Hamiltonian Matrix and Reduced Density Matrix Construction with Nonlinear Wave Functions

Ron Shepard*

Chemistry Division, Argonne National Laboratory, Argonne, Illinois 60439

Received: October 25, 2005; In Final Form: April 19, 2006

An efficient procedure to compute Hamiltonian matrix elements and reduced one- and two-particle density matrices for electronic wave functions using a new graphical-based nonlinear expansion form is presented. This method is based on spin eigenfunctions using the graphical unitary group approach (GUGA), and the wave function is expanded in a basis of product functions (each of which is equivalent to some linear combination of all of the configuration state functions), allowing application to closed- and open-shell systems and to ground and excited electronic states. In general, the effort required to construct an individual Hamiltonian matrix element between two product basis functions $H_{MN} = \langle M|\hat{H}|N\rangle$ scales as $\mathcal{O}(\beta n^4)$ for a wave function expanded in n molecular orbitals. The prefactor β itself scales between N^0 and N^2, for N electrons, depending on the complexity of the underlying Shavitt graph. Timings with our initial implementation of this method are very promising. Wave function expansions that are orders of magnitude larger than can be treated with traditional CI methods require only modest effort with our new method.

1. Introduction

In previous work,[1] we introduced a computational method based on a new expansion basis form for electronic wave functions. This method is based on the graphical unitary group approach (GUGA) of Shavitt,[2−8] which utilizes a graphical representation of the unitary group approach (UGA) of Paldus.[9−11] This new approach is intended to be used in MCSCF[5,12−14] and configuration interaction[2−11] (CI) wave functions, and it is being developed within the COLUMBUS Program System,[14−16] the main emphasis of which is the accurate computation of global potential energy surfaces of ground and excited states. The wave function is expanded in a basis of product functions, and each product function depends on a relatively small number of nonlinear parameters but is equivalent to some linear combination of all the configuration state functions (CSFs). In the previous work, we developed recursive procedures to compute efficiently the overlap (scalar product) between two product basis functions $S_{MN} = \langle M|N\rangle$. In the present work, we extend that same recursive approach to efficiently compute Hamiltonian matrix elements $H_{MN} = \langle M|\hat{H}|N\rangle$, transition one-particle reduced density matrix elements D_{pq}^{MN}, and transition two-particle reduced density matrix elements d_{pqrs}^{MN}. From these quantities, ground and excited electronic state energies may be computed along with the expectation values of other arbitrary one- and two-electron operators. In general, the effort required to construct an individual Hamiltonian matrix element between two product basis functions $H_{MN} = \langle M|\hat{H}|N\rangle$ scales as $\mathcal{O}(\beta n^4)$ for a wave function expanded in n molecular orbitals. The prefactor β itself scales between N^0 and N^2, for N electrons, depending on the complexity of the underlying Shavitt graph.

2. Method

We summarize briefly the method and notation that was introduced in ref 1. Each *node*, indexed by j, of a *Shavitt graph* corresponds to an integer triple (a_j, b_j, c_j) of a row of a *Paldus ABC tableau*.[9−11] These integers are related to the number of orbitals n_j, the number of electrons N_j, and the spin quantum number S_j according to

$$N_j = 2a_j + b_j$$
$$S_j = b_j/2$$
$$n_j = a_j + b_j + c_j \tag{1}$$

Each node j thereby corresponds to an $\hat{S}^2$ spin eigenfunction with eigenvalue $S_j(S_j + 1)$, to a specific number of electrons N_j, and to a subspace of the orbitals of dimension n_j. 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 the molecule of interest. The nodes at one level are connected with *arcs* (or *steps*) to the nodes at the adjacent levels. There are four possible step numbers that connect the nodes, denoted by the integer d with $0 \le d \le 3$. 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 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 (the *step vector*) in that walk. In a typical Shavitt graph, an individual node may be touched by many walks, so it is convenient to organize the graph on the basis of storage of the nodes; the

* shepard@tcg.anl.gov.

10.1021/jp060336g CCC: $33.50 © 2006 American Chemical Society
Published on Web 06/16/2006

TABLE 1: Characterization of Step Numbers[a]

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

[a] $\Delta x = x_{p+1} - x_p$ for $x = a$, b, c, N, and S at level p.

storage of the connecting arcs, and other information discussed below associated with that node, is called a *distinct row table* (DRT). Each CSF may be assigned a contiguous integer index m with $1 \le m \le N_{\mathrm{csf}}$ which may be computed as a summation of the integer *arc weights* that are associated with the arcs.

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

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. 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 determine the integer CSF index m from a given step vector.

In a *product basis function*, a numerical *arc factor* is assigned to each of the arcs in a given Shavitt graph. These arc factors are 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. The CSF coefficient x_m associated with a particular walk m is taken as the product of the arc factors in that walk. That is, in analogy to eq 2

$$x_m = \prod_{p=0}^{n-1} \alpha_{d_{p,m} j_{p,m}} = \prod_{p=0}^{n-1} \alpha_{\mu(p,m)} \qquad (3)$$

Because 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. A product basis function, denoted $|M\rangle$, is then defined in terms of these CSFs $|\tilde{m}\rangle$ and their coefficients x_m^M as

$$|M\rangle = \sum_{m=1}^{N_{\mathrm{csf}}} x_m^M |\tilde{m}\rangle \qquad (4)$$

and thus corresponds to a particular linear combination of all the CSFs in the linear expansion basis. The product function $|M\rangle$ is a linear combination of spin eigenfunctions; therefore, it also is an eigenfunction of $\hat{S}^2$. There can be several sets of arc factors, each of which is associated with a corresponding product function through eqs 3 and 4. The mapping of a set of arc factors to the vector of CSF coefficients will be denoted as $\mathbf{x}^M \equiv \mathscr{L}(\alpha^M)$. This is a many-to-one mapping, because more than one set of arc factors α map to the same CSF coefficient vector $\mathbf{x}$. This lack of uniqueness may be eliminated[1] by introducing an arc phase and normalization convention based on the lower walk *partial product functions* that are associated with each node of the Shavitt graph. A set of arc factors α^M that satisfy this normalization convention is in *standard form*, and such a set of arc factors may be characterized by a smaller number of essential variational parameters φ^M. The number of these essential variables[1] is given by $N_\varphi = N_{\mathrm{arc}} - \mathrm{N}_{\mathrm{row}} + 1$ where N_{arc} and N_{row} are the number of arcs and nodes, respectively, in the Shavitt graph. Table 2 shows the relation between N_{csf}, N_{row}, and N_φ for a set of Shavitt graphs that correspond to singlet full-CI calculations with $n = N$ ranging from $n = 2$ to $n = 46$. The goal of the product function approach is to make as much as possible of the effort that is related to wave function manipulation, interpretation, optimization, and storage depend on N_{row} rather than the much larger number N_{csf}.

A single product function $|M\rangle$ is itself equivalent to a complicated linear combination of CSFs, and it is capable of describing molecular bond dissociation processes, spin recoupling, and electron correlation, and it has other interesting features.[1] A single product function has sufficient flexibility, through its associated arc factors, to cover the entire set of CSFs within the underlying linear expansion space.[1] However, a single product function cannot represent an arbitrary vector (i.e., an arbitrary linear combination of CSFs) within that space. To allow for additional flexibility, we define a general linear combination wave function as

$$|\psi\rangle = \sum_{M}^{N_\alpha} c_M |M\rangle \qquad (5)$$

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

$$\mathbf{Hc} = \mathbf{Sc}E \qquad (6)$$

with $H_{MN} = \langle M|\hat{H}|N\rangle$ and $S_{MN} = \langle M|N\rangle$. In this approach, $\mathbf{H}$ and $\mathbf{S}$ are roughly analogous to the corresponding subspace matrices in a Lanczos or Davidson type of iterative procedure[17] used commonly in traditional CI approaches. The efficient computation of the metric matrix $\mathbf{S}$ has been described previously;[1] this is achieved using a recursive procedure with effort that scales as N_{row} and that does not depend on N_{csf}. The computation of the Hamiltonian matrix $\mathbf{H}$ in the product function basis is the subject of the present work, and we seek a similar recursive algorithm with effort that does not depend on N_{csf}. Through the Ritz variational principle, the lowest eigenvalues computed from the product function basis in eq 6 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, this general approach is applicable to both ground and excited electronic states. As the expansion set N_α increases, the eigenvalues from eq 6 are bounded from below only by the eigenvalues of the underlying CSF space,[1] that is, the linear combination product function form does not represent an inherent formal limitation on the accuracy of the computed wave function. The wave function $|\psi\rangle$ is a spin eigenfunction because the product basis functions $|M\rangle$ are eigenfunctions of $\hat{S}^2$; therefore, our method does not suffer from spin contamination or spin instabilities.

If $\mathbf{x}^M = \mathscr{L}(\alpha^M)$ and $\mathbf{x}^N = \mathscr{L}(\alpha^N)$ are the vectors of CSF coefficients of the product functions defined by α^M and α^N, then $H_{MN} = (\mathbf{x}^M)^T\hat{\mathbf{H}}\mathbf{x}^N$ (where $\hat{\mathbf{H}}$ is the Hamiltonian matrix in the CSF basis) is the matrix element of interest. One way to compute this quantity would be to construct the vectors $\mathbf{x}^M$ and $\mathbf{x}^N$ explicitly, and to compute the matrix−vector and scalar products directly from these expanded vectors using traditional CI technology. Our approach discussed below requires both less computational effort and less storage for large expansions than this straightforward approach.

TABLE 2: Statistics for Singlet Full-CI Wave Function Expansions

$n = N$	N_{csf}	N_{row}	N_{φ}	N_{pair} [a]	N_{value} [b]	time [c]
2	3	5	2	10	26	0.00
4	20	14	13	43	278	0.00
6	175	30	39	120	1058	0.00
8	1 764	55	86	261	2682	0.00
10	19 404	91	160	486	5466	0.00
12	226 512	140	267	815	9726	0.00
14	2 760 615	204	413	1268	15778	0.01
16	34 763 300	285	604	1865	23938	0.04
18	449 141 836	385	846	2626	34522	0.07
20	5 924 217 936	506	1145	3571	47846	0.13
22	79 483 257 308	650	1507	4720	64226	0.21
24	1 081 724 803 600	819	1938	6093	83978	0.34
26	14 901 311 070 000	1015	2444	7710	107418	0.54
28	207 426 250 094 400	1240	3031	9591	134862	0.82
30	2 913 690 606 794 775	1496	3705	11756	166626	1.21
32	41 255 439 318 353 700	1785	4472	14225	203026	1.75
34	588 272 005 095 043 500	2109	5338	17018	244378	2.49
36	8 441 132 926 294 530 000	2470	6309	20115	290998	3.46
38	121 805 548 126 430 067 900	2870	7391	23656	343202	4.66
40	1 766 594 752 418 700 032 400	3311	8590	27541	401306	6.27
42	25 739 723 541 439 406 257 200	3795	9912	31830	465626	8.25
44	376 607 675 256 599 252 232 000	4324	11363	36543	536478	11.19
46	5 531 425 230 331 301 517 157 500	4900	12949	41700	614178	14.43

[a] N_{pair} is the total number of pairs of nodes in the Shavitt graph (that contribute to upper-triangular Shavitt loops) and the number of vertices in the auxiliary pair graph. [b] N_{value} is the total number of segment values. [c] Times are in seconds on a 2.5 GHz PowerMac G5 (PPC 970) to construct a single $\langle M|\hat{H}|N\rangle$ matrix element involving two product functions $|M\rangle$ and $|N\rangle$.

