

8 Quantum Chemistry DMRG in a Local Basis

E. Miles Stoudenmire

Center for Computational Quantum Physics

Flatiron Institute, New York, NY 10010 USA

Contents

1	Introduction	2
2	DMRG and matrix product states	2
2.1	Matrix product state form of the wavefunction	3
2.2	Overview of the DMRG algorithm	5
2.3	MPO forms of Hamiltonians	8
3	Quantum chemistry: brief overview and discretization methods	9
3.1	Basis set discretization	10
3.2	Grid discretization	13
3.3	Comparison of basis set and grid discretization	14
4	Local bases for quantum chemistry DMRG	14
4.1	Approach 1: hybrid grid and basis set, or “sliced basis” approach	15
4.2	Approach 2: multi-sliced gausslet basis	19
4.3	DMRG with sliced-basis or multi-sliced discretization	27
5	Conclusions and future directions	29

1 Introduction

The electronic structure problem, and the quantum many-body problem in general, is an exponentially hard problem. A full accounting of the quantum mechanical behavior of N electrons would take truly astronomical resources once N is greater than about 30 electrons. This is far fewer than in most chemical or solid state systems! Electronic structure is also a continuum problem, so to apply most computational methods one must discretize the Hamiltonian. The approximations needed to handle the exponentially large many-body wavefunction and the multi-scale details of continuum discretization involve subtle tradeoffs.

By now there are many computational approaches for simulating systems of interacting electrons, but each has major limitations. In quantum chemistry, many computational approaches start from a non-interacting picture, where the electronic wavefunction is a Slater determinant—the fermionic analogue of a product state. Interactions are treated by summing multiple Slater determinants. Such approaches break down when the system becomes *strongly correlated*, meaning that exponentially many Slater determinants are necessary to express the wavefunction accurately. Other approaches like quantum Monte Carlo are limited by the complicated sign structure of the wavefunction. Thus a demand remains for complementary approaches to electronic structure, so that a computational technique exists to treat every type of system.

The computational method which is the focus of this chapter is the *density matrix renormalization group* (DMRG) algorithm. DMRG is interesting for a number of reasons. It can handle strong correlation very naturally, often working better when the system is strongly correlated versus weakly correlated. DMRG is a controlled method with an arbitrary accuracy level determined by the user, though it may come at a high cost depending on the specific system. Finally, DMRG is just one of a number of methods for optimizing *tensor network* wavefunctions, in particular *matrix product states* (MPS), which we define and discuss in more detail below. The tensor network / MPS perspective has become instrumental in adding new capabilities to DMRG and pushing its range of applications, and has opened up an entire field of study into tensor network methods. We will see later how a tensor network representation of the electronic structure Hamiltonian enables very favorable scaling of DMRG calculations for chemistry.

In what follows, we first introduce and review the basics of DMRG, what it accomplishes and how it works. Then we review the essentials of quantum chemistry, defining the electronic structure problem and discussing various ways of discretizing it for computational approaches. We finally turn to recently developed techniques for transforming the electronic structure problem into a form especially suitable for DMRG, built around spatially local choices of basis.

2 DMRG and matrix product states

The *density matrix renormalization group* (DMRG) algorithm is a method for optimizing a particular class of wavefunctions, primarily with the goal of finding ground states of many-body quantum systems [1–4]. The class of wavefunctions DMRG optimizes are known as *matrix product states* (MPS) [5, 4]; these form a very powerful class that can represent wavefunctions

of widely different systems and notably do not suffer from issues related to strong correlation. Though DMRG works best for studying one-dimensional lattice model systems, it can be successfully applied to narrow two-dimensional lattice models [6] and to ab initio Hamiltonians such as in quantum chemistry [7, 8]. DMRG also has extensions which can treat excited states, time evolution, finite-temperature systems, and open systems. Much can be said about technical aspects of DMRG and properties of MPS, such as the number of variational parameters needed for an MPS to represent ground states of various systems accurately. The extensive review Ref. [4] provides a detailed review of MPS techniques in the DMRG context, and Ref. [9] discusses mathematical aspects of the class of MPS wavefunctions and how their complexity scales with properties such as the entanglement entropy of the wavefunction they represent.

