take the positive value.

If each of the nodes j, other than the root, has η_j lower connecting arcs that are parametrized by $(\eta_j - 1)$ essential variables, then it follows that the total number of essential variables required to characterize an entire set of arc factors α is given by the expression

$$\begin{aligned} N_\varphi &= \sum_{j(\neq \text{root})}^{N_{\text{row}}} (\eta_j - 1) = \Big(\sum_{j(\neq \text{root})}^{N_{\text{row}}} \eta_j\Big) - (N_{\text{row}} - 1) \\ &= N_{\text{arc}} - (N_{\text{row}} - 1) \end{aligned} \tag{14}$$

If there are more variables than N_φ given above, then the representation of α in standard form in terms of those variables is either not unique or some of those variables are unnecessary, and if there are fewer variables than this number, then there are α sets that cannot be represented. In other words, this number of variables is both necessary and sufficient to represent an arbitrary α in standard form, and, in turn, that α in standard form is sufficiently flexible to represent (to within a sign) an arbitrary normalized product function.

We note in passing that other arc factor normalization conventions are also possible. For example, if a reference walk (with a nonzero coefficient) is chosen, then a scale factor for each level may be chosen such that the arcs that are touched by the reference walk all have arc factors of +1. The end result would be that the CSF coefficient for that reference walk would have the value $x_m = +1$, with the other CSF coefficients being scaled accordingly. This is equivalent to the traditional intermediate normalization convention. Another reasonable convention would be to choose scaling factors for each node such that the lower arc factor with largest magnitude always has a value of $+1$ after scaling. This is a slightly different way to achieve an intermediate normalization in which the CSF coefficient of largest magnitude after scaling would have a value of $+1$. Yet another reasonable convention would be to start at the head of the Shavitt graph and proceed down level by level, and scale the upper arc factors for each node to maintain normalization of the upper walk partial product functions rather than the lower walk partial product functions. It is possible that these conventions, or other similar conventions, would have particular advantages in special situations. In such a situation, it is very easy to take an arbitrary set of arc factors and to transform them to satisfy the desired normalization convention without changing the resulting product function $|M\rangle$ other than by an overall scaling.

Referring back to the DRT in Figure 1, we note that the number of essential variables is $N_\varphi = 15 - (9 - 1) = 7$. For an 8-CSF expansion space, there are indeed 7 degrees of freedom for a normalized wave function, which means that, for this small example, a single product function $|M\rangle$ has the same flexibility as the linear expansion space. It may be proven by induction that for an arbitrary Shavitt graph $N_\varphi < N_{\text{csf}}$; that is, there are always fewer essential variables than there are total walks in a Shavitt graph. In some small expansions, such as the graph in Figure 1, N_φ takes its maximum value of $N_\varphi = (N_{\text{csf}} - 1)$, and in these cases an arbitrary normalized wave function within the space may be represented by a single product function.

In general, however, and for most expansions of interest, a single product function is not sufficiently flexible to reproduce an arbitrary vector within the underlying linear expansion space. Consequently, we write a more general linear combination wave function expansion as

$$|\psi\rangle = \sum_{M}^{N_\alpha} c_M |M\rangle \tag{15}$$

in which the N_α product functions $|M\rangle$ form an expansion basis. The optimization of the linear expansion coefficients to minimize the energy expectation value takes the form of a generalized symmetric eigenvalue equation

$$\mathbf{Hc} = \mathbf{Sc}E \tag{16}$$

with $H_{MN} = \langle M|\hat{H}|N\rangle$ and $S_{MN} = \langle M|N\rangle$. In the present work we will discuss in detail only the computation of the metric matrix **S**. The detailed discussion of the computation of the Hamiltonian matrix **H** will be deferred to a future publication. Through the Ritz variational principle, the lowest eigenvalues computed from the product function basis in eq 16 are upper bounds to the corresponding eigenvalues of the underlying linear CSF expansion space, which in turn are upper bounds to the exact full-CI eigenvalues. Consequently, the general approach outlined here is applicable to both ground and excited electronic states.

If $\mathbf{x}^M = \mathcal{A}(\alpha^M)$ and $\mathbf{x}^N = \mathcal{A}(\alpha^N)$ are the vectors of CSF coefficients of the product functions defined by α^M and α^N, then $S_{MN} = \mathbf{x}^M{\cdot}\mathbf{x}^N$ is the scalar product between the two product basis functions. One way to compute this quantity would be to compute the vectors $\mathbf{x}^M$ and $\mathbf{x}^N$ explicitly, and to compute the scalar products directly from these expanded vectors. The effort for such an approach would scale linearly with N_{csf} and with the total number of distinct elements of the matrix **S**; because **S** is symmetric, this effort would be proportional to $N_{\text{csf}}N_\alpha(N_\alpha+1)$. Depending on the details of the implementation, the storage requirements with this approach might include also the $(N_{\text{csf}}N_\alpha)$ elements of $\mathbf{X} = [\mathbf{x}^1|\mathbf{x}^2|...\mathbf{x}^{N\alpha}]$. Our actual approach requires both less computational effort and less storage for large expansions.

We approach the solution to this task recursively. Suppose that we have available the overlaps of the partial product functions for two expansion terms, labeled $|M_j\rangle$ and $|N_j\rangle$ for all nodes j at some level, and consider computing the overlaps of the partial product functions at the next higher level. Referring to Figure 2, suppose that we have the quantities $\gamma_j^{MN} = \langle M_j|N_j\rangle$ for $j = 1...4$. Using

$$\begin{aligned}
|M_5\rangle &= \alpha_{0,1}^M|M_1 \otimes 0\rangle + \alpha_{1,2}^M|M_2 \otimes 1\rangle + \\
&\quad \alpha_{2,3}^M|M_3 \otimes 2\rangle + \alpha_{3,4}^M|M_4 \otimes 3\rangle \\
|N_5\rangle &= \alpha_{0,1}^N|N_1 \otimes 0\rangle + \alpha_{1,2}^N|N_2 \otimes 1\rangle + \\
&\quad \alpha_{2,3}^N|N_3 \otimes 2\rangle + \alpha_{3,4}^N|N_4 \otimes 3\rangle
\end{aligned} \tag{17}$$

it follows that

$$\begin{aligned}
\gamma_5^{MN} &= \langle M_5|N_5\rangle \\
&= \alpha_{0,1}^M\,\alpha_{0,1}^N\langle M_1 \otimes 0|N_1 \otimes 0\rangle + \\
&\quad \alpha_{1,2}^M\,\alpha_{1,2}^N\langle M_2 \otimes 1|N_2 \otimes 1\rangle + \\
&\quad \alpha_{2,3}^M\alpha_{2,3}^N\langle M_3 \otimes 2|N_3 \otimes 2\rangle + \\
&\quad \alpha_{3,4}^M\,\alpha_{3,4}^N\langle M_4 \otimes 3|N_4 \otimes 3\rangle \\
&= \alpha_{0,1}^M\,\alpha_{0,1}^N\,\gamma_1^{MN} + \alpha_{1,2}^M\,\alpha_{1,2}^N\,\gamma_2^{MN} + \\
&\quad \alpha_{2,3}^M\alpha_{2,3}^N\gamma_3^{MN} + \alpha_{3,4}^M\,\alpha_{3,4}^N\,\gamma_4^{MN}
\end{aligned} \tag{18}$$

This procedure may be applied to each of the nodes at the higher level. The overlap computation procedure begins at the tail of the graph with the assignment of the vacuum overlap $\gamma_{\text{tail}}^{MN} = \langle \text{vac}|\text{vac}\rangle = 1$ and proceeds upward level by level, until the graph head is reached, at which time we have

$$S_{MN} = \mathbf{x}^M{\cdot}\mathbf{x}^N = \langle M|N\rangle = \gamma_{\text{head}}^{MN} \tag{19}$$

which is the quantity of interest. Note that, by definition and by construction, $\gamma_j^{MN} = \gamma_j^{NM}$ for all nodes j in the Shavitt graph. For each node in the Shavitt graph, there are two floating point multiplications and one addition for each connecting lower arc, so the effort for this procedure scales only with the number of nodes, not as the number of walks. The storage requirements consist of the γ^{MN} array, of length N_{row}, and the resulting **S** array. (This assumes the entire γ array is kept; only storage of the rows for two levels at a time is strictly necessary for the procedure.) Some minor efficiency can be gained by computing the γ^{MN} arrays for a range of M and N values simultaneously. In this case, the storage scales as the product of N_{row} and the number of S_{MN} values that are being computed, which might be anywhere between 1 and $N_\alpha(N_\alpha + 1)/2$. This procedure is outlined in Figure 4.