A Hamiltonian matrix element in the product function basis may be written using standard GUGA notation as

$$\begin{aligned} H_{MN} &= \langle M|\hat{H}|N\rangle \\ &= \sum_{m,n} x_m^M\, x_n^N \langle \tilde{m}|\hat{H}|\tilde{n}\rangle \\ &= \sum_{m,n} x_m^M\, x_n^N \left(\sum_{p,q} h_{pq} \langle \tilde{m}|\hat{E}_{pq}|\tilde{n}\rangle + \frac{1}{2} \sum_{p,q,r,s} g_{pqrs} \langle \tilde{m}|\hat{e}_{pqrs}|\tilde{n}\rangle \right) \end{aligned} \qquad (7)$$

h_{pq} and g_{pqrs} are the one- and two-electron Hamiltonian integrals indexed by the molecular orbital indices *p*, *q*, *r*, and *s*. The operators $\hat{E}_{pq}$ and $\hat{e}_{pqrs} \equiv \hat{E}_{pq}\hat{E}_{rs} - \delta_{rq}\hat{E}_{ps}$ are the generators and the normal-order generator products. The matrix elements $\langle \tilde{m}|\hat{E}_{pq}|\tilde{n}\rangle$ and $\langle \tilde{m}|\hat{e}_{pqrs}|\tilde{n}\rangle$ are the one- and two-electron coupling coefficients, and in traditional MCSCF and CI approaches,[2-18] these are the primary quantities of interest in computing matrix-vector products. The commutation relation $[\hat{E}_{pq},\hat{E}_{rs}] = \delta_{rq}\hat{E}_{ps} - \delta_{ps}\hat{E}_{rq}$ results in the operator identity $\hat{e}_{pqrs} = \hat{e}_{rspq}$.

In the current approach, we write the Hamiltonian matrix element as

$$\begin{aligned} H_{MN} &= \sum_{p,q} h_{pq} \Big(\sum_{m,n} x_m^M\, x_n^N \langle \tilde{m}|\hat{E}_{pq}|\tilde{n}\rangle\Big) + \\ &\qquad \frac{1}{2} \sum_{p,q,r,s} g_{pqrs} \Big(\sum_{m,n} x_m^M\, x_n^N \langle \tilde{m}|\hat{e}_{pqrs}|\tilde{n}\rangle\Big) \\ &= \sum_{p,q} h_{pq} \langle M|\hat{E}_{pq}|N\rangle + \frac{1}{2} \sum_{p,q,r,s} g_{pqrs} \langle M|\hat{e}_{pqrs}|N\rangle \\ &= \sum_{p,q} h_{pq} D_{pq}^{MN} + \frac{1}{2} \sum_{p,q,r,s} g_{pqrs} d_{pqrs}^{MN} \\ &= \mathrm{Tr}(\mathbf{hD}^{MN}) + \frac{1}{2}\mathrm{Tr}(\mathbf{gd}^{MN}) \end{aligned} \qquad (8)$$

Because of the index symmetry of the one- and two-electron integrals (i.e., $h_{pq} = h_{qp}$, $g_{pqrs} = g_{pqsr}$, and $g_{pqrs} = g_{rspq}$ with all quantities assumed to be real), we may choose to use in eq 8 the symmetrized one- and two-particle transition density matrices, which are defined as

$$\begin{aligned} D_{pq}^{MN} &= \frac{1}{2} \sum_{m,n} x_m^M\, x_n^N \langle \tilde{m}|\hat{E}_{pq} + \hat{E}_{qp}|\tilde{n}\rangle = \frac{1}{2}\langle M|\hat{E}_{pq} + \hat{E}_{qp}|N\rangle \\ d_{pqrs}^{MN} &= \frac{1}{4} \sum_{m,n} x_m^M\, x_n^N \langle \tilde{m}|\hat{e}_{pqrs} + \hat{e}_{pqsr} + \hat{e}_{qprs} + \hat{e}_{qpsr}|\tilde{n}\rangle \\ &= \frac{1}{4}\langle M|\hat{e}_{pqrs} + \hat{e}_{pqsr} + \hat{e}_{qprs} + \hat{e}_{qpsr}|N\rangle \end{aligned} \qquad (9)$$

These expressions define the normalization and indexing conventions of the density matrices in this work (see also refs 13 and 18). The adjoint operator identity $\hat{E}_{pq} = \hat{E}_{qp}^{\dagger}$ results in the identities $\mathbf{D}^{MN} = \mathbf{D}^{NM}$ and $\mathbf{d}^{MN} = \mathbf{d}^{NM}$. Given the one- and two-particle transition density matrix elements, it is straightforward to combine these quantities with the appropriate Hamiltonian integrals to compute the matrix element H_{MN}. Given the expansion coefficients for two states $|\psi_I\rangle$ and $|\psi_J\rangle$ from eqs 5 and 6, the state transition density matrices may be computed as

$$\begin{aligned} D_{pq}^{IJ} &= \frac{1}{2}\langle \psi_I|\hat{E}_{pq} + \hat{E}_{qp}|\psi_J\rangle = \sum_{M,N} c_M^I\, c_N^J\, D_{pq}^{MN} \\ d_{pqrs}^{IJ} &= \frac{1}{4}\langle \psi_I|\hat{e}_{pqrs} + \hat{e}_{pqsr} + \hat{e}_{qprs} + \hat{e}_{qpsr}|\psi_J\rangle = \sum_{M,N} c_M^I\, c_N^J\, d_{pqrs}^{MN} \end{aligned} \qquad (10)$$

From these quantities, arbitrary expectation values ($I = J$) and transition properties ($I \neq J$) may be computed. Thus, the transition reduced density matrices $\mathbf{D}^{MN}$ and $\mathbf{d}^{MN}$ in the product function basis are the fundamental quantities in this approach. We now focus on the efficient computation of these reduced density elements.

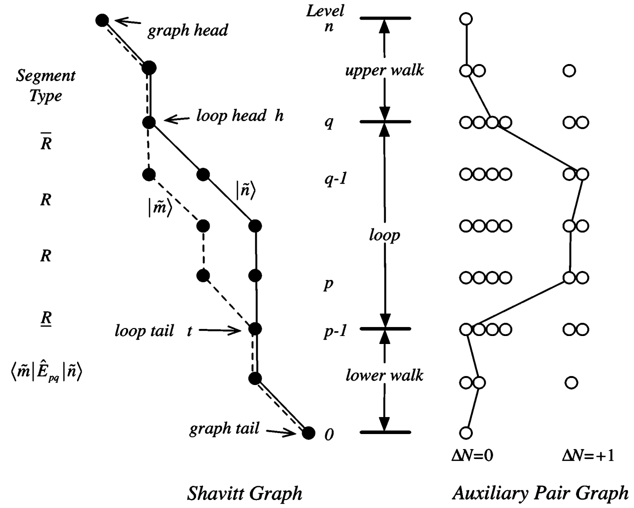

Figure 1. The graphical representation is shown for two walks $|\tilde{m}\rangle$ and $|\tilde{n}\rangle$ and the associated coupling coefficient $\langle \tilde{m}|\hat{E}_{pq}|\tilde{n}\rangle$. The representation on the left is the Shavitt graph, and the representation on the right is the auxiliary pair graph. In the Shavitt graph, the bra walk $|\tilde{m}\rangle$ is represented with the dotted lines, and the ket $|\tilde{n}\rangle$ is represented with the solid lines. The two walks coincide in the lower walk region between the graph tail and the loop tail t, and they coincide in the upper walk region between the loop head h and the graph head. The coupling coefficient depends only on the middle section where the walks differ, which constitutes the Shavitt loop, and its value is the product of the segment values within the loop range. In the pair graph, each vertex corresponds to a pair of nodes at the same level in the Shavitt graph, and the bra and ket walk pair in the Shavitt graph is represented as a single path in the node-pair graph. Each edge in the node-pair graph corresponds to a segment of the Shavitt graph, which in turn consists of the bra arc and ket arc pair that connects the bra and ket nodes at one level to the bra and ket nodes at the next higher level. The vertices of the pair graph are grouped according to the occupation difference $\Delta N = N_{\text{bra}} - N_{\text{ket}}$.

In the GUGA approach, coupling coefficients are computed from *Shavitt loops*. See Figure 1 for an example of a Shavitt loop and its corresponding one-particle coupling coefficient. CSFs $|\tilde{m}\rangle$ and $|\tilde{n}\rangle$ correspond to two walks from the tail to the head of the Shavitt graph. In general, there is a common section at the bottom of the graph in which the two walks share the same arcs and nodes, there is a middle section in which the two walks differ, and there is a common section at the top of the graph where the two walks again share the same arcs and nodes. The middle section is called the Shavitt loop, the bottom section is called the *lower walk* of the Shavitt loop, and the upper section is called the *upper walk* of the Shavitt loop. The node at the top of the Shavitt loop at level q, denoted h in Figure 1, is called the *loop head*, and the node at the bottom of the Shavitt loop at level $(p - 1)$, denoted t, is called the *loop tail*. One of the important features of the GUGA approach is that the coupling coefficient value depends only on the Shavitt loop and does not depend on the upper and lower walks, even in situations in which those partial walks contain singly or doubly occupied orbitals. For the typical Shavitt graphs that occur in MCSCF and CI calculations, there are relatively few Shavitt loops, out of the very large total number possible, that have nonzero loop values. In traditional CI approaches based on GUGA, the emphasis is therefore focused on the efficient construction of the individual Shavitt loops that have nonzero loop values and on the efficient organization of this loop construction, so that the computed coupling coefficients can be combined efficiently with the corresponding one- and two-electron Hamiltonian integrals. A Shavitt loop consists of a sequence of *segments*, which are the bra and ket steps at a given level within the loop. The 16 possible *segment shapes* (the combinations of the 4 bra steps and 4 ket steps) are listed in Table 3. The coupling coefficient value may be computed as a product of *segment values* (or in some cases as a linear combination of 2 such products); this property is important to the present method. Figure 2 shows schematically all of the unique loop shapes that are used in our procedure. This figure is based on Figure 8 in ref 6. The schematic segments that are shown are indicative only of the occupation difference between the bra and the ket segments of the loop, not of specific actual segment shapes. The segment value types associated with each level of each loop shape are indicated next to the loop diagram in Figure 2. The bra branch of each loop shape is indicated with a dotted line, and the ket branch is indicated with a solid line. If the bra and ket branches are exactly the same over some orbital range, then a single line is shown in the loop shape figure with a D (diagonal) segment value type (which is shorthand for the segment type RL^0). If the bra and ket branches are not necessarily exactly the same, but rather correspond to $\Delta N = 0$ and $\Delta\Delta N = 0$ (see Table 3), then the bra and ket branches are drawn close together but are distinct.

TABLE 3: Segment Shapes Categorized by Occupation Change[a]

$\Delta\Delta N$	$(d_{\text{bra}}, d_{\text{ket}})$	ΔN_p	segment value types[b]
−2	(0, 3)	−1	$\bar{R}\underline{L}$
		0	$\overline{RR}^0$
−1	(2, 3), (1, 3), (0, 2), (0, 1)	−1	$\underline{L}$
		0	$\bar{R}$, $W\bar{R}$, $R\underline{L}^1$
		+1	$R\bar{R}^0$, $R\bar{R}^1$
0	(3, 3), (2, 2), (2, 1), (1, 2), (1, 1), (0, 0)	−1	L
		0	W, WW, $\overline{RL}^1$,$\underline{RL}^1$, RL^0, RL^1
		+1	R, WR, $\underline{R}\bar{R}$
		+2	RR^0, RR^1
+1	(3, 2), (3, 1), (2, 0), (1, 0)	0	$\bar{L}$, $\underline{R}L^1$
		+1	$\underline{R}$, $\underline{R}W$, $R\bar{L}^1$
		+2	$R\underline{R}^0$, $R\underline{R}^1$
+2	(3, 0)	+2	$\underline{RR}^0$

[a] $\Delta N = N_{\text{bra}} - N_{\text{ket}}$; $\Delta\Delta N = \Delta N_p - \Delta N_{p-1}$. [b] The superscripts correspond to the $x = 0$ or $x = 1$ spin recoupling index.