2.1 Matrix product state form of the wavefunction

Before introducing the DMRG algorithm itself, let us first define and discuss matrix product states (MPS). Because the focus of this chapter is quantum chemistry, let us focus the discussion to wavefunctions of electrons, which are just fermions that have a spin. Thus the many-body Hilbert space will be a product of single-site (or single-orbital) spaces which are four dimensional, corresponding to the states $\{|0\rangle, |\uparrow\rangle, |\downarrow\rangle, |\uparrow\downarrow\rangle\}$. Indices $s_j = 1, 2, 3, 4$ will refer to these four states on site j of a discrete lattice system (which could be a discretized form of a continuum quantum chemistry system as we will see later in Section 3).

The most general form of a many-body wavefunction on an N site system is

$$|\Psi\rangle = \sum_{\{s\}} \Psi^{s_1 s_2 s_3 \dots s_N} |s_1 s_2 s_3 \dots s_N\rangle. \quad (1)$$

All of the parameters of this wavefunction are stored in the amplitudes $\Psi^{s_1 s_2 s_3 \dots s_N}$ which have the form of an N -index tensor. The fact that this tensor has 4^N distinct components is one manifestation of the exponential many-body problem. For readers not used to the second quantization formalism for describing fermions, note that any choice of amplitudes in Eq. (1) yields a properly antisymmetrized fermionic wavefunction, even if the amplitude tensor has no particular symmetry properties itself. This is because all operators acting on this wavefunction and the basis states $|s_1 s_2 s_3 \dots s_N\rangle$ are defined in terms of fundamental raising and lowering operators $\hat{c}_{j\sigma}^\dagger$ and $\hat{c}_{j\sigma}$ (where $\sigma = \uparrow, \downarrow$) which anti-commute with each other.

2.1.1 Matrix product states

The challenge in dealing with the wavefunction Eq. (1) is finding a manageable representation of the amplitude tensor $\Psi^{s_1 s_2 s_3 \dots s_N}$. Fortunately, for any N -index tensor there is a powerful factorization known as the matrix product state, having the following form:

$$\Psi^{s_1 s_2 s_3 s_4 s_5 s_6} = \sum_{\{\alpha\}} A_{\alpha_1}^{s_1} A_{\alpha_1 \alpha_2}^{s_2} A_{\alpha_2 \alpha_3}^{s_3} A_{\alpha_3 \alpha_4}^{s_4} A_{\alpha_4 \alpha_5}^{s_5} A_{\alpha_5}^{s_6} \quad (2)$$

where the above equation shows the example of an MPS for a $N = 6$ index tensor. Note how each of the factor tensors $A_{\alpha_{j-1}\alpha_j}^{s_j}$ carries exactly one of the original *physical* indices s_j . Note

also how each factor tensor carries two *bond* indices α_i , except for the first and last tensor which carry only one bond index.¹

The range, m , of the bond indices $\alpha_i = 1, 2, \dots, m$ is called the *bond dimension* of the MPS. Although this dimension can vary for each bond index, for simplicity we will sometimes refer to a single overall bond dimension by which we mean the maximum or typical one. The bond dimension of an MPS controls both the expressivity the MPS, meaning its ability to accurately represent complicated wavefunctions, and the cost of computations with the MPS, such as steps of the DMRG algorithm which scale as m^3 . For a large enough bond dimension, an MPS can represent any wavefunction, but in the worst case this requires the bond dimension to be exponentially large. What makes MPS so useful is that in practice they can accurately capture ground states and low-lying excited states of one-dimensional (1D) or quasi-one-dimensional systems for modest bond dimensions, usually in just the many hundreds or few thousands. And in many interesting cases, such as 1D Hamiltonians with finite-range interactions, the accuracy corresponding to a given bond dimension is essentially independent of system size.²