We now characterize product basis functions and wave functions on the basis of the arc factors and partial product overlaps. Using the lower walk partial product

```
set $\boldsymbol{\gamma}^{MN} = \mathbf{0}$ for all M,N pairs
set $\gamma_{tail}^{MN} = 1$ for all M,N pairs
do p = 1, n ! loop over levels
    do j = First(p), Last(p) ! loop over nodes in level p
        Compute $\gamma_j^{MN}$ for all M,N pairs
    enddo j
enddo p
set $S_{MN} = \gamma_{head}^{MN}$ for all M,N pairs
```

Figure 4. Outline of the efficient computation of $S_{MN} = \langle M|N\rangle$.

functions $|M_j\rangle$ and the analogous upper walk partial product functions $|\bar{M}_j\rangle$, defined with the arcs and corresponding arc factors from the node j up to the head of the graph, the product function contribution from a particular node j is $|M_j \otimes \bar{M}_j\rangle$. A product function may be written as a summation of these contributions from all of the nodes at a particular level p.

$$|M\rangle = \sum_{\substack{j \\ \text{(in level } p)}} |M_j \otimes \bar{M}_j\rangle \tag{20}$$

A *transition node density* for node j may be defined as

$$\begin{aligned} D_j^{MN} &= \langle M_j \otimes \bar{M}_j | N_j \otimes \bar{N}_j\rangle \\ &= \gamma_j^{MN}\,\bar{\gamma}_j^{MN} \end{aligned} \tag{21}$$

For $M = N$, the node density D_j^{MM}, which is a nonnegative quantity, gives a measure of the overall importance of that particular node to the product function $|M\rangle$. The procedure described above for the computation of the γ^{MN} array begins at the graph tail and proceeds upward, one level at a time, to the graph head. In an entirely analogous manner, the array $\bar{\gamma}^{MN}$ with elements $\bar{\gamma}_j^{MN} = \langle \bar{M}_j|\bar{N}_j\rangle$ may be computed recursively from the head of the Shavitt graph down to a particular node. Referring to Figure 2 and eq 18, we define for example

$$\begin{aligned} \bar{\gamma}_5^{MN} = {} & \alpha_{0,5}^{M}\,\alpha_{0,5}^{N}\,\bar{\gamma}_9^{MN} + \alpha_{1,5}^{M}\,\alpha_{1,5}^{N}\,\bar{\gamma}_8^{MN} + \\ & \alpha_{2,5}^{M}\,\alpha_{2,5}^{N}\,\bar{\gamma}_7^{MN} + \alpha_{3,5}^{M}\,\alpha_{3,5}^{N}\,\bar{\gamma}_6^{MN} \end{aligned} \tag{22}$$

The value $\bar{\gamma}_{\text{head}}^{MN} = 1$ is assigned for the head of the graph, and then the lower $\bar{\gamma}_j^{MN}$ elements are computed, level by level, until the graph tail is reached. At this point,

$$S_{MN} = \bar{\gamma}_{\text{tail}}^{MN} \tag{23}$$

in analogy with eq 19. The overlap of two product functions may be written in terms of these transition node densities as

$$\langle M|N\rangle = \sum_{\substack{j \\ \text{(in Level } p)}} \gamma_j^{MN}\,\bar{\gamma}_j^{MN} = \sum_{\substack{j \\ \text{(in Level } p)}} D_j^{MN} \tag{24}$$

Equations 19 and 23 are special cases of this more general expression.

In an analogous manner, a particular arc factor contributes to a product function according to

$$|M\rangle = \sum_{\substack{\mu \\ \text{(in Level } p)}} \alpha_\mu^M |M_{\text{Bottom}(\mu)} \otimes d_\mu \otimes \bar{M}_{\text{Top}(\mu)}\rangle \tag{25}$$

and a *transition arc density* may be defined as

$$\begin{aligned} D_\mu^{MN} &= \alpha_\mu^M\,\alpha_\mu^N \langle M_{\text{Bottom}(\mu)} \otimes d_\mu \otimes \bar{M}_{\text{Top}(\mu)} | N_{\text{Bottom}(\mu)} \otimes d_\mu \otimes \\ &\qquad \bar{N}_{\text{Top}(\mu)}\rangle \\ &= \alpha_\mu^M\,\alpha_\mu^N \gamma_{\text{Bottom}(\mu)}^{MN}\,\bar{\gamma}_{\text{Top}(\mu)}^{MN} \end{aligned} \tag{26}$$

An overlap of two product functions may be written in terms of transition arc densities as

$$\langle M|N\rangle = \sum_{\substack{\mu \\ \text{(in Level } p)}} \alpha_\mu^M\,\alpha_\mu^N\,\gamma_{\text{Bottom}(\mu)}^{MN}\,\bar{\gamma}_{\text{Top}(\mu)}^{MN} = \sum_{\substack{\mu \\ \text{(in Level } p)}} D_\mu^{MN} \tag{27}$$

For $M = N$, D_μ^{MM}, which is a nonnegative quantity, gives a measure of the overall importance of that particular arc to the product function $|M\rangle$.

Using eq 15, the wave function overlap may be written in terms of node densities at some level p as

$$\begin{aligned} \langle\psi|\psi\rangle &= \sum_{M,N} c_M c_N \langle M|N\rangle \\ &= \sum_{\substack{j \\ \text{(in Level } p)}} \sum_{M,N} c_M c_N\,\gamma_j^{MN}\,\bar{\gamma}_j^{MN} \\ &= \sum_{\substack{j \\ \text{(in Level } p)}} \sum_{M,N} c_M c_N\,D_j^{MN} \\ &= \sum_{\substack{j \\ \text{(in Level } p)}} D_j^{\psi} \end{aligned} \tag{28}$$

and it may be written in terms of arc densities and individual arc factors as

$$\begin{aligned} \langle\psi|\psi\rangle &= \sum_{M,N} c_M c_N \langle M|N\rangle \\ &= \sum_{\substack{\mu \\ \text{(in Level } p)}} \sum_{M,N} c_M c_N\,\alpha_\mu^M\,\alpha_\mu^N\,\gamma_{\text{Bottom}(\mu)}^{MN}\,\bar{\gamma}_{\text{Top}(\mu)}^{MN} \\ &= \sum_{\substack{\mu \\ \text{(in Level } p)}} \sum_{M,N} c_M c_N\,D_\mu^{MN} \\ &= \sum_{\substack{\mu \\ \text{(in Level } p)}} D_\mu^{\psi} \end{aligned} \tag{29}$$

In the above expressions D_j^{ψ} is the *wave function node density* for node j and D_μ^{ψ} is the *wave function arc density* for arc μ. These quantities allow product basis functions and wave functions to be compared and characterized in terms of nodes and arcs of the Shavitt graph.