Given a Shavitt loop T and its loop value $\mathcal{W}(T)$, the upper and lower walks are then generated only in order to determine the set of CSF indices m and n which share that loop value. That is, $\langle \tilde{m}|\hat{E}_{pq}|\tilde{n}\rangle = \mathcal{W}(T)$ (or $\langle \tilde{m}|\hat{e}_{pqrs}|\tilde{n}\rangle = \mathcal{W}(T)$, as appropriate) for all CSFs m and n associated with that Shavitt loop. Figure 3 outlines the traditional GUGA procedure for the direct computation of the contributions from a single Shavitt loop to a matrix−vector product $\mathbf{w} = \hat{\mathbf{H}}\mathbf{x}$ in the CSF basis. The difference $(m - n)$ of these walk indices depends only on the arc weights of the Shavitt loop and is therefore the same for all possible upper and lower walks. Furthermore, given, for example, a set of arc weights computed with a forward lexical scheme, all of the lower walks associated with a given upper walk will occur within a contiguous sequence of index values, the number of which is the *lower walk node weight*, $\underline{x}_t$. The total contributions to a Hamiltonian matrix in the CSF basis, from a particular Shavitt loop, therefore consists of a number of codiagonal sequences of equal value and all of length $\underline{x}_t$ and the number of these contiguous sequences is the *upper walk node weight* $\bar{x}_h$. The total number of matrix contributions from a particular Shavitt loop is given by the product $(\underline{x}_t\bar{x}_h)$, and this

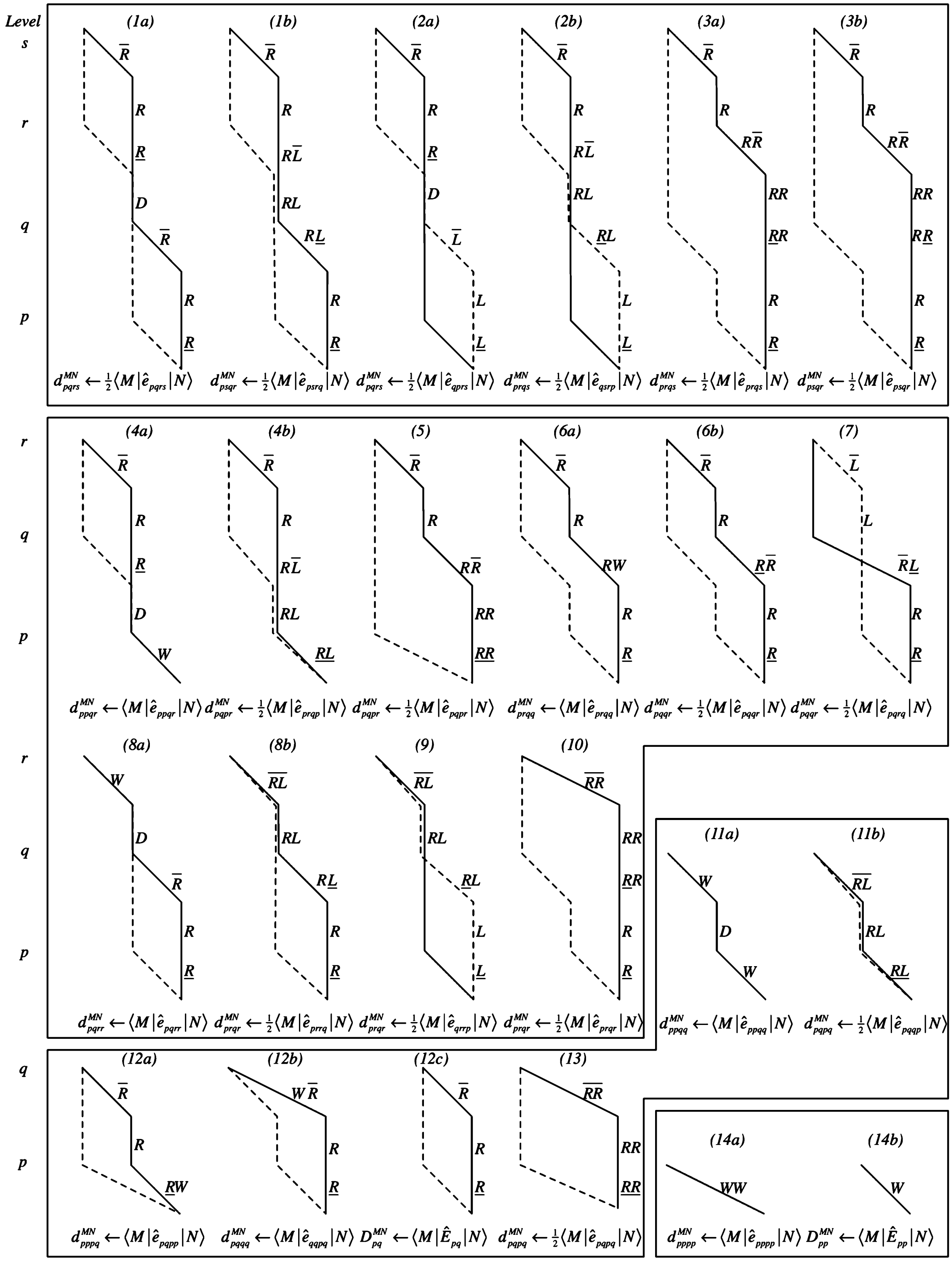
Level
s
r
q
p
(1a)
$\bar{R}$
R
$\underline{R}$
D
$\bar{R}$
R
$\underline{R}$
$d^{MN}_{pqrs} \leftarrow \frac{1}{2}\langle M|\hat{e}_{pqrs}|N\rangle$
(1b)
$\bar{R}$
R
$R\bar{L}$
RL
$R\underline{L}$
R
$\underline{R}$
$d^{MN}_{psqr} \leftarrow \frac{1}{2}\langle M|\hat{e}_{psrq}|N\rangle$
(2a)
$\bar{R}$
R
$\underline{R}$
D
$\bar{L}$
L
$\underline{L}$
$d^{MN}_{pqrs} \leftarrow \frac{1}{2}\langle M|\hat{e}_{qprs}|N\rangle$
(2b)
$\bar{R}$
R
$R\bar{L}$
RL
$\underline{R}L$
L
$\underline{L}$
$d^{MN}_{prqs} \leftarrow \frac{1}{2}\langle M|\hat{e}_{qsrp}|N\rangle$
(3a)
$\bar{R}$
R
$R\bar{R}$
RR
$\underline{R}R$
R
$\underline{R}$
$d^{MN}_{prqs} \leftarrow \frac{1}{2}\langle M|\hat{e}_{prqs}|N\rangle$
(3b)
$\bar{R}$
R
$R\bar{R}$
RR
$R\underline{R}$
R
$\underline{R}$
$d^{MN}_{psqr} \leftarrow \frac{1}{2}\langle M|\hat{e}_{psqr}|N\rangle$
r
q
p
(4a)
$\bar{R}$
R
$\underline{R}$
D
W
$d^{MN}_{ppqr} \leftarrow \langle M|\hat{e}_{ppqr}|N\rangle$
(4b)
$\bar{R}$
R
$R\bar{L}$
RL
$\underline{RL}$
$d^{MN}_{pqpr} \leftarrow \frac{1}{2}\langle M|\hat{e}_{prqp}|N\rangle$
(5)
$\bar{R}$
R
$\underline{R}\bar{R}$
RR
$\underline{RR}$
$d^{MN}_{pqpr} \leftarrow \frac{1}{2}\langle M|\hat{e}_{pqpr}|N\rangle$
(6a)
$\bar{R}$
R
$\underline{R}W$
R
$\underline{R}$
$d^{MN}_{prqq} \leftarrow \langle M|\hat{e}_{prqq}|N\rangle$
(6b)
$\bar{R}$
R
$\underline{R}\bar{R}$
R
$\underline{R}$
$d^{MN}_{pqqr} \leftarrow \frac{1}{2}\langle M|\hat{e}_{pqqr}|N\rangle$
(7)
$\bar{L}$
L
$\bar{R}\underline{L}$
R
$\underline{R}$
$d^{MN}_{pqqr} \leftarrow \frac{1}{2}\langle M|\hat{e}_{pqrq}|N\rangle$
r
q
p
(8a)
W
D
$\bar{R}$
R
$\underline{R}$
$d^{MN}_{pqrr} \leftarrow \langle M|\hat{e}_{pqrr}|N\rangle$
(8b)
$\overline{RL}$
RL
$R\underline{L}$
R
$\underline{R}$
$d^{MN}_{prqr} \leftarrow \frac{1}{2}\langle M|\hat{e}_{prrq}|N\rangle$
(9)
$\overline{RL}$
RL
$\underline{R}L$
L
$\underline{L}$
$d^{MN}_{prqr} \leftarrow \frac{1}{2}\langle M|\hat{e}_{qrrp}|N\rangle$
(10)
$\overline{RR}$
RR
$R\underline{R}$
R
$\underline{R}$
$d^{MN}_{prqr} \leftarrow \frac{1}{2}\langle M|\hat{e}_{prqr}|N\rangle$
(11a)
W
D
W
$d^{MN}_{ppqq} \leftarrow \langle M|\hat{e}_{ppqq}|N\rangle$
(11b)
$\overline{RL}$
RL
$\underline{RL}$
$d^{MN}_{pqpq} \leftarrow \frac{1}{2}\langle M|\hat{e}_{pqqp}|N\rangle$
q
p
(12a)
$\bar{R}$
R
$\underline{R}W$
$d^{MN}_{pppq} \leftarrow \langle M|\hat{e}_{pqpp}|N\rangle$
(12b)
$W\bar{R}$
R
$\underline{R}$
$d^{MN}_{pqqq} \leftarrow \langle M|\hat{e}_{qqpq}|N\rangle$
(12c)
$\bar{R}$
R
$\underline{R}$
$D^{MN}_{pq} \leftarrow \langle M|\hat{E}_{pq}|N\rangle$
(13)
$\overline{RR}$
RR
$\underline{RR}$
$d^{MN}_{pqpq} \leftarrow \frac{1}{2}\langle M|\hat{e}_{pqpq}|N\rangle$
(14a)
WW
$d^{MN}_{pppp} \leftarrow \langle M|\hat{e}_{pppp}|N\rangle$
(14b)
W
$D^{MN}_{pp} \leftarrow \langle M|\hat{E}_{pp}|N\rangle$

Figure 2. The list of unique Shavitt loop shapes used to construct the full set of density matrix elements. The coefficients of each of the Shavitt loop values along with the density matrix indices are given below the loop shape. The segment value types at each level are written to the right of the loop shape diagram. The loop shapes are grouped according to the number of distinct orbital indices.

product satisfies $(\underline{x}_t\bar{x}_h) \leq N_{csf}$. In the construction of a Hamiltonian matrix−vector product, each Shavitt loop therefore is associated with $2(\underline{x}_t\bar{x}_h)$ arithmetic operations, a multiply and an add for each combination of CSF indices m and n.