The name *matrix product state* comes from the fact that any single amplitude of the quantum state the MPS represents can be computed as a product of matrices. Say we want to know the amplitude for the state $|0\ 0\ \uparrow\ 0\ \downarrow\ 0\rangle$ in other words the tensor component $\Psi^{0\ 0\ \uparrow\ 0\ \downarrow\ 0}$. We can obtain this by fixing the physical indices in Eq. (2):

$$\Psi^{0\ 0\ \uparrow\ 0\ \downarrow\ 0} = \sum_{\{\alpha\}} A_{\alpha_1}^0 A_{\alpha_1\alpha_2}^0 A_{\alpha_2\alpha_3}^\uparrow A_{\alpha_3\alpha_4}^0 A_{\alpha_4\alpha_5}^\downarrow A_{\alpha_5}^0. \quad (3)$$

Note that after the physical indices are set to fixed values, the factor tensors have at most two free indices. So the above expression can be computed by just treating $A_{\alpha_1}^0$ as a vector, multiplying it with matrix $A_{\alpha_1\alpha_2}^0$, and so on until all of the bond indices are contracted, resulting in the amplitude as a scalar with a computational effort scaling as m^2 .

If a certain wavefunction, such as a ground state, can be successfully approximated by an MPS with a modest bond dimension independent of system size, then one has obtained a hugely compressed representation. Observe that an MPS of typical bond dimension m has a number of parameters which scales as $4Nm^2$ versus 4^N for an arbitrary uncompressed wavefunction. So if m does not depend on N , or depends only very weakly on N , then the number of parameters grows only linearly with system size which is very manageable. In many interesting cases the number of parameters and thus the cost of calculation can be reduced even more using symmetries such as spin symmetry or particle number conservation.

2.1.2 Tensor diagrams

Expressions in traditional tensor notation such as Eq. (2) are tedious to write and difficult to read for computations involving MPS. Fortunately there is a fully rigorous graphical notation for expressing tensor computations known as *diagram notation* [10]. In diagram notation, tensors

¹Another more common name for an MPS or tensor network bond index is *virtual index*.

²There is a logarithmic dependence of bond dimension with system size for critical 1D systems, for which MPS still work very well in practice.

are represented as shapes and indices as lines. Connecting two lines implies those indices are summed over or *contracted*.

As an example, consider two tensors A_{ij} and B_{jkl} . The tensor C_{ilk} resulting from contracting A and B over the index j can be notated diagrammatically as

$$\begin{array}{c} l \\ | \\ i - \text{orange circle} - k \\ | \\ C \end{array} = \begin{array}{c} l \\ | \\ i - \text{blue circle} - \text{green circle} - k \\ | \quad | \\ A \quad B \\ j \end{array} = \sum_j A_{ij} B_{jkl}$$

Common matrix operations can be expressed as tensor diagrams as:

$$\begin{array}{c} \text{green circle} - \text{purple circle} \\ | \quad | \\ i \quad j \end{array} = \sum_j M_{ij} v_j$$

$$\begin{array}{c} \text{green circle} - \text{orange circle} \\ | \quad | \\ i \quad j \end{array} = A_{ij} B_{jk} = AB$$

$$\begin{array}{c} \text{red circle} - \text{blue circle} \\ | \quad | \\ i \quad j \end{array} = A_{ij} B_{ji} = \text{Tr}[AB]$$

Tensor diagram notation has many advantages, such as removing the need to name every index in tensor expressions. Diagrams also make it easy to see how many indices the result of a computation will have. For example, if all of the index lines in a complicated diagram are contracted with another line, then the result must be a scalar, as in the $\text{Tr}[AB]$ example above. The advantage of tensor diagrams becomes most apparent for complicated networks of contracted tensors. For example, the diagram for the MPS of Eq. (2) is just

$$\Psi^{s_1 s_2 s_3 s_4 s_5 s_6} = \begin{array}{c} s_1 \quad s_2 \quad s_3 \quad s_4 \quad s_5 \quad s_6 \\ | \quad | \quad | \quad | \quad | \quad | \\ \text{blue circle} - \text{blue circle} \end{array} \quad (4)$$

which is much simpler than the expression in Eq. (2). Note that not only the α_i bond indices but even the physical s_j indices can be suppressed when using diagrams, but we have shown the physical indices above to make comparison to Eq. (2) easier.