Another property of interest for electronic wave functions is the orbital occupation (i.e., a diagonal element of the one-particle density matrix), which may be computed for normalized wave functions as

$$D_{pp} = \langle\psi|\hat{E}_{pp}|\psi\rangle = \sum_{M,N} c_M c_N \langle M|\hat{E}_{pp}|N\rangle \tag{30}$$

where $\hat{E}_{pp}$ is a weight generator of the unitary group.[1,2,9,12]

Substitution using eq 25 in terms of the arcs at level p gives

$$\begin{aligned} D_{pp} &= \sum_{\substack{\mu \\ \text{(in Level } p)}} \Delta N_{d_\mu} \sum_{M,N} c_M c_N \, \alpha_\mu^M \, \alpha_\mu^N \, \gamma^{MN}_{\text{Bottom}(\mu)} \, \bar{\gamma}^{MN}_{\text{Top}(\mu)} \\ &= \sum_{\substack{\mu \\ \text{(in Level } p)}} \Delta N_{d_\mu} \sum_{M,N} c_M c_N \, D_\mu^{MN} \\ &= \sum_{\substack{\mu \\ \text{(in Level } p)}} \Delta N_{d_\mu} D_\mu^{\psi} \end{aligned} \tag{31}$$

The ΔN_d value is the occupation of the arc as given in Table 1. In this manner, the computation of the orbital occupations may be performed with an effort that depends only on N_α and N_{row}. The expression for the off-diagonal density matrix elements D_{pq} is more complicated and will be examined in more detail in a later publication. However, the computation of these elements follows the same general approach as above. Namely, the explicit construction and storage of the CSF vectors $\mathbf{x}^M$ is avoided, and as much effort as possible is cast into the form of recursive procedures that scale as N_{row} rather than N_{csf}.

We do not consider in detail here the analogous computation of the matrix **H**. We note here only that the efficient **H** matrix construction follows the same general guidelines as those used to construct **S**, the orbital occupations, and the node and arc densities. Namely, the explicit construction and storage of the CSF vectors $\mathbf{x}^M$ is avoided, and as much effort as possible is cast into the form of recursive procedures that scale as N_{row} rather than N_{csf}.

We have not discussed the use of point group symmetry in the above formulation. We have considered two possible approaches. One is to ignore point group symmetry entirely. In this case a product wave function would, in general, contain mixtures of wave functions corresponding to different irreducible representations (irreps), and the expectation values will consist of averages over these irreps. That is, in general, a product function would have symmetry contamination. Presumably, the linear combination of several product functions would then allow the desired wave function and the desired expectation value to emerge. The other possibility, which is the one we have chosen for the results presented in this paper, is based on symmetry-dependent arc weights.[17] This approach, which is used by the GUGA codes in COLUMBUS,[10,11] allows the walks belonging to each separate irrep at each node to be identified and indexed separately. In particular, we define at each node j and for all irreps Γ a set of partial product functions denoted $|M_j^\Gamma\rangle$ and we compute, for example, the corresponding overlaps $\gamma_j^{MN,\Gamma}$. Consequently, the CSF expansions with this approach all correspond to the single irrep (or, in the case of state averaging, to the specific set of irreps) of interest, and the expectation values will have no unwanted symmetry contamination. Given this choice of symmetry treatment, there are two possibilities for the arc weights. Either a single arc factor could be associated with each arc and used for walks corresponding to all irreps, or separate arc factors could be associated with each symmetry version of each arc. The latter choice would result in more flexibility in the product function but at the cost of more arc factors. In the present work, we have chosen the former simpler approach. Our actual implementation is only slightly more complicated than the simple treatment presented above, which ignores point group symmetry. The other possible options for treating point group symmetry will be examined in more detail in the future.

Given a product function $|M\rangle$, it might be useful to characterize it qualitatively by examining a few of the CSF coefficients of largest magnitude. One way to do this would be to compute the CSF vector **x** and to sort the elements. Using an efficient sort procedure, the effort for this would scale between N_{csf} and N_{csf} Log(N_{csf}) depending on how many coefficients are computed and sorted. A more efficient procedure results from the observation that the coefficients of largest magnitude within a partial product function $|M_j\rangle$ may be determined by examining just the coefficients of largest magnitude of the partial product functions of the lower connecting nodes. This suggests a recursive procedure to extract the largest coefficients from $|M_{\text{head}}\rangle$. For example, if the largest Q coefficients are required, then the overall effort for this recursive procedure scales as the product QN_{row}, eliminating all factors related to N_{csf}. For $Q \ll N_{\text{csf}}$, this is much more efficient than the **x**-construction approach.

It is possible to represent electronic wave functions in terms of primitive Slater determinants instead of spin-eigenfunctions, and the spin−orbital occupations of those Slater determinants can be represented efficiently using graphical approaches[18] similar to GUGA. Product wave functions may be formulated in terms of arc factors in these graphical representations in a manner entirely analogous to the method described above. Unless additional constraints of some kind are imposed on the arc factors, such product functions in general would not be $\hat{S}^2$ spin-eigenfunctions. However, such product functions would have more flexibility (i.e., more arc factors within each product function) than those presented in the present work, and it is not clear whether that additional flexibility would compensate in a practical way for the disadvantages associated with the spin contamination.

3. Results and Dicussion

We first discuss a few general features of the product functions described in the previous section. We note that if all of the arc factors α_{dj} are set to zero except those touched by a particular walk m, then we have the identity $|M\rangle = |\tilde{m}\rangle$ where $|\tilde{m}\rangle$ is a primitive expansion CSF. The form of the product function is therefore capable of representing any individual CSF in the expansion space, regardless of its excitation level relative to some reference CSF.

In the limit that $N_\alpha = N_{\text{csf}}$, with $|M\rangle = |\tilde{m}\rangle$ for all m, we would have an orthonormal product function basis with full rank. In other words, the use of the product basis rather than the primitive CSF basis represents no inherent formal limitation or approximation relative to the underlying linear expansion space. Of course, the truncated product function bases that will be used in practice for large wave function expansions will not have full rank. The remaining question is how many expansion terms N_α will be required to represent with sufficient accuracy the Hamiltonian eigenvectors of chemical interest. This is discussed below.

Consider next the standard form for a product function based on the PPMC expansion space.[9,19] A Shavitt graph for a six-orbital PPMC expansion space is shown in Figure 5. In a PPMC expansion, the orbitals are grouped into pairs, and occupation restrictions are imposed such that exactly two electrons occupy each of the orbital pairs. Furthermore, no open-shell CSFs are included, so each orbital is either empty or doubly occupied in each of the expansion CSFs. As discussed elsewhere,[9] such a wave function expansion form allows both single and multiple chemical bonds to be broken without spurious charge contamination. With these restric-

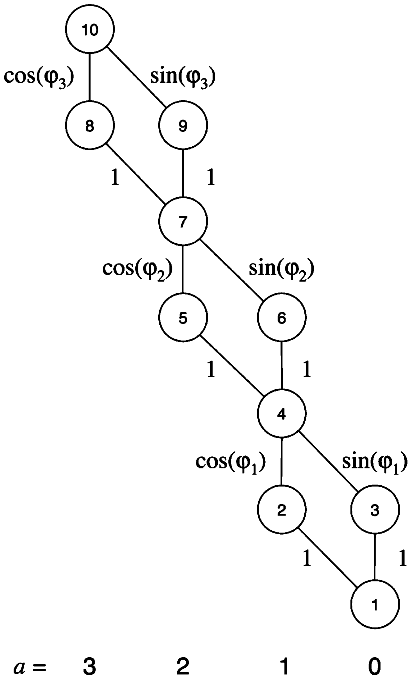

Figure 5. Shavitt graph and arc factors in standard form shown for a six-orbital PPMC expansion space. The product function for this graph is equivalent to a nonlinear pp-GVB wave function.

tions, there are two arcs at each orbital level, and the two nodes in each of the odd levels have only a single lower connecting arc each, whereas the single node at each of the even orbital levels has two lower connecting arcs. For the odd-level nodes the lower arc factors are $\alpha_{dj} = 1$; that is, they contribute no essential variables. For each even-level node, the two lower arc factors are determined by a single essential variable. The unit two-sphere parametrization discussed in the previous section reduces to the simple form

$$\left.\begin{aligned}\alpha_0 &= \cos(\varphi)\\ \alpha_3 &= \sin(\varphi)\end{aligned}\right\} \quad \text{for lower arcs in even levels} \tag{32}$$