In a typical implementation of this approach, only Shavitt loops that contribute to the upper triangle of the Hamiltonian matrix in the CSF basis are constructed explicitly, and the bra and ket CSF indices are interchanged in order to generate the contribution to the other half of the (symmetric) Hamiltonian matrix; in this case, each (nondiagonal) Shavitt loop is associated with 4 $(\underline{x}_t\bar{x}_h)$ arithmetic operations in a matrix−vector product operation. It is only these distinct loop shapes that are shown in Figure 2. If a reverse lexical index scheme is used to determine the arc weights, then the above comments still apply, but the roles of the upper and lower walks are reversed and the upper/lower triangle designation of the individual loop shapes

```
$\beta = (\mathscr{W}(T) \cdot h_{pq})$ or $\beta = (\mathscr{W}(T) \cdot g_{pqrs})$
DO upper = 1, $\bar{x}_h$ ! loop over upper walks
    Compute $m_0$ and $n_0$ for this upper walk
    DO lower = 1, $\underline{x}_l$    ! loop over lower walks
        $m = m_0$ + lower
        $n = n_0$ + lower
        $w_m = w_m + \beta \cdot x_n$
    ENDDO
ENDDO
```

Figure 3. Outline of $\mathbf{w} = \hat{\mathbf{H}}\mathbf{x}$ in the CSF basis with the traditional GUGA approach. This is the contribution from a single Shavitt loop value $\mathscr{W}(T)$. The same upper and lower walk structure applies to both one-electron and two-electron Hamiltonian integral contributions.

TABLE 4: Node Pairs[a]

ΔN	$(\Delta a, \Delta b)$
-1	$(0, -1), (-1, +1)$
0	$(0, 0), (-1, +2), (+1, -2)$
$+1$	$(0, +1), (+1, -1)$
$+2$	$(0, +2), (+1, 0), (+2, -2)$

[a] Node pairs that contribute to upper triangle Shavitt loops; $\Delta N = N_{\text{bra}} - N_{\text{ket}}$; $\Delta a = a_{\text{bra}} - a_{\text{ket}}$; $\Delta b = b_{\text{bra}} - b_{\text{ket}}$. Each node of the Shavitt graph may be paired with other nodes at the same level characterized by these $(\Delta a, \Delta b)$ values.

is sometimes switched; the MCSCF DRT and internal part of the CI DRT used in the COLUMBUS program system employ a reverse lexical index scheme.[5,13−16]

At this point, we introduce an auxiliary graphical structure that will be used in the development of the current method. This will be called the *auxiliary pair graph*. Like the corresponding Shavitt graph, it is ordered by orbital level from bottom to top. Each level of the pair graph has a set of nodes which, to avoid confusion with the Shavitt graph, will be called *vertices*. Each vertex of the pair graph corresponds to a pair of nodes (j, j') in the same level of the Shavitt graph. The only node pairs of interest are those that contribute to the Shavitt loops associated with the coupling coefficients in eq 7. The first step in constructing the pair graph is to determine this list of node pairs at each level. The node pairs are characterized first by the quantity ΔN with $\Delta N = N_{\text{bra}} - N_{\text{ket}}$. The node pairs that contribute to the Hamiltonian matrix satisfy $|\Delta N| \leq 2$, and the node pairs that contribute to the upper triangle of the CSF Hamiltonian matrix correspond to the subset $-1 \leq \Delta N \leq 2$. The $\Delta N = -2$ node pairs contribute only to the lower triangle of the Hamiltonian matrix, so they are not referenced in our approach in which only the distinct upper triangle Shavitt loops are generated and the bra−ket interchange is applied in order to generate the other triangle terms. The node pairs are next characterized by $\Delta b = b_{\text{bra}} - b_{\text{ket}}$. The node pairs that contribute to Hamiltonian matrix elements satisfy the spin condition $|\Delta b| \leq 2$. The result from both the occupation and the spin conditions is that an individual node in a Shavitt graph can be paired with, at most, only 10 other nodes (at the same level). These 10 possibilities, each of which corresponds to a unique $(\Delta a, \Delta b)$ combination, are enumerated in Table 4. Thus, the total number of node pairs satisfies $N_{\text{pair}} < 10N_{\text{row}}$, and for large Shavitt graphs, the number of node pairs increases only linearly with the number of nodes (rather than quadratic as the term might imply). Table 2 shows some actual node pair counts for a selection of Shavitt graphs, and it is seen that the total number of node pairs ranges between 2 and 9 times the number of nodes. At the beginning of the procedure, all of the possible node pairs for the given Shavitt graph are determined, and they are grouped in such a way that they can be indexed by orbital level and by ΔN; other than this grouping, we impose in our initial implementation no further ordering of the node pairs. This set of node pairs defines the vertices of the auxiliary pair graph. The next step of the pair graph construction consists of enumerating all possible segments that connect node pairs at one level to the node pairs at the next higher level. These segments are specific instances of the segment shapes enumerated in Table 3. The connection between two vertices of the pair graph will be called an *edge*. Thus, in the present discussion, the Shavitt graph consists of nodes connected by arcs, whereas the pair graph consists of vertices connected by edges.

The auxiliary pair graph does not replace the Shavitt graph in this formulation; in fact, it contains, in principle, no new information. Rather, it simplifies the conceptual treatment of Shavitt loop construction and coupling coefficient evaluation. This is demonstrated in the right-hand side of Figure 1 in which the Shavitt loop is represented as a single directed walk on the auxiliary pair graph. In general, each pair of CSFs that form a nonzero coupling coefficient corresponds to a path on the node-pair graph from its tail to its head. There is a bottom part of this path that corresponds to the lower walk of the Shavitt loop, there is a middle section of the path that corresponds to the Shavitt loop itself, and there is an upper section of the path that corresponds to the upper walk of the Shavitt loop. Pairs of CSFs that do not satisfy the ΔN and Δb conditions at each level (and therefore would result in zero coupling coefficient values) cannot be represented as a path on the pair graph. The segment values of the Shavitt loop on the left of Figure 1 are associated with edges of the pair graph, and the coupling coefficient value, which is computed as the product of segment values of the Shavitt loop in the Shavitt graph, corresponds to a product of edge values in the node-pair graph. A particular edge of the pair graph may be associated with several different Shavitt loops of the same shape and the same segment value and also with Shavitt loops of different loop shapes; each edge may thereby be associated with several different segment values.

When constructing a transition density matrix with the traditional GUGA approach, the following contribution is accumulated for each Shavitt loop T:

$$D_{pq}^{MN} \leftarrow \mathscr{W}(T) \cdot \sum_{\text{upper}(T)} \sum_{\text{lower}(T)} x_m^M x_n^N \tag{11}$$

If only the upper-triangle Shavitt loops are generated, then the summation includes also the bra−ket interchange terms. Thus, there are either 2 ($\underline{x}_l \bar{x}_h$) or 4 ($\underline{x}_l \bar{x}_h$) arithmetic operations, depending on how the CSF index symmetry is treated, associated with each Shavitt loop value in the construction of the density matrix element, the same operation count as for a Hamiltonian matrix−vector product. The density matrix element is the summation of these contributions over all Shavitt loops

$$D_{pq}^{MN} = \sum_T \mathscr{W}(T) \cdot \sum_{\text{upper}(T)} \sum_{\text{lower}(T)} x_m^M x_n^N \tag{12}$$

We now apply the above density matrix construction approach to a pair of product functions. It is convenient to separate the arc factors from eq 3 into the three parts corresponding to the lower walk range, the loop range, and the upper walk range.

$$x_m^M = \left(\prod_u^{(\text{lower})} \alpha_{\mu(u,m)}^M\right) \cdot \left(\prod_u^{(\text{loop})} \alpha_{\mu(u,m)}^M\right) \cdot \left(\prod_u^{(\text{upper})} \alpha_{\mu(u,m)}^M\right) \tag{13}$$

This separation allows the summation of CSF coefficient products to be written as

$$\sum_{\text{upper}}\sum_{\text{lower}} x_m^M x_n^N = \sum_{\text{upper}}\sum_{\text{lower}} [(\prod_u^{(\text{lower})} \alpha^M_{\mu(u,m)}\alpha^N_{\mu(u,n)}) \cdot (\prod_u^{(\text{loop})} \alpha^M_{\mu(u,m)}\ \alpha^N_{\mu(u,n)}) \cdot (\prod_u^{(\text{upper})} \alpha^M_{\mu(u,m)}\ \alpha^N_{\mu(u,n)})]$$

$$= [\sum_{\text{lower}}(\prod_u^{(\text{lower})} \alpha^M_{\mu(u,m)}\ \alpha^N_{\mu(u,n)})] \cdot (\prod_u^{(\text{loop})} \alpha^M_{\mu(u,m)}\ \alpha^N_{\mu(u,n)}) \cdot [\sum_{\text{upper}}(\prod_u^{(\text{upper})} \alpha^M_{\mu(u,m)}\ \alpha^N_{\mu(u,n)})]$$

$$= \gamma_t^{MN} \cdot (\prod_{u=(p-1)}^{(q-1)} \alpha^M_{\mu(u,m)}\ \alpha^N_{\mu(u,n)}) \cdot \bar{\gamma}_h^{MN} \qquad (14)$$

The arrays γ^{MN} and $\bar{\gamma}^{MN}$ have been introduced in ref 1, and they are used in the construction of the overlap matrix $S_{MN} = \langle M|N\rangle$, of the derivatives $\partial S_{MN}/\partial\alpha_\mu$, and in the construction of node and arc densities. These arrays, which do not depend on the individual Shavitt loop, can be computed once, using a recursive algorithm with an effort that scales only as the number of nodes in the graph N_{row}, and reused for all Shavitt loops. The transition density construction for product functions from a single Shavitt loop can then be written

$$D_{pq}^{MN} \leftarrow \gamma_t^{MN} \cdot \mathscr{W}(T) \cdot (\prod_{u=(p-1)}^{(q-1)} \alpha^M_{\mu(u,m)}\ \alpha^N_{\mu(u,n)}) \cdot \bar{\gamma}_h^{MN} \qquad (15)$$

It is clear already that the product function leads to a much simpler procedure than the traditional GUGA CI approach because the 4 $(\underline{x}_t\bar{x}_h)$ arithmetic operations associated with the explicit DO loops over the upper and lower walks are eliminated and replaced instead with simple array lookups.