2.2 Overview of the DMRG algorithm

The goal of the DMRG algorithm is to find the ground state of a given Hamiltonian in MPS form. For DMRG to be efficient, not only must the wavefunction be represented efficiently as

an MPS, but the Hamiltonian has to be represented efficiently too. As we will discuss later, finding a compact representation of the Hamiltonian is one of the central challenges in applying DMRG to problems in quantum chemistry. But for the purpose of this section, let us just assume the Hamiltonian is a sum of strictly local terms, such as the Hubbard model in one dimension. Local Hamiltonians have the property that they can be represented as a tensor network known as a *matrix product operator* (MPO).³ This means the Hamiltonian H , viewed as a tensor with $2N$ indices, can be written as a contracted product of factor tensors as follows

$$H_{s_1 s_2 s_3 s_4 s_5 s_6}^{s'_1 s'_2 s'_3 s'_4 s'_5 s'_6} = \begin{array}{c} s'_1 \quad s'_2 \quad s'_3 \quad s'_4 \quad s'_5 \quad s'_6 \\ | \quad | \quad | \quad | \quad | \quad | \\ \circ - \circ - \circ - \circ - \circ - \circ \\ | \quad | \quad | \quad | \quad | \quad | \\ s_1 \quad s_2 \quad s_3 \quad s_4 \quad s_5 \quad s_6 \end{array} \quad (5)$$

In Section 2.3 we briefly discuss some details of how to define the MPO tensors which represent a given Hamiltonian, but for this section let us assume the Hamiltonian is given in MPO form and focus on optimizing the MPS approximation to the ground state.

Finding an MPS approximation to the ground state of H means that the MPS obeys the eigenvalue equation

$$\begin{array}{c} \circ - \circ - \circ - \circ - \circ - \circ \\ | \quad | \quad | \quad | \quad | \quad | \\ \circ - \circ - \circ - \circ - \circ - \circ \end{array} \simeq E_0 \begin{array}{c} \circ - \circ - \circ - \circ - \circ - \circ \\ | \quad | \quad | \quad | \quad | \quad | \\ \circ - \circ - \circ - \circ - \circ - \circ \end{array} \quad (6)$$

where E_0 is the smallest extremal eigenvalue of H .

A very efficient way to find extremal eigenvalues of Hermitian matrices is by using *iterative eigensolver algorithms* such as the Lanczos or Davidson algorithms. Without going into a detailed description of these algorithms, the main operation needed to perform them is the multiplication of the current approximate ground state wavefunction by H . The key idea of DMRG is to use an iterative eigensolver algorithm to improve the approximate ground state in MPS form, but *only one or two tensors at a time*—here for pedagogical reasons we will discuss the case of just improving one MPS tensor at a time.

When only improving one tensor of an MPS, the other temporarily frozen MPS tensors can be interpreted as defining a sub-basis of the full Hilbert space in which the unfrozen MPS tensor is defined. For a correct implementation of DMRG, transformations of the MPS tensors need to be carried out to ensure this sub-basis is an orthonormal basis, but here we will omit these steps and refer the interested reader to Ref. [4] for further discussion of this important point.

Let us say that we are currently improving the third tensor of an MPS defined on six sites. Then the step of multiplying H by this tensor in order to use an iterative algorithm such as Lanczos

³Hamiltonians which are not strictly local, such as in systems with long-range Coulomb interactions, can still be approximated well by MPOs using appropriate compression techniques.

to improve it is equivalent to performing the following tensor contractions

(7)

Note that the resulting tensor on the right-hand side above has the same index structure as the third MPS tensor, so that it is possible for the MPS tensor being optimized to obey an eigenvalue equation, when interpreting all of the other tensors—both the frozen MPS tensors and the MPO tensors—as a square “matrix” multiplying the unfrozen third MPS tensor. The reason the frozen MPS tensors appear on both the top and bottom of the left-hand side of the above expression is that they are acting on the Hamiltonian MPO as a transformation which changes the basis of H from the full Hilbert space to just the subspace of the indices of the third MPS tensor.