These arc factors are shown explicitly in Figure 5. The linear expansion space for the PPMC wave function in general has dimension $N_{\text{csf}} = 2^{n/2}$. It is easy to verify that the product function may be written in the step-vector form

$$|\{\cos(\varphi_1)30 + \sin(\varphi_1)03\} \otimes \{\cos(\varphi_2)30 + \sin(\varphi_2)03\} \otimes \cdots \otimes \{\cos(\varphi_{n/2})30 + \sin(\varphi_{n/2})03\}\rangle = |G_1 \otimes G_2 \otimes \cdots \otimes G_{n/2}\rangle \tag{33}$$

or in first-quantization form using the antisymmetrized product of geminals

$$\mathcal{A}[\{\cos(\varphi_1)\chi_1{}^2 + \sin(\varphi_1)\chi_1{}^{*2}\}\{\cos(\varphi_2)\chi_2{}^2 + \sin(\varphi_2)\chi_2{}^{*2}\} \cdots \{\cos(\varphi_{n/2})\chi_{n/2}{}^2 + \sin(\varphi_{n/2})\chi_{n/2}{}^{*2}\}] \tag{34}$$

in which the orbitals in pair k are denoted χ_k and χ_k*. This is the normalized form of a nonlinear pp-GVB wave function.[19] For this particular type of Shavitt graph, corresponding to PPMC expansions, the product function based on that same Shavitt graph is exactly equivalent to the corresponding nonlinear pp-GVB wave function. The product functions for more general Shavitt graphs (e.g., those that have the above nodes and arcs as a subset) have the capability to represent pp-GVB wave functions by setting the additional arc factors to zero. For a molecule that dissociates into a set of noninteracting, two-electron, two-orbital, singlet fragments, the nonlinear pp-GVB wave function reproduces the corresponding exact full-CI wave function. Consequently, a single product function that is based on a Shavitt graph that contains the PPMC Shavitt graph as a subgraph also has this capability.

If the fragments of such a dissociation are not simple two-electron subunits (or other special cases such as discussed above for Figure 1), then in general a single product function will not be equivalent to the full-CI wave function. Assume that all of the orbitals associated with these individual singlet fragments are grouped together in the Shavitt graph, and that there are no additional spin or occupation restrictions that would prevent the molecule from dissociating properly into these fragments. In this case the product function for the entire molecule can be written as products of the fragment functions, the energy expectation value will be the sum of the energies of the fragments, and an important size-consistency property will be satisfied. This property also holds for some mixtures of singlet and nonsinglet fragments. Further work is necessary to fully understand this feature of the product functions described in this work.

We next turn to the question of accuracy of the linear combination wave functions with respect to product basis dimension N_α. At present, we do not have the capability to optimize the linear coefficients c_M in eq 15 and the nonlinear arc factors α^M directly to minimize the energy expectation value. We do, however, have the capability to compute a Hamiltonian eigenvector $\mathbf{v}^{\text{exact}}$ independently and, given that vector, to vary the coefficients c_M and arc parameters α^M to minimize the error $\sigma^2 = |\mathbf{v} - \mathbf{v}^{\text{exact}}|^2$. This is a somewhat simpler task than direct energy-based optimization, yet it still can answer the immediate questions about accuracy and convergence with respect to N_α. The details of this optimization process are given in the Appendix. Much of the technology presented in the Appendix will also apply to the computation and optimization of the energy expectation value. Our optimization approach is based on the efficient computation of the metric **S** using the recursive method discussed in section 2, and on the efficient computation of the quantities $\mathbf{z} = \mathbf{X}^T\mathbf{v}^{\text{exact}}$, along with the analytic derivatives of these quantities with respect to the essential variables φ. We have computed the σ^2 error for a representative set of Hamiltonian eigenvectors. The molecules and their wave function expansions are summarized in Table 3. We have used a variety of optimization approaches including both those that require gradients and those that do not. Our results presented below are computed with the CG_DESCENT() procedure of Hager and Zhang,[20] which is a conjugate gradient optimization procedure with line searches that does require gradients.

The first wave function is a 6-orbital, 6-electron, CASSCF expansion for the N_2 molecule using D_{2h} point group symmetry. There are 175 CSFs total in the expansion space, 32 of which belong to the A_{1g} irrep of the ground state. We examine two bond distances for this molecule, one at R_e (=2.074a_0) and one at a stretched bond distance of $R = 4.0a_0$. At R_e the largest CSF coefficient is 0.9856, whereas at the stretched distance the largest CSF coefficient is 0.6334 meaning that the corresponding CSF constitutes only about 40% of the wave function. The product basis functions were optimized in two different ways. In the sequential approach, each product function is added to the basis, optimized to minimize σ^2, and then those variables are frozen as the next product function is added. Each optimization consists of optimizing only the φ^M for $M = N_\alpha$ variables

TABLE 3: Wave Function Summary

wave function	N_{row}	N_{arc}	N_{φ}	N_{csf}	$N^{\Gamma}_{\mathrm{csf}}$
$\mathbf{6}^{6}$ N_2 in D_{2h}	30	68	39	175	32 + 20 + 20 + 20 + 23 + 20 + 20 + 20
$\mathbf{2}^{2}\mathbf{2}^{2}\mathbf{2}^{2}\mathbf{2}^{2}\mathbf{2}^{2}\mathbf{2}^{2}$ C_2H_4 in C_s	52	90	39	3012	3012
$\mathbf{10}^{12}$ H_2CO in C_{2v}	85	230	146	13860	3644 + 3384 + 3496 + 3336
$\mathbf{12}^{18}$ O_3 in C_{2v}	90	236	147	15730	4067 + 3858 + 3962 + 3843
SD from $\mathbf{3}^{6}\mathbf{4}^{4}$ H_2O in C_s	105	226	122	40539	20465 + 20074

and the **c** vector. This is analogous to the way that subspace bases are expanded in Ritz subspace eigenvalue methods such as the Davidson and Lanczos methods. The second optimization approach consists of optimizing all of the φ^M variables for $M = 1, ..., N_\alpha$ along with **c** at each step. For a given N_α, this is computationally more demanding than the sequential optimization approach, but as seen in Figure 6 it converges much faster. For the R_e calculation, the full optimization approach converges to $\sigma^2 < 10^{-7}$ with $N_\alpha = 3$ for the full optimization, whereas $N_\alpha = 10$ product functions are required to achieve that accuracy with sequential optimization. In comparing the R_e with the $R = 4$ calculation with full optimization, it is seen that the R_e error is less than the $R = 4$ error for $N_\alpha = 1$ and $N_\alpha = 2$; both the R_e and $R = 4$ errors are $\sigma^2 < 10^{-10}$ for $N_\alpha = 3$. This calculation is too small to reliably measure the time required to compute the gradient, $\tilde{g}_{rM} = \mathrm{d}\sigma^2(\varphi)/\mathrm{d}\varphi_r^M$, with either a finite difference approach or the analytic approach described in the Appendix.

The next row of Table 3 corresponds to a six-pair (12-orbital, 12-electron) RCI-GVB wave function expansion for the ground state of the ethylene molecule, C_2H_4. This expansion has the same orbital pair occupation restrictions as the PPMC expansion, but all possible open-shell CSFs and all possible spin couplings are included in the expansion space.[9] Although the PPMC expansion is sufficiently flexible to eliminate any spurious ionic contamination during any dissociation process (e.g., to the fragments H, C_2H_3, CH_2, CCH_2, etc.), it cannot, in general, dissociate to fragments that are the correct spin-eigenfunctions. The more flexible RCI-GVB expansion on the other hand does dissociate to spin-eigenfunction fragments and includes also the most important interpair correlation[19] contributions. The product function optimization is again performed in both ways, with sequential optimization and with full optimization. As seen in Figure 6, the error with full optimization is significantly less than the error with sequential optimization. In the full optimization case, $N_\alpha = 4$ product functions are required to achieve $\sigma^2 < 10^{-3}$, and $N_\alpha = 10$ are required to achieve $\sigma^2 < 10^{-4}$. The error decreases reasonably well for up to $N_\alpha = 5$, and then it slows; it is not clear why this occurs, or if it is simply an optimization artifact of some kind. The sequential optimization requires $N_\alpha = 10$ to achieve $\sigma^2 < 10^{-3}$. This calculation is too small to reliably measure the time required to compute an analytic gradient, but a finite difference gradient requires about 0.11 s on a 2.8 GHz P4 computer.