Consider next the computation of the $\mathscr{W}(T)\cdot(\prod_{u=(p-1)}^{(q-1)}\alpha^M_{\mu(u,m)}\ \alpha^N_{\mu(u,n)})$ product in eq 15 for a single Shavitt loop. In the GUGA approach, the loop value is itself a product of segment values

$$\mathscr{W}(T) = \prod_{u=(p-1)}^{(q-1)} \mathscr{W}(T_u) \qquad (16)$$

$T_u \equiv T_u(d_{\text{bra}}, d_{\text{ket}}, b_{\text{bra}}, b_{\text{ket}}; Q_u)$ is the segment of the Shavitt loop T that connects the bra and ket nodes at level u to the bra and ket nodes at level $(u + 1)$, Q_u is the segment value type at level u, and $\mathscr{W}(T_u)$ is the segment value associated with that segment.[2-8] In the pair graph representation, T_u corresponds to an individual edge connecting two vertices. This segment factorization allows the loop product contribution to be written in the factored form

$$W(T)\prod_{u=(p-1)}^{(q-1)} \alpha^M_{\mu(u,m)}\alpha^N_{\mu(u,n)} = \prod_{u=(p-1)}^{(q-1)} W(T_u)\alpha^M_{\mu(u,m)}\ \alpha^N_{\mu(u,n)}$$

$$= \prod_{u=(p-1)}^{(q-1)} F^{MN}_{\nu(u),\nu'(u+1)}(Q_u) \qquad (17)$$

The *segment factor* $F^{MN}_{\nu(u),\nu'(u+1)}(Q_u) \equiv \mathscr{W}(T_u)\alpha^M_{\mu(u,m)}\ \alpha^N_{\mu(u,n)}$ is the combination of the GUGA segment value and the bra and ket arc factors at level u. $\nu(u)$ and $\nu'(u + 1)$ are the vertices of the pair graph that are connected by the edge that corresponds to that segment. The density contribution from a single Shavitt loop can be written as

$$D_{pq}^{MN} \leftarrow \gamma_t^{MN}[\prod_{u=(p-1)}^{(q-1)} F^{MN}_{\nu(u),\nu'(u+1)}(Q_u)]\bar{\gamma}_h^{MN} \qquad (18)$$

We now examine in detail how the contributions from all Shavitt loops T can be computed efficiently. The total density is given by the summation of contributions over all Shavitt loops T.

$$D_{pq}^{MN} = \sum_T \gamma_{t(T)}^{MN} \cdot [\prod_{u=(p-1)}^{(q-1)} F^{MN}_{\nu(u),\nu'(u+1)}(Q_u)] \cdot \bar{\gamma}_{h(T)}^{MN} \qquad (19)$$

In the auxiliary pair graph, this summation corresponds to the contributions from all paths with nonzero loop values. All Shavitt loops that terminate at the head node h share the common factor $\bar{\gamma}_h^{MN}$. If all of these loop contributions are summed together, their density contributions could be computed with a single multiplication by that common factor $\bar{\gamma}_h^{MN}$. That is, if $V_{\nu'}^q$ with $\nu' = \nu'_{(h,h)}$ is the total contribution from all Shavitt loops with head node h

$$V_{\nu'}^q = \sum_T^{(\text{head}h)} \gamma_{t(T)}^{MN} \cdot [\prod_{u=(p-1)}^{(q-1)} F^{MN}_{\nu(u),\nu'}(Q_u)] \qquad (20)$$

then the density can be computed as

$$D_{pq}^{MN} = \sum_h V^q_{\nu'(h,h)}\ \bar{\gamma}_h^{MN} \qquad (21)$$

where the summation is limited to the $N_{\text{row}}(q)$ nodes h at level q. If there are N_ω total Shavitt loops, this approach would reduce the total number of arithmetic operations by the quantity $2[N_\omega - N_{\text{row}}(q)]$ (i.e., N_ω separate multiply−add operations with the factors $\bar{\gamma}_h^{MN}$ would be replaced by $N_{\text{row}}(q)$ operations).

We next consider how the contributions from all Shavitt loops with the head node h can be computed. As shown schematically in the auxiliary pair graph patch in Figure 4, the Shavitt loops with head node h at level q all pass through vertices at level $(q - 1)$ and terminate at the vertex corresponding to the diagonal node pair $\nu' = \nu'_{(h,h)}$. We assume for the moment that all Shavitt loop contributions from the connecting vertices at level $(q - 1)$ are available. Let these quantities be denoted V_ν^{q-1}. In Figure 4, it is assumed that there are four vertices that are associated with edges that connect to the head node h; the actual number of connecting vertices for the various Shavitt loop segments is limited by the $\Delta\Delta N$ segment classification in Table 3 and also by the available arcs and nodes of the Shavitt graph. By using the (arbitrary) vertex labels in Figure 4, the total Shavitt loop contribution from node h may be written as

$$V_1^q = V_2^{q-1}\ F^{MN}_{2,1}(\bar{R}) + V_3^{q-1}F^{MN}_{3,1}(\bar{R}) + V_5^{q-1}F^{MN}_{5,1}(\bar{R}) + V_7^{q-1}F^{MN}_{7,1}(\bar{R}) \qquad (22)$$

In general, for an arbitrary loop shape, the total contributions to the head node h may be computed as

$$V^q_{\nu'(h,h)} = \sum_\nu V_\nu^{q-1}F^{MN}_{\nu,\nu'(h,h)}(Q_{q-1}) \qquad (23)$$

$\nu'_{(h,h)}$ is the diagonal node pair (h, h) associated with the loop

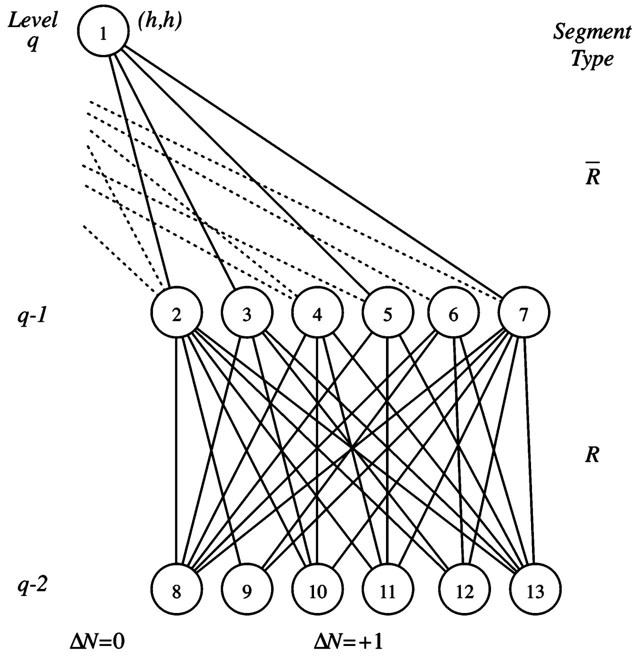

Figure 4. The Shavitt loop contributions represented on the auxiliary pair graph for all contributions to node h at level q. The $\Delta N = 0$ pair graph vertex denoted (h, h) corresponds to the Shavitt graph node h. For this Shavitt loop shape, there are four edges, corresponding to $\Delta\Delta N = -1$ in Table 3, that connect the (h, h) vertex to four $\Delta N = +1$ vertices at level $(q - 1)$. Each vertex at level $(q - 1)$ is connected to up to six vertices at level $(q - 2)$ corresponding to the $\Delta\Delta N = 0$ segments in Table 3. The vertices at level $(q - 1)$ connect to other $\Delta N = 0$ head nodes, which are not shown explicitly. Each vertex ν in the pair graph corresponds to a V_ν value. The V_ν^u values at level u in the pair graph are computed recursively from the V_ν^{u-1} values at level $(u - 1)$ and the $F_{\nu,\nu'}^{MN}(Q_{u-1})$ values associated with the connecting edges.

head h in level q. The summation ranges over all vertices ν at level $(q - 1)$ that connect to $\nu'_{(h,h)}$ with segment type Q_{q-1}.

This same procedure can then be invoked for the segment factors between levels $(q - 2)$ and $(q - 1)$. For example, the contributions to node $\nu = 2$ in Figure 4 may be computed as

$$V_2^{q-1} = \sum_{\nu=8}^{13} V_\nu^{q-2} F_{\nu,2}^{MN}(R) \tag{24}$$

in which node $\nu = 2$ is connected to the six nodes shown at the lower level. In general, the node-pair values $V_{\nu'}^u$ can be computed recursively from the V_ν^{u-1} node-pair values and the $F_{\nu,\nu'}^{MN}(Q_{u-1})$ segment factors that are common to all Shavitt loops that share that segment. At an arbitrary level within the Shavitt loop, we have the general recursion relation

$$V_{\nu'}^u = \sum_\nu V_\nu^{u-1} F_{\nu,\nu'}^{MN}(Q_{u-1}) \tag{25}$$

The summation is limited to only the nodes ν in level $(u - 1)$ that connect to node ν' in level u with segment type Q_{u-1}. As seen in Table 3, the number of terms in this summation ranges between one and six, depending on the $\Delta\Delta N$ segment type and on the available edges in the pair graph. The common factors $F_{\nu,\nu'}^{MN}(Q_{u-1})$ at each level within the loop range can be separated, level by level, until all that remains are the γ_t^{MN} factors for all possible loop tails t at level $(p - 1)$. The D_{pq}^{MN} construction process then consists of an initiation at level $(p - 1)$ over all Shavitt loop tails, a level-by-level propagation of node-pair values V_ν^u up to level $(q - 1)$, and a termination at level q over all loop heads. The overall procedure for product functions consists not of constructing individual Shavitt loops and summing each of those contributions individually, but rather the propagation of all possible Shavitt loop contributions from all node pairs at one level to the node pairs at the next higher level, and finally the combination over all possible loop heads to construct the density matrix element.

The above recursive formulation is equivalent to a sequence of matrix−vector products.

$$\mathbf{V}^u = \mathbf{V}^{u-1}\mathbf{F}^{MN}(Q_{u-1}) \tag{26}$$

A transition density element may be written in symbolic form as

$$D_{pq}^{MN} = \gamma_{(p-1)}^{MN}\mathbf{F}^{MN}(Q_{p-1})\mathbf{F}^{MN}(Q_p)\cdots\mathbf{F}^{MN}(Q_{q-2})\mathbf{F}^{MN}(Q_{q-1})\bar{\gamma}_{(q)}^{MN} \tag{27}$$

The matrices $\mathbf{F}^{MN}$ are rectangular and, depending on the loop shape Q and the complexity of the underlying auxiliary pair graph, relatively sparse. The rows and columns of the $\mathbf{F}^{MN}$ matrices are indexed by the vertices of the auxiliary pair graph at the two adjacent orbital levels. $\bar{\gamma}_{(q)}^{MN}$ is a column vector indexed by diagonal head node pairs at level q, and $\gamma_{(p-1)}^{MN}$ is a row vector indexed by diagonal tail node pairs at level $(p - 1)$. Our recursive procedure outlined above is equivalent to the computation of this product in left-to-right order, and the node pair values $\mathbf{V}^u$ in eq 26 are the intermediate (row) vectors at level u in this series of matrix−vector products.

It has been assumed in the above discussion that the orbital levels p and q are given, and that the contributions from all possible Shavitt loops are included into the D_{pq}^{MN} matrix element for that pair of orbital indices. Given that D_{pq}^{MN} has been constructed using the recursive level-by-level procedure of eq 25, consider the construction of the $D_{p(q+1)}^{MN}$ element. This would, in principle, consist of the same initiation step at level $(p - 1)$ and propagation with factors $F_{\nu,\nu'}^{MN}(R)$ from level p up to level $(q - 1)$, then another propagation with $F_{\nu,\nu'}^{MN}(R)$ from level $(q - 1)$ to level q, and finally the termination with the $F_{\nu,\nu'}^{MN}(\bar{R})$ segment factors from level q to the loop head nodes at level $(q + 1)$. The first steps of this process, initiation at level $(p - 1)$ and propagation up to level $(q - 1)$, are repetitions of the D_{pq}^{MN} construction steps. This suggests a procedure in which these common initiation and propagation steps are performed only once, and that effort is shared for both density elements. This is accomplished by constructing D_{pq}^{MN} first, then discarding the termination information at level q, propagating up a single level from $(q - 1)$ to q, and then terminating at level $(q + 1)$ to form $D_{p(q+1)}^{MN}$. Given the node-pair values $\mathbf{V}^{q-1}$ from the D_{pq}^{MN} construction step, the effort required to construct $D_{p(q+1)}^{MN}$ is only the propagation step to form $\mathbf{V}^q$ and the termination step at level $(q + 1)$. Our algorithm to achieve this reuse of the shared node-pair values for the entire range of orbital indices p and q for the one-particle density is shown in Figure 5.