For the contractions in Eq. (7) to be efficient, it is important in practice to perform them in a certain order. To begin with, the projection of the Hamiltonian MPO tensors into the frozen MPS tensors is performed efficiently by iterating the following pattern of contractions, showing just the case of tensors to the right of the site being optimized:

(8)

A similar pattern is carried involving frozen MPS tensors to the left of the site to be optimized. The resulting “Hamiltonian projection” tensors are saved in memory for reuse in the iterative optimization loop and in later steps of DMRG when the optimization returns to the current site. Then, within the iterative eigensolver algorithm, an efficient pattern of contractions for carrying out one step of multiplication by H as in Eq. (7) is given by

(9)

where in the first transformation above the saved Hamiltonian projection tensors were recalled from memory.

Having performed just a few iterations of the eigensolver algorithm for the third MPS tensor (the iterative eigensolver should *not* be fully converged, since the other MPS tensors are not fully optimized), the DMRG algorithm continues by freezing the improved third tensor and next optimizing the fourth MPS tensor, then the fifth tensor, etc. until reaching the end of the system. Then the tensors are optimized one at a time in reverse order, until returning to the first site, completing what is called one sweep of DMRG. In cases where DMRG is very well suited for the problem, very accurate results can often be obtained in fewer than ten sweeps. But for challenging systems many more sweeps may be needed.

One other note about the convergence of DMRG is that the single-site algorithm outlined above may get stuck in a local minimum unless extra steps are included in the algorithm, such as using a noise term [11] or a subspace expansion step [12]. Both of these approaches usually lead to very robust convergence for a wide variety of systems. Another important and frequently used variant of DMRG involves optimizing two neighboring MPS tensors at a time. In addition to helping with convergence, optimizing two tensors together allows one to easily adapt the dimension of the bond between them, letting it grow or shrink as necessary to reach a desired accuracy goal while using a few parameters as possible.

Finally, it is important to consider the scaling of the DMRG algorithm when applying it to a given system. DMRG scales as $p_1 m^3 k + p_2 m^2 k^2$ where m is the wavefunction MPS bond dimension, k is the Hamiltonian MPO bond dimension, and p_1, p_2 are constant prefactors which depend on further implementation details. So although the leading cost is driven by the complexity of the wavefunction MPS, a very important driver of cost can also be the complexity of the Hamiltonian and how efficiently it can be represented as an MPO.

2.3 MPO forms of Hamiltonians

Though it is not necessary or always advantageous to represent the Hamiltonian in MPO form in order to carry out DMRG optimization, it can be very convenient to do so. For applications such as quantum chemistry, MPO techniques can also offer huge efficiency gains when used to compress the long-range Coulomb interaction terms, as discussed further in Section 4.3.2. While full discussion of how to construct MPO Hamiltonian representations is beyond the scope of this chapter, let us discuss one illustrative case to motivate MPO constructions.

Consider the one-dimensional Hubbard model, which shares some similarities with the Hamiltonians one encounters in quantum chemistry calculations, such as a Hilbert space of mobile electrons, as well as orbital-hopping and Coulomb interaction terms, though admittedly very local versions of such terms. Recall that the Hamiltonian of this model is

$$\hat{H} = -t \sum_{j\sigma} (\hat{c}_{j\sigma}^\dagger \hat{c}_{j+1\sigma} + \hat{c}_{j+1\sigma}^\dagger \hat{c}_{j\sigma}) + U \sum_j \hat{n}_{j\uparrow} \hat{n}_{j\downarrow}. \quad (10)$$