The third molecule in Table 3 is a 10-orbital, 12-electron, CASSCF in C_{2v} symmetry for the formaldehyde molecule. With

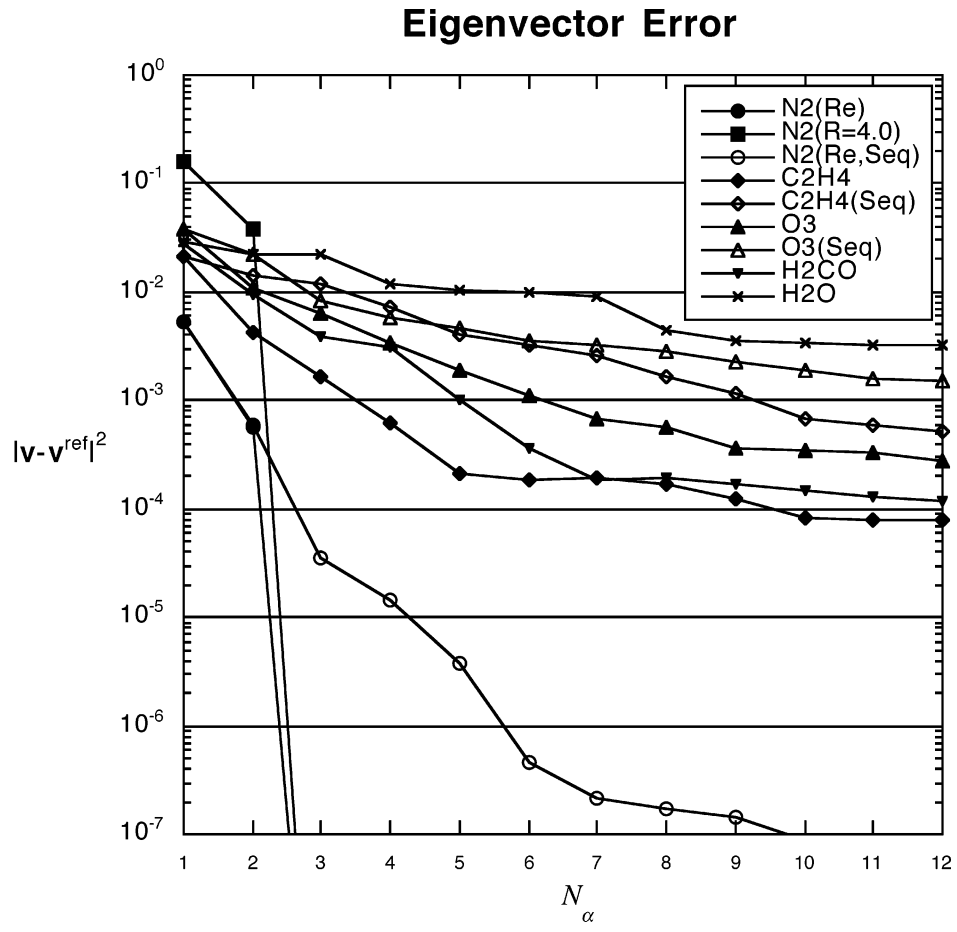

Figure 6. Eigenvector error as a function of product basis dimension N_α for some representative wave function expansions.

full optimization, $N_\alpha = 5$ product functions are required to achieve $\sigma^2 < 10^{-3}$, and $N_\alpha = 14$ are required to achieve $\sigma^2 < 10^{-4}$. The error decreases reasonably well for up to $N_\alpha = 7$, and then it slows. This calculation is too small to reliably measure the time required to compute an analytic gradient, but a finite difference gradient for $N_\alpha = 1$ requires about 0.48 s.

The fourth molecule in Table 3 is a 12-orbital, 18-electron, CASSCF in C_{2v} symmetry for the ozone molecule. With full optimization, $N_\alpha = 7$ product functions are required to achieve $\sigma^2 < 10^{-3}$, and convergence to $\sigma^2 < 10^{-4}$ is not achieved by $N_\alpha = 16$. The error decreases reasonably well for up to about $N_\alpha = 9$, and then it slows. The sequential optimization does not achieve convergence to even $\sigma^2 < 10^{-3}$ by $N_\alpha = 16$. This calculation is too small to reliably measure the time required to compute an analytic gradient, but a finite difference gradient requires about 0.57 s.

The fifth molecule in Table 3 is a single- and double-excitation CI from the 4-orbital, 4-electron, CASSCF in C_{2v} symmetry for the water molecule using a standard cc-pVDZ orbital basis set. With full optimization, $N_\alpha = 7$ product functions are required to achieve $\sigma^2 < 10^{-2}$, and convergence to $\sigma^2 < 10^{-3}$ is not achieved even by $N_\alpha = 16$. The error decrease stalls several times along the way. Convergence of the conjugate gradient procedure was particularly problematic for this calculation. The analytic gradient for this calculation takes about 0.0001 s, whereas a finite difference gradient for $N_\alpha = 1$ requires about 1.48 s on a 2.8 GHz P4 computer. We see that there is about a 10^4 ratio difference in efficiency when computing analytic gradients using the procedure outlined in the Appendix. The conjugate gradient optimization for this problem requires between 1.5×10^4 and 2.0×10^6 function and gradient evaluations, depending on N_α; thus this calculation would not have been practical without the analytic gradient procedure described in the Appendix.

The overall efficiency of the evaluations of σ^2 and of the analytic gradients $\tilde{g}_{rM} = d\sigma^2(\varphi)/d\varphi_r^M$ is encouraging. Much of the technology involved in this process will be used eventually in the evaluation of **H** matrix elements and in the direct energy-based optimization of **c** and φ. In the current calculations it is necessary to have several vectors of length N_{csf} for debugging and development purposes. Eventually all reference to any vectors of length N_{csf} will be removed from the procedure, and much larger expansions can be examined. It is also encouraging that for the sample calculations so far, it appears that typically $N_\alpha < \sim 20$ will be sufficient to represent wave functions to chemical accuracy using a full-optimization approach. The sequential-optimization approach converges slower than the full-optimization approach. This suggests that Davidson-like or Lanczos-like approaches to build the product function subspaces may not be optimal. The water molecule calculation exhibits several kinds of convergence problems, both with respect to σ^2 as a function of N_α and during the low-level conjugate gradient optimization steps. At this point we are hopeful that these convergence problems can be solved. It is also observed that if the numerical optimization procedure is started with different initial guesses for φ, the $N_\alpha = 1$ solution will sometimes converge to different product functions; in all of these cases, the $N_\alpha \geq 2$ full-optimization solutions are the same regardless of the initial guess. This suggests that, at least for some situations, there are local minima encountered during the convergence trajectory that thwarts convergence to the desired global minimum error solution. It remains to be seen if this applies also to the direct energy-based optimization of φ.

4. Conclusions

A new expansion form has been described for electronic wave functions that is based on the GUGA method of Shavitt. The wave function is a linear combination of product basis functions, and each product basis function in turn is formally equivalent to a linear combination of configuration state functions that comprise an underlying linear expansion space. The CSF expansion coefficients that define the basis functions are nonlinear functions of a set of arc factors α^M, and the arc factors themselves may be represented in terms of a smaller number of essential variables φ^M. A standard form has been defined for the arc factors that allows for an intuitive physical interpretation, and it allows different wave functions to be compared by examining the individual arc factors. Node densities and arc densities have been defined that allow the individual product functions and the resulting wave functions to be compared, analyzed, and characterized in terms of the nodes and arcs of the Shavitt graph. Preliminary calculations suggest that $N_\alpha < \sim 20$ basis functions are sufficient to approximate typical Hamiltonian eigenvectors. The method described here is appropriate for both ground and excited states and to both closed and open shell molecules. In some cases, the method described here will be size-consistent with respect to the dissociation of molecules into fragments. There are many computations described in this work (such as basis function overlaps, orbital occupations, node densities, arc densities, and the determination of the largest CSF coefficient, or of the largest few coefficients) that may be cast directly and efficiently in terms of the arc factors and that do not require the explicit computation and storage of the CSF expansion vector.