It should be pointed out that our choice of beginning the propagation procedure at the level $(p - 1)$ and propagating up to level q is arbitrary; it would be just as easy to begin the procedure at level q and propagate down to level $(p - 1)$. That is, eq 27 could just as easily be evaluated in right-to-left order, or some other combination of right-to-left for some levels and left-to-right for other levels. This recursive procedure with reuse of shared node-pair values may be applied to both one-particle Shavitt loops and also to the various two-particle Shavitt loops. As seen in Figure 2, there are only 24 different Shavitt loop

```
DO $p = 1, n$
    Compute $D_{pp}^{MN}$ .
    Initiate $V_{\nu'}^{p} = \sum_{t} \gamma_{t}^{MN} F_{\nu_{(t,t)},\nu'}^{MN}\left(\underline{R}\right)$.
    DO $q = (p+1), n$
        Terminate $V_{\nu'_{(h,h)}}^{q} = \sum_{\nu} V_{\nu}^{q-1} F_{\nu,\nu'_{(h,h)}}^{MN}\left(\bar{R}\right)$.
        Compute $D_{pq}^{MN} = \sum_{h} V_{\nu_{(h,h)}}^{q} \bar{\gamma}_{h}^{MN}$ .
        Propagate $V_{\nu'}^{q} = \sum_{\nu} V_{\nu}^{q-1} F_{\nu,\nu'}^{MN}\left(R\right)$ to prepare for the next $q$.
    ENDDO $q$
ENDDO $p$
```

Figure 5. The recursive procedure to compute the reduced transition density matrix $\mathbf{D}^{MN}$.

```
DO $p = 1, n$
    Compute $d_{pppp}^{MN}$ and $D_{pp}^{MN}$ from loop shapes (14ab).
    Initiate $V_{\nu'}^{p}$ for all distinct bottom loop segments.
    DO $q = (p+1), n$
        Terminate all 2-index loop shapes, (11ab)–(13), and
            compute $d_{ppqq}^{MN}$, $d_{pqpq}^{MN}$, $d_{pppq}^{MN}$, $d_{pqqq}^{MN}$, and $D_{pq}^{MN}$ .
        Propagate $V_{\nu'}^{q} = \sum_{\nu} V_{\nu}^{q-1} F_{\nu,\nu'}^{MN}\left(Q_{q-1}\right)$ to prepare for the 3- and 4-index loops.
        DO $r = (q+1), n$
            Terminate all 3-index loop shapes, (4ab)–(10), and
                compute $d_{ppqr}^{MN}$, $d_{pqpr}^{MN}$, $d_{prqq}^{MN}$, $d_{pqqr}^{MN}$, $d_{pqrr}^{MN}$, and $d_{prqr}^{MN}$ .
            Propagate $V_{\nu'}^{r} = \sum_{\nu} V_{\nu}^{r-1} F_{\nu,\nu'}^{MN}\left(Q_{r-1}\right)$ to prepare for the 4-index loops.
            DO $s = (r+1), n$
                Terminate all 4-index loop shapes, (1ab)–(3ab),
                    and compute $d_{pqrs}^{MN}$, $d_{psqr}^{MN}$, and $d_{prqs}^{MN}$ .
                Propagate $V_{\nu'}^{s} = \sum_{\nu} V_{\nu}^{s-1} F_{\nu,\nu'}^{MN}\left(Q_{s-1}\right)$ to prepare for the next $s$.
            ENDDO $s$
            Propagate $V_{\nu'}^{r} = \sum_{\nu} V_{\nu}^{r-1} F_{\nu,\nu'}^{MN}\left(Q_{r-1}\right)$ to prepare for the next $r$.
        ENDDO $r$
        Propagate $V_{\nu'}^{q} = \sum_{\nu} V_{\nu}^{q-1} F_{\nu,\nu'}^{MN}\left(Q_{q-1}\right)$ to prepare for the next $q$.
    ENDDO $q$
ENDDO $p$
```

Figure 6. Outline of the recursive procedure to compute the full set of reduced transition density matrices $\mathbf{D}^{MN}$ and $\mathbf{d}^{MN}$.

shapes, so the programming effort that is required to implement this approach for all one- and two-particle operators is relatively modest. It is also seen in Figure 2 that a two-particle density matrix element sometimes consists of contributions from two Shavitt loop shapes. Our recursive algorithm for the construction of the full set of both the one- and two-particle transition density matrix elements is outlined in Figure 6. There is no component of effort of this recursive algorithm that scales explicitly as N_{csf}, and there is no storage requirement that scales as N_{csf}.

3. Results and Discussion

In this section, we consider the computational costs that are associated with our new method in terms of storage and floating point arithmetic effort. In contrast to the existing codes that compute Shavitt loops and individual loop values, there is relatively little arithmetic effort that is involved with this method. In the traditional GUGA CI applications, there are 4 $(\underline{x}_t\bar{x}_h)$ arithmetic operations associated with each Shavitt loop value (assuming the bra−ket interchange is invoked on the upper triangular Shavitt loop shapes). In the specific case of traditional MR−CI(SD), only partial Shavitt loops are constructed in the internal orbital space, the Shavitt loop contributions in the larger external orbital space are treated implicitly, and consequently, there is even a larger ratio of arithmetic effort to Shavitt loop construction effort involved. In the product function case, however, these arithmetic operations associated with the upper and lower walks are eliminated entirely and replaced with array lookups. This makes the efficiency of the segment factor propagation critical to the overall performance. To address this critical aspect of the method, we have developed an efficient algorithm to propagate the segment factor contributions from one level to the next. This algorithm is discussed in detail in the Appendix.

TABLE 5: Operation Counts[a]

t	N_t^{IPT}	DO loop	$N_{u,t}$
1	6	p	1
2	24	q	$(u - 1)$
3	27	r	$(u - 1)(u - 2)/2$
4	6	s	$(u - 1)(u - 2)(u - 3)/6$

[a] N_t^{IPT} is the number of initiations, propagations, and terminations that occur for the full set of Shavitt loop shapes. The DO-loop indices correspond to those in Figure 6. $N_{u,t}$ is the number of times each set of propagations occur at orbital level u for the group of t-index Shavitt loop shapes.

Examination of Figure 6 reveals four nested DO loops, suggesting an effort $\Theta(n)$ that scales as a fourth-order polynomial in the molecular orbital basis size n. Table 5 summarizes the computational effort associated with each group of Shavitt loop shapes. The integer t corresponds to the number of distinct orbital indices in the group of Shavitt loop shapes: $t = 1$ corresponds to the one-index loop shapes (14*ab*), $t = 2$ corresponds to the two-index loop shapes (11*ab*)−(13), and so forth. The p, q, r, and s labels correspond to the DO-loop indices in Figure 6. N_t^{IPT} is the total number of initiations, propagations, and terminations (IPT) that occur at each DO-loop level. For example, in the q DO loop, there are 6 Shavitt loop shapes ((11*ab*), (12*abc*), and (13)) that are terminated to construct the d_{ppqq}^{MN}, d_{pqpq}^{MN}, d_{pppq}^{MN}, d_{pqqq}^{MN}, and D_{pq}^{MN} density elements, there are 12 propagations at level q that prepare for the three- and four-index loop shapes, and there are 6 propagations to prepare for the next value of q. These 24 (=6 + 12 + 6) IPT operations are all of the general form of eqs 23 and 25. $N_{u,t}$ is the number of times the propagation and termination steps occur for each orbital level u within the tth nested DO loop. For example, the two-index loop shape contributions in the q DO loop are terminated $(q - 1)$ times at level q in order to generate the $D_{1,q}^{MN}$, $D_{2,q}^{MN}$, ..., $D_{q-1,q}^{MN}$ density elements. Note that there are more terminations at the higher levels than at the lower levels; some consequences of this are discussed below. The effort required for the method is proportional to the sum over orbital levels of the product of the number of IPTs at each orbital level u, the number of times that orbital level occurs in the nested DO loop structure, and β_u the effort required to propagate the node-pair values at that orbital level. A reasonable approximation for this effort is that $\beta_u \propto N_{\mathrm{row}}(u)$ or $\beta_u \propto N_{\mathrm{pair}}(u)$. This total is

$$\Theta(n) = \sum_{t=1}^{4}\sum_{u=1}^{n}\beta_u N_{u,t} N_t^{\mathrm{IPT}} \tag{28}$$

The simplest performance model is to assume that $\beta_u = \beta \propto N_{\text{row}}/n$ is the same for each orbital level. With this simple assumption, the total effort is estimated by

$$\begin{aligned}\Theta(n) &= \beta\sum_{t=1}^{4}\sum_{u=1}^{n} N_{u,t}N_t^{\text{IPT}} \\ &= \beta\sum_{u=1}^{n}[6 + 24(u-1) + 27(u-1)(u-2)/2 + \\ &\qquad (u-1)(u-2)(u-3)] \\ &= \frac{1}{4}\beta(n^4 + 12n^3 + 5n^2 + 6n) \qquad (29)\end{aligned}$$

For large orbital basis sets, the total effort with this assumption is $\mathcal{O}(\beta n^4)$. This is, of course, a very approximate model, but it should be applicable in the general sense if no other information about the Shavitt graph structure is available. In general, each level of the Shavitt graph will have a different number of nodes and node pairs, so the computational effort required for IPT will vary from level to level. In the following, we discuss specific examples of Shavitt graphs and the corresponding β_u factors. There are about $n^2/2$ $\mathbf{D}^{MN}$ elements and $n^4/8$ $\mathbf{d}^{MN}$ elements that must be constructed. Accounting for the orbital index symmetry, the computation of $\text{Tr}(\mathbf{hD}^{MN})$ requires about $n^2/2$ multiply−add operations, and $\text{Tr}(\mathbf{gd}^{MN})$ requires about $n^4/8$ multiply−add operations to form $H_{MN} = \langle M|\hat{H}|N\rangle$. With this parametrization, the effort required to construct $S_{MN} = \langle M|N\rangle$ scales[1] as $\mathcal{O}(\beta n)$. Thus, for large orbital basis sets and Shavitt graphs with reasonably large β_u factors, we expect the overall effort to be dominated by the $\mathbf{d}^{MN}$ construction of the four-index Shavitt loop shapes.

The timings in Table 2 show that our method is very efficient, and it scales very well for increasing wave function expansion length, compared to traditional CI approaches. The smaller calculations in Table 2 were done both with traditional GUGA CI procedures and with our new procedure, and all density elements and energies were verified for correctness. For the larger calculations which cannot be done with the current capabilities of traditional CI methodology, an arbitrary set of nonzero arc factors were chosen in order to generate the necessary statistics and timings. For these full-CI wave functions, the actual number of nodes in level u is given by $N_{\text{row}}(u) = (m+1)(m+2)/2$ for $m = \min(u, n-u)$; that is, the number of nodes per level starts at 1 at level zero, increases to a maximum at level $n/2$, and then decreases back down to 1 at level n. If we assume that $\beta_u \propto N_{\text{row}}(u)$, then the total effort may be modeled as

$$\begin{aligned}\Theta(n) &\propto \sum_{t=1}^{4}\sum_{u=1}^{n} N_{\text{row}}(u)N_{u,t}N_t^{\text{IPT}} \\ &= (13n^6 + 300n^5 + 1855n^4 + 4140n^3 + 3172n^2 + \\ &\qquad 3120n)/1920 \quad (30)\end{aligned}$$

Thus, for these $n = N$ full-CI wave functions, we see that the effort to construct a Hamiltonian matrix element scales formally as $\mathcal{O}(n^6)$ for our procedure. With the appropriate overall scaling, the above model agrees very well with the timings given in Table 2. However, if the timings in Table 2 are fit to a general fourth-order polynomial, the resulting agreement is also very good. This suggests that the wave functions in Table 2 are not large enough to observe the above predicted $\mathcal{O}(n^6)$ asymptotic scaling behavior.