Acknowledgment. The author is grateful to Professor J. Simons who, as graduate advisor, instilled a lasting interest in electronic structure theory. The author thanks Professor I. Shavitt for many helpful discussions regarding GUGA in general and this work in particular. This work was supported by the U.S. Department of Energy by the Office of Basic Energy Sciences, Division of Chemical Sciences, under contract W-31-109-ENG-38.

Appendix

In this Appendix, we show the details of the computation and optimization of the quantity

$$\sigma^2(\alpha,\mathbf{c}) = |\mathbf{v} - \mathbf{v}^{\mathrm{ref}}|^2 \tag{35}$$

with

$$\mathbf{v} = \sum_{M=1}^{N_\alpha} \mathbf{x}^M c_M = \mathbf{X}\mathbf{c} \tag{36}$$

N_α is the number of product function expansion terms, $\mathbf{x}^M \equiv \mathbf{x}(\alpha^M) = \mathcal{L}(\alpha^M)$ is the linear representation of a product basis function defined by the Mth set of arc factors α^M, and c_M are the subspace expansion coefficients. We assume that the expansion vectors $\mathbf{X}(\alpha)$ are linearly independent. The quantity $\sigma^2(\alpha,\mathbf{c})$ therefore depends on the linear expansion coefficients c_M and on the nonlinear arc factors $\alpha \equiv \{\alpha_\mu^M; \mu = 1, ..., N_{\mathrm{arc}}, M = 1, ..., N_\alpha\}$. We use the index μ as a shorthand for the combination of the step and node index (d, j) of an individual arc in the Shavitt graph. The goal is to minimize the quantity $\sigma^2(\alpha,\mathbf{c})$ with respect to the linear and nonlinear parameters. The reference vector $\mathbf{v}^{\mathrm{ref}}$ is assumed to be fixed throughout this optimization procedure, and in most cases it will be normalized, $|\mathbf{v}^{\mathrm{ref}}| = 1$, but the equations in this section will accommodate the general case.

The first task we examine is the optimization of the linear coefficients c_M for a given set of vectors $\mathbf{X}$. To this end, we write

$$\begin{aligned}\sigma^2(\alpha,\mathbf{c}) &= (\mathbf{v} - \mathbf{v}^{\mathrm{ref}})^{\mathrm{T}}(\mathbf{v} - \mathbf{v}^{\mathrm{ref}}) \\ &= \mathbf{v}^{\mathrm{T}}\mathbf{v} - 2\mathbf{v}^{\mathrm{T}}\mathbf{v}^{\mathrm{ref}} + \mathbf{v}^{\mathrm{ref\,T}}\mathbf{v}^{\mathrm{ref}} \\ &= \mathbf{c}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{c} - 2\mathbf{c}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{v}^{\mathrm{ref}} + |\mathbf{v}^{\mathrm{ref}}|^2 \\ &= \mathbf{c}^{\mathrm{T}}\mathbf{S}\mathbf{c} - 2\mathbf{c}^{\mathrm{T}}\mathbf{z} + |\mathbf{v}^{\mathrm{ref}}|^2 \end{aligned} \tag{37}$$

with

$$\mathbf{S} \equiv \mathbf{S}(\alpha) = \mathbf{X}(\alpha)^{\mathrm{T}}\mathbf{X}(\alpha) \tag{38}$$

$$\mathbf{z} \equiv \mathbf{z}(\alpha) = \mathbf{X}(\alpha)^{\mathrm{T}}\mathbf{v}^{\mathrm{ref}}$$

Differentiation of $\sigma^2(\alpha,\mathbf{c})$ with respect to a coefficient c_K, setting the result to zero, and solving for the vector $\mathbf{c}$ give the unique optimal linear expansion coefficients.

$$\mathbf{c}^{\mathrm{opt}}(\alpha) = \mathbf{S}(\alpha)^{-1}\mathbf{z}(\alpha) \tag{39}$$

Because this relation holds for any set of linearly independent expansion vectors $\mathbf{X}(\alpha)$, it defines the dependence of $\mathbf{c}^{\mathrm{opt}}$ on the nonlinear parameters α. For these optimal linear coefficients, the following relations hold:

$$\mathbf{v}^{\mathrm{T}}\mathbf{v} = |\mathbf{v}|^2 = \mathbf{c}^{\mathrm{opt\,T}}\mathbf{S}\mathbf{c}^{\mathrm{opt}} = \mathbf{z}^{\mathrm{T}}\mathbf{S}^{-1}\mathbf{z} = \mathbf{z}^{\mathrm{T}}\mathbf{c}^{\mathrm{opt}} \tag{40}$$

$$\mathbf{v}^{\mathrm{T}}\mathbf{v}^{\mathrm{ref}} = \mathbf{z}^{\mathrm{T}}\mathbf{S}^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{v}^{\mathrm{ref}} = \mathbf{z}^{\mathrm{T}}\mathbf{S}^{-1}\mathbf{z} = \mathbf{v}^{\mathrm{T}}\mathbf{v} \tag{41}$$

$$\sigma^2(\alpha;\mathbf{c}^{\mathrm{opt}}(\alpha)) = |\mathbf{v}^{\mathrm{ref}}|^2 - \mathbf{z}(\alpha)^{\mathrm{T}}\mathbf{S}(\alpha)^{-1}\mathbf{z}(\alpha) \tag{42}$$

We note in passing that if θ is the angle between the vectors $\mathbf{v}$ and $\mathbf{v}^{\mathrm{ref}}$, then $\cos(\theta) = |\mathbf{v}|/|\mathbf{v}^{\mathrm{ref}}|$ and $\sin^2(\theta) = \sigma^2(\alpha;\mathbf{c}^{\mathrm{opt}})/|\mathbf{v}^{\mathrm{ref}}|^2$. This gives a geometrical picture of the optimization in terms of minimization of the angle θ.

We next address the minimization of $\sigma^2(\alpha) \equiv \sigma^2(\alpha;\mathbf{c}^{\mathrm{opt}})$ with respect to the nonlinear parameters α. There is no closed-form solution to this problem, as there is for the linear coefficients $\mathbf{c}^{\mathrm{opt}}$, so the arc factors must be optimized numerically. We seek values α^{opt} that satisfy the local condition

$$\left.\frac{\mathrm{d}\sigma^2(\alpha)}{\mathrm{d}\alpha_\mu^M}\right|_{\alpha^{\mathrm{opt}}} = 0 \qquad \text{for all } \mu \text{ and } M \tag{43}$$

We use a numerical approach to this optimization problem that requires function evaluations and gradient evaluations at arbitrary values of the parameters α. For this purpose, it is necessary to compute efficiently the function value, $\sigma^2(\alpha)$, along with the quantities $\mathrm{d}\sigma^2(\alpha)/\mathrm{d}\alpha_\mu^M$ for all of the arcs μ and product basis functions M. We outline below our approach to evaluate these quantities efficiently.

Computation of $\sigma^2(\alpha)$. We first examine the computation of the quantity $\sigma^2(\alpha)$. According to eq 42, the vector $\mathbf{z}$ and the matrix $\mathbf{S}$ are required. The vector $\mathbf{z}$ is computed as outlined in Figure 7. Each individual walk is generated on the Shavitt graph using a recursive tree-search algorithm (which we implement using a stack). As the walk is generated, the partial products of the α_μ^M arc factors are computed and stored in a stack indexed by the orbital level. As the tree-search algorithm reaches the graph head for each complete walk, the corresponding element of the vector $\mathbf{x}^M$ is available. With this approach, the effort per walk to compute a CSF coefficient decreases from a constant n down to an average value of about $\mathrm{Log}_4(n)$ for large expansions. This is because once one walk has been generated, much of the effort involved in computing the arc factor products for the next walk reuses the existing stack information. On the other hand, there is little additional overhead for using the tree-search algorithm to generate the walks even in the worst case situations in which most of the stack elements must be regenerated for each walk.

```
Set $z_{1:N_\alpha} = \mathbf{0}$
do $m$ = 1, $N_{csf}$
    Generate the walk corresponding to $m$ using
      the recursive tree-search algorithm.
    Compute $X_{m,1:N_\alpha}$ using the partial products
      in the stack
    Update $z_{1:N_\alpha} \leftarrow v^{ref}{}_m \cdot X_{m,1:N_\alpha}$
enddo $m$
```

Figure 7. Outline of the computation of of $\mathbf{z} = \mathbf{X}(\alpha)^{\mathrm{T}}\mathbf{v}^{\mathrm{ref}}$.