Using traditional CI methodology, the effort to construct H_{MN} for the full-CI expansions in Table 2 scales as $\mathcal{O}(N_{\text{csf}}n^4)$; each row of the Hamiltonian matrix in the CSF basis contains $\sim N^2n^2$ nonzero elements, and $\mathbf{w}^N = \hat{\mathbf{H}}\mathbf{x}^N$ requires a multiply and add operation for each nonzero element. The subsequent scalar product $H_{MN} = \mathbf{x}^M\cdot\mathbf{w}^N$ requires only $\mathcal{O}(N_{\text{csf}})$ additional effort. The details of various efficient full-CI methods are given in refs 19−22. As discussed in more detail in ref 1, $N_{\text{csf}} \approx (8\cdot 4^n)/(\pi n^2)$ for large n for these $n = N$ full-CI singlet wave functions, which results in an overall scaling of about $\mathcal{O}(n^2 4^n)$ to construct H_{MN} using traditional CI methods. Our method has much better scaling properties than the traditional CI approach for the construction of a Hamiltonian matrix element between two product functions. In fact, the results for the largest expansions in the bottom half of Table 2 cannot, practically speaking, be performed using traditional CI methodology using current hardware technology. For example, Table 9 in ref 22 gives timings of 7928 s/iteration for an $n = N = 14$ calculation on the stilbene molecule (on an IBM SP3 with peak CPU performance of 1.5 GFLOP/s). Almost all of this time is the computation of the $\mathbf{w} = \hat{\mathbf{H}}\mathbf{x}$ matrix−vector product. As seen in Table 2, this time can be compared to 0.01 s using our recursive algorithm with product functions (on a PowerMac G5 with peak CPU performance of 10 GFLOP/s). Using the above operation count estimates, and ignoring the practical difficulties of storing vectors of length $N_{\text{csf}} > 10^{20}$, a rough estimate to compute H_{MN} for the $n = 46$ row of Table 2 using traditional CI technology is about $1.85\cdot 10^{24}$ s (over one million times longer than the age of the universe). In contrast, our timing results in Table 2 show that only a modest effort of a few seconds is required by our recursive algorithm for this Shavitt graph, which corresponds to an expansion space of $N_{\text{csf}} \approx 5.5\cdot 10^{24}$, or over 9.2 mol of CSFs. (We note in passing that the CSF count given in Table 9 in ref 22 is incorrect; the correct CSF count in Table 2 was used in the above extrapolation estimate.) This is, of course, somewhat of an apples-to-oranges comparison, because the traditional CI methodology allows for the computation of the matrix−vector product and matrix element H_{MN} using arbitrary CSF vectors $\mathbf{x}^M$ and $\mathbf{x}^N$, whereas our new method only allows for vectors $\mathbf{x}^M$ and $\mathbf{x}^N$ that may be represented in terms of arc factors according to eq 3. Furthermore, these timings are for only a single matrix element H_{MN} (which is analogous to a subspace matrix element in a Davidson iterative procedure), and at this time, we do not know how many of these matrix elements will be required with our new method in order to compute accurate energy values and wave functions. Nonetheless, the dramatic differences in timings, storage, and overall effort suggest the tremendous potential of our new method.

Consider a singlet full-CI wave function expansion in which $n > N$. The Shavitt graph in this case begins at level 0 with a single node and builds up to a maximum number of nodes at level $N/2$. The number of nodes at this level is given by $(N+2)(N+4)/8$. Between levels $N/2$ and $(n - N/2)$, every level in the Shavitt graph is repeated with the same (a, b) nodes, the same node pairs, the same set of connecting segments in the pair graph, and the same segment values $\mathcal{W}(T_u)$. (The segment factors $F^{MN}_{\nu,\nu'}(Q_u)$ would be different at each level because of the arc factors.) Above level $(n - N/2)$, the number of nodes per level begins to decrease, down to one node at level n. This situation involving the repeated levels can be exploited by using pointers at each level into a common set of segment values. In this special case, the storage requirements for the segment values would not increase for increasing n (with fixed N). Furthermore, in this situation, the β_u factor to propagate from level to level does not change in the middle section of the Shavitt graph ($\beta_u \propto N^2/8$), so the overall effort to compute $\mathbf{d}^{MN}$, and also H_{MN}, would scale as $\mathcal{O}(N^2n^4)$. For fixed N, the effort would increase only as $\mathcal{O}(n^4)$ as orbitals are added. This scaling would apply, for example, to the common situation in which a sequence of increasingly accurate orbital basis sets is used for a given molecular system in order to extrapolate to an infinite basis set limit.

Consider the application of the product function approach to a MR−CI Shavitt graph. In this situation, the number of electrons that occupy the external orbitals is constrained to some maximum value. This constraint is achieved by deleting the majority of nodes in the external orbital levels in order to limit the cumulative occupancy of these orbitals. The external orbitals are placed typically at the bottom of the Shavitt graph.[5,6,16] In the MR−CI(SD) case, for example, the external orbital levels have up to 2 electrons, resulting in only the 4 nodes corresponding to (a, b) pairs (0, 0), (0, 1), (0, 2), and (1, 0) (above level 1 in the external space). According to Table 4, these 4 nodes result in 14 node pairs per level. The effort required to propagate the segment factors within the external orbital space is therefore constant ($\beta_u \propto 14$) and relatively small regardless of the total number of electrons in the system. Increasing the orbital basis size results in a procedure that scales only as $\mathcal{O}(14n^4)$. If the complexity of the external orbital space were increased, for example, to allow up to 3 electrons in the external orbital space for a MR−CI(SDT) expansion, then in addition to the above 4 nodes there would also be nodes with (a, b) pairs (0, 3) and (1, 1) at each level (above level 2), and these 6 nodes would result in 26 node pairs. The computational effort would scale as $\mathcal{O}(26n^4)$, and for a given n, the effort would be about $(^{26}/_{14})$ larger than for the MR−CI(SD) calculation. That is, the MR−CI(SDT) H_{MN} calculation would be expected to cost about twice as much as the MR−CI(SD) calculation. Allowing 4 electrons in the external space for a MR−CI(SDTQ) expansion would require adding the nodes (0, 4), (1, 2), and (2, 0) to each of the external orbital levels (above level 3), and these 9 nodes result in 46 node pairs and an effort that would scale as $\mathcal{O}(46n^4)$; for a given n, the total effort would be about $(^{46}/_{26})$ larger than for the MR−CI(SDT) calculation, and it would be about $(^{46}/_{14})$ larger than the MR−CI(SD) calculation. As shown in Table 5, propagations occur at the top of the Shavitt graph with this approach more often than the propagations near the bottom of the Shavitt graph. Consequently, for MR−CI Shavitt graphs, it would appear to be more efficient to place the internal orbitals (which have more node pairs per level) at the bottom of the Shavitt graph where propagations occur fewer times and to place the external orbitals (which have fewer node pairs per level) at the top where propagations occur more frequently. Alternatively, the traditional orbital placement could be chosen, but the recursive propagation procedure could be performed from the top of the Shavitt graph down to the bottom. These orbital ordering considerations apply also to other Shavitt graphs in which different orbital occupation restrictions are imposed on different orbital subsets. The impact of these kinds of optimizations will be examined in the future.

All of the above timings and discussions of scaling with respect to orbital basis and molecule size apply only to the particular situation of the construction of a single Hamiltonian matrix element with two product basis functions, $H_{MN} = \langle M|\hat{H}|N\rangle$. At this point, we do not know how difficult the optimization will be to minimize the energy with respect to the arc factors. This is a very nonlinear optimization problem, and this could result in a difficult numerical procedure. This optimization will be the focus of future effort, and it is expected to share the efficient recursive technology that has been developed in this work. We also have at this time only limited experience[1] with determining the number of product basis functions N_α that will be required to achieve chemical accuracy in global PES computations. The size of N_α will affect the total number of H_{MN} matrix elements that must be computed. However, with the somewhat remarkable timings we see for the computation of a single H_{MN} with this method, we feel that the new method shows great promise even if N_α must be as large as several dozen or several hundred.

As discussed in ref 1, we exploit point group symmetry in this approach by introducing symmetry-dependent chaining indices[23] in the DRT representation of the Shavitt graph. This is appropriate for point groups with one-dimensional irreducible representations (irreps), specifically to the D_{2h} point group and its subgroups. This allows the upper and lower walks at each node to be grouped by the irrep index Γ. For the H_{MN} construction step, this requires propagation of separate $\mathbf{V}$ products for each Γ. If point group symmetry were ignored, then only one set of products would be propagated. Therefore, the use of point group symmetry increases the cost of the calculation by approximately a factor of h, the number of irreps in the point group. In contrast, the use of point group symmetry in traditional CI approaches results in significant reductions in the effort (various parts of the effort and storage requirements are reduced by factors of h to h^3) to compute a matrix−vector product. Even with this cost increase, there are still two major advantages to using point group symmetry with our new method. The first is that it eliminates any unwanted symmetry contamination from the computed wave functions and the energy and property values. This simplifies several aspects of the calculation of potential energy surfaces and molecular properties. The second is that the H_{MN} elements for several Γ may be computed with relatively little additional effort beyond that required for a single irrep. In traditional CI approaches, the calculation of each irrep involves basically an entirely new calculation, with relatively little shared effort between the separate calculations. With our approach, a single set of propagations results in all h H^{Γ}_{MN} elements. This simplifies several types of calculations, including the simultaneous computation of several ground and excited states of several Γ and the computation of state-averaged energies and properties in which the computed energy or property is a weighted average, with arbitrary user-specified weights, over several individual states. Furthermore, it is straightforward with this approach to compute transition density matrices $\mathbf{D}^{MN}$ and $\mathbf{d}^{MN}$ for arbitrary Γ_M and Γ_N. This allows the computation of transition properties, such as transition dipole moments which may be used for the computation of spectral amplitudes.

4. Conclusions

An efficient procedure has been presented to compute Hamiltonian matrix elements and reduced one- and two-particle density matrices for electronic wave functions using a new graphical-based nonlinear expansion form. The wave functions computed with this method are spin eigenfunctions and may be chosen to belong to pure irreducible representations (for the D_{2h} point group and its subgroups). Wave functions in this approach are expanded in a product function basis, and each product basis function depends, in a nonlinear way, on a relatively small number of variational parameters. In general, the effort required to construct an individual Hamiltonian matrix element between two product basis functions $H_{MN} = \langle M|\hat{H}|N\rangle$ scales as $\mathcal{O}(\beta n^4)$ for a wave function expanded in n molecular orbitals. The prefactor β itself scales between N^0 and N^2, for N electrons depending on the complexity of the underlying Shavitt graph. In previous work, the computation of the corresponding overlap matrix element $S_{MN} = \langle M|N\rangle$ was shown to require effort proportional to $\mathcal{O}(\beta\text{n})$. There is no component of the effort or of the storage requirements for this matrix element construction procedure that is proportional to N_{csf}. These matrix elements allow for the computations of ground and excited electronic states. Timings with our initial implementation of this method are very promising. Wave function expansions that are orders of magnitude larger than can be treated with traditional CI methods require only modest effort with our new method.

Acknowledgment. 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, Geosciences and Biosciences, under contract W-31-109-ENG-38.

Appendix

In this appendix, we discuss the computation of the reduced transition density matrices $\mathbf{D}^{MN}$ and $\mathbf{d}^{MN}$ which are used in the construction of the Hamiltonian matrix element $H_{MN} = \langle M|\hat{H}|N\rangle$. In section 2, the recursive nature of this construction process for product basis functions $|M\rangle$ and $|N\rangle$ was exposed. In this appendix, we discuss further details of our initial implementation, directed primarily toward readers interested in implementing this method or in modifying an existing implementation.

We examine first the underlying data structures we use to carry out the algorithm outlined in Figure 6. This is based on the auxiliary pair graph introduced in section 2. The first step is the determination of the list of node pairs according to Table 4 and the determination of the edges (connecting segments) according to Table 3. Given this representation of the pair graph, we next consider two different choices to compute the segment factors $F^{MN}_{\nu,\nu'}(Q_u)$. The first consists of looping over the vertices of the pair graph at level u, indexing the edge links from each vertex that correspond to segment T_u, and performing a table lookup based on d_{bra}, d_{ket}, b_{bra}, and b_{ket} to access the appropriate segment value $\mathscr{W}(T_u)$. This segment value is then combined with the arc factors to form $F^{MN}_{\nu,\nu'}(Q_u)$, and the contribution is accumulated into the appropriate V^{u+1}_{ν} node-pair value according to eq 25. The second choice[24] consists of precomputing all possible segment values for each segment T_u at all levels u. These segment values are grouped by level so that they can be accessed simply, with minimal overhead, during the **V** propagation. The propagation step then consists of the segment value lookup, multiplication by the arc factors to form the segment factor $F^{MN}_{\nu,\nu'}(Q_u)$, and accumulation into V^{u+1}_{ν}. The first algorithm has the advantage of smaller lookup tables and smaller memory requirements, whereas the second algorithm has the advantage of a simpler innermost DO-loop structure. Also, the second algorithm has the advantage that only nonzero segment values need to be saved, whereas the first algorithm includes lookups of a small fraction of zero segment values (which are tested and ignored). There are 28 segment value types in our implementation; these are grouped according to the ΔN at the top of the segment and enumerated in Table 3. Table 2 shows the total number of segment values for a selection of Shavitt graphs. It is clear from this table that the total number of segment values is modest even for the largest wave function expansions, so this is the algorithm that has been used for the timings in Table 2. Future work will focus on other efficient methods to propagate the **V** array values. For example, if the node-pair vertices are ordered appropriately, the sparse matrix−vector product in our current formulation of eq 27 might be replaced with dense-subblocked or dense-banded matrix−vector products, allowing for higher computational efficiency for the same, or almost the same, total arithmetic operation count.

After the segment value tables are computed, the transition density matrix elements may be computed using the procedure outlined in Figure 6. It might appear that we would need more than 24 node-pair value arrays **V**, one or two for each Shavitt loop shape in Figure 2. In our implementation, only six node-pair value arrays are required. This is due to four different reasons. First, most of the propagation steps are shared by several different Shavitt loop shapes. For example, the lower-level initiation and propagation of the ($\underline{RR}$...) segment factor products is shared by the loop shapes (1*ab*), (3*ab*), (6*ab*), (7), (8*ab*), (10), and (12*bc*). Instead of propagating 12 equivalent sets of values (or 17 sets, if the $x = 0$, 1 sets are counted separately), only one set is propagated over the range of common orbital levels.

Second, of the distinct shapes that are propagated, many of them correspond to node pairs with different ΔN values, and they therefore occupy different elements if stored in the same **V** array. At the lowest orbital level p, only six sets of unique loop values are initiated. These correspond to segment shapes $\underline{L}$ (stored in $\Delta N = -1$ node-pair entries), W and $\underline{RL}^1$ (in $\Delta N = 0$ node pairs), $\underline{R}$ and $\underline{R}W$ (in $\Delta N = +1$ node pairs), and $\underline{RR}^0$ (in $\Delta N = +2$ node pairs). These values require only two sets of node-pair arrays **V** because the ΔN labels are distinct except for the two conflicting cases. As the values are propagated from one level to the next, they are forked and merged as appropriate, and the maximum number of ΔN conflicts at any level in our procedure is six, which results in the requirement for that number of node-pair arrays **V**.

The third reason that relatively few node-pair value arrays are needed is that some of the individual Shavitt loop propagations can be combined before the loop head is reached. This occurs for (1*a*)−(2*a*), (1*b*)−(3*b*), and (2*b*)−(3*a*) pairs of loop shapes. For example, loop shapes (1*a*) and (2*a*) correspond to the operators $\hat{e}_{pqrs}$ and $\hat{e}_{qprs}$, respectively. Both of these loop shapes contribute to the same density matrix element d^{MN}_{pqrs}. In the procedure outlined above, these two contributions could be propagated individually, and the results combined when the loop shapes are terminated at level s. However, between levels r and s, both of the propagations would invoke exactly the same $F^{MN}_{\nu,\nu'}(R)$ and $F^{MN}_{\nu,\nu'}(\bar{R})\bar{\gamma}^{MN}_{h}$ factors and exactly the same node-pair indices. Therefore, it is possible to combine the node-pair values at level r and then to propagate only a single set of node-pair values for the combined (1*a*) + (2*a*) contribution. The same early combination can be applied to the loop shapes (1*b*) + (3*b*) and to loop shapes (2*b*) + (3*a*). Consequently, in the innermost DO loops of the procedure, only three sets of node-pair values need to be propagated rather than six sets, reducing the corresponding computational effort and storage requirements by a factor of one-half.

The fourth reason that only relatively few node value arrays are required is due to the treatment of the $x = 0$ and $x = 1$ spin-recoupled components[6,8,11] of the two-particle operators. In general, we have $e_{pqrs} = e^{0}_{pqrs} + e^{1}_{pqrs}$, and we use the identities

$$\left.\begin{aligned} e^{0}_{psqr} &= e^{0}_{prqs} \\ e^{1}_{psqr} &= -e^{1}_{prqs} \\ e^{0}_{psrq} &= -\frac{1}{2}\, e^{0}_{pqrs} = -\frac{1}{2}\, e_{pqrs} \end{aligned}\right\} \quad \text{for } p, q < r, s \qquad (31)$$

to reduce the number of values that must be propagated. Because these are general operator identities, they hold also for $\langle \tilde{m}|e^{x}_{pqrs}|\tilde{n}\rangle$ matrix elements over primitive CSFs and for the $\langle M|e^{x}_{pqrs}|N\rangle$ linear combinations. For example, we propagate only the $x = 0$ components of loop shape (3*a*), only the $x = 1$ component of loop shape (3*b*), and the sum and difference of these two values are the loop values of interest that form the density matrix elements. This means that only two sets of values need to be propagated for these two loop shapes rather than four sets, halving both the computational effort and the storage requirements for these loop shapes. These operator identities are used to reduce the computational effort and storage requirements for loop shapes (1*ab*), (2*ab*), (3*ab*), (4*ab*), (8*ab*), and (11*ab*). The final result is that only one contribution is required from each of the loop shapes in Figure 2.

Figure 2 is based on Figure 8 in ref 6, but there are a few minor differences. Instead of associating a Hamiltonian integral with each loop shape as in ref 6, Figure 2 shows the density matrix element to which it contributes along with the coefficient that results from the symmetrization in eq 9. In our current procedure, loop shape (9) is not computed at all, since it is equivalent to the bra−ket interchange of loop shape (8*b*) if all possible upper-loop segments are allowed (i.e., without upper-triangle restrictions imposed). Loop shape (7) in Figure 2 is the bra−ket interchanged form of the corresponding loop in ref 6; this interchange allows the lower range propagation to be shared with loop shapes (1*ab*) (and (3*ab*), (6*ab*), (8*ab*), etc.). We compute the loop shape (14*ab*) contributions, d^{MN}_{pppp} and D^{MN}_{pp}, using the procedure described in ref 1. This is because these single-level loop shapes require no propagation, and the fact that the initiation and termination occur logically at the same level makes these density contributions somewhat of a special case. To be consistent with the procedure outlined in Figure 6, all three-index loop shapes are labeled *p*, *q*, and *r*, and all two-index shapes are labeled *p* and *q*. As seen in Figure 6, all four-index loop shapes, which are grouped together in Figure 2, are terminated at the same point in the procedure, all three-index shapes are terminated at the same point, and all two-index shapes are terminated at the same point. Except for these minor changes in notation and indexing conventions, the loop shapes in Figure 2 correspond directly to those in ref 6.

For the various Shavitt loop contributions, there are several ways to compute the final density contributions. For example, instead of loop shapes (2*ab*) as shown in Figure 2, which correspond to an upper-triangle loop contribution, it is possible to compute the bra−ket interchanged version, which corresponds to the equivalent lower-triangle loop contribution. The segment types for the $(2a)^T$ loop, for example, would be, from bottom to top, ($\underline{R}$, R, $\bar{R}$, D, L, $\underline{L}$, $\bar{L}$), and the $(2b)^T$ segment types would be ($\underline{R}$, $\bar{R}$, R, $R\underline{L}$, RL, $\bar{R}L$, L, $\bar{L}$). Computing the contributions from these loops in this way eliminates entirely the need to propagate the two sets of node-pair values corresponding to ($\underline{L}$, L, ...) because these segment values are not shared by any other loop shape. Computing $(2ab)^T$ reduces the propagation effort within the *p*, *q*, and *r* DO loops of the procedure, and it reduces the number of necessary value arrays **V** from six to five, but it appears to eliminate the possibility of the early combination at level *r* of these node-pair values with the loop shape (1*a*) and (3*a*) terms, because these latter terms require (... R, $\bar{R}$) propagation up to level *s* whereas the transposed $(2ab)^T$ terms require (... L, $\bar{L}$) propagation. This increases the effort within the innermost *s* loop because five sets of values must be propagated rather than three. We have computed the density elements both ways, and we find that for the calculations in Table 2, the $(2ab)^T$ loop propagation results in slightly less total effort for $n \leq 12$, whereas the (2*ab*) loop propagation, which allows for the early combination, results in less effort for $n >$ 12. However, another possibility exists which combines the advantages of both approaches. This consists of propagating $(2ab)^T$ up to level *r* and then interchanging the bra−ket values with the ket−bra values. These interchanged values are identical to those that would have been computed if (2*ab*) loop propagations had occurred. This is a consequence of the symmetric segment value convention[6,11] for which $\mathscr{W}(T^\dagger{}_u) = \mathscr{W}(T_u)$ with $T^\dagger{}_u(d_{\mathrm{bra}}, d_{\mathrm{ket}}, b_{\mathrm{bra}}, b_{\mathrm{ket}}; Q_u) = T_u(d_{\mathrm{ket}}, d_{\mathrm{bra}}, b_{\mathrm{ket}}, b_{\mathrm{bra}}; Q^\dagger{}_u)$. These interchanged values can be combined early at level *r* with the (1*a*) and (3*a*) terms, and the remaining propagations require only three **V** arrays within the innermost *s* loop. These and other similar optimization possibilities will be explored in the future.

References and Notes

(1) Shepard, R. *J. Phys. Chem. A* **2005**, *109*, 11629.

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

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

(4) 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 Laboratories: Columbus, OH, June 1979.

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

(6) 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 (Lecture Notes in Chemistry 22)*; Hinze, J., Ed.; Springer-Verlag: Berlin, 1981.

(7) 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, 1983; pp 279−293.

(8) 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.

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

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

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

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

(13) Shepard, R. The Multiconfiguration Self-Consistent Field Method. In *Ab Initio Methods in Quantum Chemistry II*; Advances in Chemical Physics 69; Lawley, K. P., Ed.; Wiley: New York, 1987; pp 63−200.

(14) 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.

(15) 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.

(16) 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.

(17) Shepard, R.; Shavitt, I.; Lischka, H. *J. Comput. Chem.* **2002**, *23*, 1121.

(18) 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.

(19) Gan, Z.; Harrision, R. J. *J. Comput. Chem.* Submitted for publication.

(20) Ansaloni, R.; Bendazzoli, G. L.; Evangelisti, S.; Rossi, E. *Comput. Phys. Commun*. **2000**, *128*, 496.

(21) Duch, W. *Int. J. Quantum Chem.* **1990**, *S24*, 683.

(22) Klene, M.; Robb, M. A.; Frisch, M. J.; Celani, P. *J. Chem. Phys.* **2000**, *113*, 5653.

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

(24) This algorithm was suggested by I. Shavitt, private communication, September, 2005.