Once the linear expansion coefficients x_m^M are available for walk m, they are then multiplied by the corresponding element of $\mathbf{v}^{\mathrm{ref}}$ and accumulated into the z_M element of the vector $\mathbf{z}$. Some minor efficiency can be gained by computing the products corresponding to several M values simultaneously. The overall computational effort scales roughly as the product $N_{\mathrm{csf}}N_\alpha$ $\mathrm{Log}_4(n)$ (for large wave function expansions) because of the use of the recursive algorithm to generate the walks. The total memory required for this step consists of storage of the vector $\mathbf{v}^{\mathrm{ref}}$, and storage of the stacks involved with the walk generation and the arc factor products. For large expansions, or in a parallel environment, $\mathbf{v}^{\mathrm{ref}}$ can be split into segments of arbitrary size, and the procedure can be applied separately to each segment independently and result $\mathbf{z}$ can be globally summed at the end. Storage of the fully expanded, linearized expansion vectors $\mathbf{X}$ is not necessary. Because the elements of $\mathbf{v}^{\mathrm{ref}}$ are assumed to be independent quantities, it is difficult to imagine how the effort proportional to N_{csf} could be eliminated from this step.

The computation of $\mathbf{S}$ has already been discussed in section 2; this step does not require any effort proportional to N_{csf}. With the arrays $\mathbf{S}$ and $\mathbf{z}$ available, The quantity $\sigma^2(\alpha)$ can be computed in the straightforward manner: (1) solve the linear equation $\mathbf{S}\mathbf{c}^{\mathrm{opt}} = \mathbf{z}$, and then (2) compute $\sigma^2(\alpha) = |\mathbf{v}^{\mathrm{ref}}|^2 - \mathbf{z}^{\mathrm{T}}\mathbf{c}^{\mathrm{opt}}$.

Computation of $\mathrm{d}\sigma^2(\alpha)/\mathrm{d}\alpha_{\mathbf{n}}^P$. We now focus on the computation of the gradients that are used for the optimization process. From eq 42, two separate contributions to the gradient are required.

$$\begin{aligned} g_{(\nu P)} &= \frac{\mathrm{d}\sigma^2(\alpha;\mathbf{c}^{\mathrm{opt}})}{\mathrm{d}\alpha_\nu^P} \\ &= -2\mathbf{c}^{\mathrm{opt\,T}}\frac{\mathrm{d}\mathbf{z}(\alpha)}{\mathrm{d}\alpha_\nu^P} - \mathbf{z}^{\mathrm{T}}\frac{\mathrm{d}\mathbf{S}(\alpha)^{-1}}{\mathrm{d}\alpha_\nu^P}\mathbf{z} \\ &= g_{(\nu P)}^z + g_{(\nu P)}^S \end{aligned} \tag{44}$$

For brevity, we use the shorthand notation A' to denote $\mathrm{d}A/\mathrm{d}\alpha_\nu^P$ for some arbitrary quantity A. Differentiating the relation $\mathbf{S}\mathbf{S}^{-1}$

$= \mathbf{1}$ to arrive at the identity $(\mathbf{S}^{-1})' = -\mathbf{S}^{-1}\mathbf{S}'\mathbf{S}^{-1}$, allows the required gradient contributions to be written

$$\mathbf{g}^z = -2\mathbf{c}^{\text{opt T}}\mathbf{z}(\alpha)'$$
$$\mathbf{g}^S = \mathbf{c}^{\text{opt T}}\mathbf{S}(\alpha)'\mathbf{c}^{\text{opt}} \tag{45}$$

We assume that we have available $\mathbf{c}^{\text{opt}}$ at the time that the gradients are to be evaluated, and we attempt to contract together the $\mathbf{c}^{\text{opt}}$ coefficients directly into the gradient vector during the computation procedure. This reduces storage requirements compared to, for example, computing the matrix $\mathbf{S}'$ itself for all combinations of ν and P, and then subsequently contracting those arrays with the vector $\mathbf{c}^{\text{opt}}$ in the computation of $\mathbf{g}^S$.

We examine first the computation of the $\mathbf{g}^z$ terms.

$$g^z_{\nu P} = -2\sum_{M=1}^{N_\alpha} c_M^{opt}\, z(\alpha)'_M$$
$$= -2\sum_{M=1}^{N_\alpha}\sum_{m=1}^{N_{\text{csf}}} c_M^{\text{opt}}\, v_m^{\text{ref}}\, x_m^M(\alpha)' \tag{46}$$

The computation of $\mathbf{z}(\alpha)'$ parallels the computation of $\mathbf{z}(\alpha)$ itself. As the walk is constructed using the tree-search algorithm, the node and step number at each orbital level are stored in the appropriate stack arrays. As the graph head is reached for the walk corresponding to m, the x_m^M product is available

$$x_m^M = \prod_{j=0}^{n-1}\alpha^M_{\mu(j,m)} \tag{47}$$

$\mu(j,m)$ is the arc that corresponds to the step from the node at level j in walk m. The derivative of this coefficient is

$$\frac{dx_m^M}{d\alpha_\nu^P} = \delta_{MP}\delta_{\nu,\mu(i,m)}\prod_{j(\neq i)}^{n-1}\alpha^M_{\mu(j,m)} \qquad \text{for } i = \text{Level}(\nu) \tag{48}$$

Each x_m^M product therefore contributes to n distinct gradient entries.

$$g^z_{\mu(i,m),M} \leftarrow -2c_M^{\text{opt}}v_m^{\text{ref}}\prod_{j(\neq i)}^{n-1}\alpha^M_{\mu(j,m)} \qquad \text{for } i = 0, ..., (n-1) \tag{49}$$

The necessary products of the arc factors for $j \neq i$ may be computed efficiently from the partial product stack with only two floating point multiplications per gradient contribution. The overall effort scales approximately as the product $nN_{\text{csf}}N_\alpha$ $\text{Log}_4(n)$ (for large wave function expansions), which is only a factor of n larger than the computation of $\mathbf{z}$ itself. No significant additional storage is required other than the result gradient $\mathbf{g}^z$ itself. This procedure is outlined in Figure 8. As discussed previously, because the elements of $\mathbf{v}^{\text{ref}}$ are assumed to be independent quantities, it is difficult to imagine how the effort proportional to N_{csf} could be eliminated from this step.

```
do $m = 1, N_{csf}$
    Generate the walk corresponding to $m$ using
      the recursive tree-search algorithm.
    do $i = 0, (n-1)$
        Update $g_{\mu(i,m),M} \leftarrow -2c_M^{opt}v_m^{ref}\prod_{j(\neq i)}^{n-1}\alpha^M_{\mu(j,m)}$ for all $M$.
    enddo $i$
enddo $m$
```

Figure 8. Outline of the computation of the $\mathbf{g}^z$ gradient contributions.

```
do $M = 1, N_\alpha$
    do $N = 1, M$
        compute $\boldsymbol{\gamma}^{MN}$ and $\bar{\boldsymbol{\gamma}}^{MN}$
        do $\nu = 1, N_{arc}$
            set $\beta = 2c_M^{opt}c_N^{opt}\bar{\gamma}^{MN}_{Top(\nu)}\gamma^{MN}_{Bottom(\nu)}$
            update $g_{\nu,M} \leftarrow \alpha_\nu^N\beta$
            if $(M \neq N)$ update $g_{\nu,N} \leftarrow \alpha_\nu^M\beta$
        enddo $\nu$
    enddo $N$
enddo $M$
```

Figure 9. Outline of the computation of $\mathbf{g}^S$ gradient contributions.

We next consider the gradient contributions $\mathbf{g}^S$. For this purpose we use the general expression given in eq 27 to compute an element S_{MN}; this expression reveals explicitly the dependence on an individual arc factor α_ν^P. Differentiation of S_{MN} with respect to an arc factor α_ν^P results in the gradient expression

$$g^S_{\nu P} = \sum_{M,N} c_M^{\text{opt}}c_N^{\text{opt}}S'_{MN}$$
$$= \sum_{M,N} c_M^{\text{opt}}c_N^{\text{opt}}\,\bar{\gamma}^{MN}_{\text{Top}(\nu)}\,\gamma^{MN}_{\text{Bottom}(\nu)}\frac{\text{d}}{\text{d}\alpha_\nu^P}(\alpha_\nu^M\alpha_\nu^N)$$
$$= \sum_{M,N} c_M^{\text{opt}}c_N^{\text{opt}}\,\bar{\gamma}^{MN}_{\text{Top}(\nu)}\,\gamma^{MN}_{\text{Bottom}(\nu)}(\alpha_\nu^M\,\delta_{NP} + \alpha_\nu^N\,\delta_{MP}) \tag{50}$$

The entire gradient $\mathbf{g}^S$ may be computed by looping over all of the arcs in the Shavitt graph and accumulating the above contributions over all product term pairs M, N. Figure 9 summarizes this procedure assuming the contributions from a single M, N pair are constructed at a time. The construction of γ^{MN} and $\bar{\gamma}^{MN}$ scales as N_{row} for each M, N pair. The computational effort for $\mathbf{g}^S$ is proportional to N_{arc} for each M, N pair, so the total effort scales as the product $N_{\text{arc}}N_\alpha(N_\alpha + 1)$. Storage requirements include the γ^{MN} and $\bar{\gamma}^{MN}$ arrays for all of the M, N pairs of interest, along with the result gradient $\mathbf{g}$. There is no computational effort or storage requirements that depends explicitly on N_{csf} for these terms.

Derivatives with Respect to Essential Variables. In the above discussion, we have assumed that the arc factors α are the fundamental independent variables of interest. For optimization of $\sigma^2 = |\mathbf{v} - \mathbf{v}^{\text{ref}}|^2$ it is more convenient, robust, and efficient to perform the optimization directly in terms of the essential variables φ defined in eqs 13. This requires gradients with respect to these essential variables. For this purpose we write $\sigma^2(\varphi) \equiv \sigma^2(\alpha(\varphi))$ and apply the chain rule to give

$$\begin{aligned}\tilde{g}_{rM} &\equiv \frac{d\sigma^2(\varphi)}{d\varphi_r^M}\\ &= \sum_{\mu N}\frac{d\sigma^2(\alpha)}{d\alpha_\mu^N}\frac{d\alpha_\mu^N}{d\varphi_r^M}\\ &= \sum_{\mu N} g_{\mu N}\frac{d\alpha_\mu^N}{d\varphi_r^M} \qquad (51)\end{aligned}$$

The gradient elements $g_{\mu N}$ computed using the procedures discussed above may be transformed to produce the desired gradient elements $\tilde{g}_{rM}$. In this manner, the above discussion of the efficient function evaluation and gradient evaluation in terms of the arc factors α applies directly to the computation of functions and gradients in terms of the essential variables φ. According to the mapping in eq 13, the $\alpha_\mu(\varphi)$ arc factors are analytic (i.e., continuous and smooth), so the essential variables φ may be considered as unconstrained independent variables for the normalized wave function. For a node j with $\eta_j > 1$ lower arcs, the transformation elements are given by

$$\frac{d\alpha^N_{d(1),j}}{d\varphi^M_{q,j'}} = -\delta_{MN}\delta_{jj'}\sin(r)\varphi^M_{q,j}/r$$

$$\frac{d\alpha^N_{d(p+1)j}}{d\varphi^M_{q,j'}} = \delta_{MN}\delta_{jj'}\left(\delta_{q,p}\frac{\sin(r)}{r} + \varphi^M_{q,j}\varphi^M_{p,j}\left(\frac{\cos(r)}{r^2} - \frac{\sin(r)}{r^3}\right)\right)$$

$$\text{for } p = 1, ..., (\eta_j - 1) \qquad (52)$$

This transformation is very sparse and localized within single product functions (δ_{MN}) and also to a single node at a time ($\delta_{jj'}$), and from a practical perspective it takes very little effort.

References and Notes

(1) Shavitt, I. *Int. J. Quantum Chem.* **1977**, *S11*, 131.

(2) Shavitt, I. *Int. J. Quantum Chem.* **1978**, *S12*, 5.

(3) Shavitt, I. *New Methods in Computational Quantum Chemistry and Their Application on Modern Super-Computers*; Annual Report to the National Aeronautics and Space Administration; Battelle Columbus Laboratories: Columbus, OH, June 1979.

(4) Lischka, H.; Shepard, R.; Brown, F. B.; Shavitt, I. *Int. J. Quantum Chem.* **1981**, *S15*, 91.

(5) Shavitt, I. The Graphical Unitary Group Approach and its Application to Direct Configuration Interaction Calculations. In *The Unitary Group for the Evaluation of Electronic Energy Matrix Elements*; Hinze, J., Ed.; Lecture Notes in Chemistry 22; Springer-Verlag: Berlin, 1981.

(6) Shavitt, I. The Unitary Group and the Electron Correlation Problem. In *New Horizons of Quantum Chemistry*; Lowdin, P.-O., Pullman, B., Eds.; D. Reidel: Dordrecht, The Netherlands, 1983; pp 279−293.

(7) Shavitt, I. *Unitary Group Approach to Configuration Interaction Calculations of the Electronic Structure of Atoms and Molecules*, *in Mathematical Frontiers in Computational Physics*, Truhlar, D. G. Ed.; Springer-Verlag: New York, 1988.

(8) Shepard, R.; Simons, J. *Int. J. Quantum Chem.* **1980**, *S14*, 211.

(9) Shepard, R. *The Multiconfiguration Self-Consistent Field Method*. In Ab Initio Methods in Quantum Chemistry II; Advances in Chemical Physics No. 69; Lawley, K. P., Ed.; Wiley: New York, 1987; pp 63−200.

(10) Shepard, R.; Shavitt, I.; Pitzer, R. M.; Comeau, D. C.; Pepper, M.; Lischka, H.; Szalay, P. G.; Ahlrichs, R.; Brown, F. B.; Zhao, J.-G. *Int J. Quantum Chem*. **1988**, *S22*, 149.

(11) Lischka, H.; Shepard, R.; Pitzer, R. M.; Shavitt, I.; Dallos, M.; Muller, T.; Szalay, P. G.; Seth, M.; Kedziora, G. S.; Yabushita, S.; Zhang, Z. *Phys. Chem. Chem. Phys.* **2001**, *3*, 664.

(12) Shepard, R. An Introduction to GUGA in the COLUMBUS Program System. In *Relativistic and Electron Correlation Effects in Molecules and Solids*; Malli, G., Ed.; Plenum: New York, 1994; pp 447−460.

(13) Paldus, J. *J. Chem. Phys.* **1974**, *61*, 5321.

(14) Paldus, J. In *Theoretical Chemistry: Advances and Perspectives*; Eyring, H., Henderson, D. J., Eds.; Academic: New York, 1976; Vol. 2, p 131.

(15) Paldus, J.; Boyle, M. *Phys. Scr.* **1980**, *21*, 295.

(16) Shepard, R. The Analytic Gradient Method for Configuration Interaction Wave Functions. In *Modern Electronic Structure Theory Part I*; Yarkony, D. R., Ed.; World Scientific: Singapore, 1995; pp 345−458.

(17) Shavitt, I. *Chem. Phys. Lett.* **1979**, *63*, 421.

(18) Duch, W. *J. Phys. A: Math. Gen.* **1985**, *18*, 3283.

(19) Harding, L. B.; Goddard, W. A., III. *J. Am. Chem. Soc.* **1975**, *97*, 6293.

(20) Hager, W. W.; Zhang, H. http://www.math.ufl.edu/∼hager/papers/CG.