An exact MPO representation of the above Hamiltonian can be obtained by choosing the MPO

tensor on site j to have the form

$$\hat{W}_{r_{j-1}c_j} = \begin{bmatrix} \hat{I}_j \\ \hat{c}_{j\uparrow} \\ \hat{c}_{j\uparrow}^\dagger \\ \hat{c}_{j\downarrow} \\ \hat{c}_{j\downarrow}^\dagger \\ U\hat{n}_{j\uparrow}\hat{n}_{j\downarrow} \quad -t\hat{c}_{j\uparrow}^\dagger \quad t\hat{c}_{j\uparrow} \quad -t\hat{c}_{j\downarrow}^\dagger \quad t\hat{c}_{j\downarrow} \quad \hat{I}_j \end{bmatrix} \quad (11)$$

where matrix elements which are not shown are equal to zero. By saying an MPO tensor is equal to an operator-valued matrix as in the expression above, what is meant is that fixing the bond entries to a given pair of values (r, c) makes the resulting tensor, now carrying only two physical indices, equal to the operator listed in the operator-valued matrix. A few examples of fixing the bond indices on the MPO tensor defined by Eq. (11) above are

$$1 \begin{array}{c} s'_j \\ \circ \\ s_j \end{array} 1 = \hat{I}_{s'_j s_j} \quad 2 \begin{array}{c} s'_j \\ \circ \\ s_j \end{array} 1 = \hat{c}_{\uparrow s'_j s_j}^{s'_j} \quad 6 \begin{array}{c} s'_j \\ \circ \\ s_j \end{array} 2 = -t\hat{c}_{\uparrow s'_j s_j}^\dagger \quad (12)$$

To apply the particular MPO above to a finite system with open boundary conditions, the open bond index on the left of the first MPO tensor is contracted with the standard basis vector \mathbf{e}_6 and the open index on the right of the last MPO tensor is contracted with the standard basis vector \mathbf{e}_1 . The motivated reader can verify by constructing and fully contracting the MPO described above on small systems that it indeed reproduces the Hubbard model Hamiltonian.

Note that the size of the MPO tensor Eq. (11) is tied very closely to the number of distinct terms making up the Hamiltonian, such as up-spin hopping versus down-spin hopping terms. Terms with distinct coefficients that act on sites different distances apart also count as distinct terms for the purposes of determining the minimum size of an MPO representation. Since we will see below that discretized quantum chemistry Hamiltonians involve many such distinct terms with different ranges and coefficients, the size of MPO needed to represent them can grow very quickly with system size unless additional techniques or approximations are used.

For many more details about MPOs, their construction, and their use in quantum chemistry see Ref. [13].

3 Quantum chemistry: brief overview and discretization methods

There is a wide range methods in chemistry for studying atoms and molecules computationally, with large variations in the degree of approximation made. In the setting of *quantum chemistry*, one attempts a fully quantum treatment of either all the electrons, or at least those electrons most important for chemical processes.

It is common to work within the Born-Oppenheimer approximation, which we will do here. This approximation treats each atomic nucleus as a classical point object with positive charge Z , which is the atomic number. For each set of nuclear positions $\{\mathbf{R}_a\}$, with a indexing each atom, one solves the Schrödinger equation $\hat{H}|\Psi_0\rangle = E_0|\Psi_0\rangle$ to find the instantaneous ground state $|\Psi_0\rangle$ of the electrons. Here \hat{H} , $|\Psi_0\rangle$, and E_0 are all functions of the nuclear coordinates, with \hat{H} given by

$$\hat{H} = \sum_{\sigma} \int_{\mathbf{r}} \hat{\psi}_{\mathbf{r}\sigma}^{\dagger} \left(-\frac{1}{2} \nabla^2 + v(\mathbf{r}) \right) \hat{\psi}_{\mathbf{r}\sigma} + \frac{1}{2} \sum_{\sigma\sigma'} \int_{\mathbf{r}\mathbf{r}'} \hat{\psi}_{\mathbf{r}\sigma}^{\dagger} \hat{\psi}_{\mathbf{r}'\sigma'}^{\dagger} u(\mathbf{r}, \mathbf{r}') \hat{\psi}_{\mathbf{r}'\sigma'} \hat{\psi}_{\mathbf{r}\sigma}. \quad (13)$$

with the summation over $\sigma = \uparrow, \downarrow$ taken over up and down spin states, and with all integrations taken over the entire 3D space, unless stated otherwise.

The nuclear positions \mathbf{R}_a and atomic numbers Z_a enter through the one-body potential $v(\mathbf{r})$

$$v(\mathbf{r}) = \sum_a \frac{-Z_a}{|\mathbf{r} - \mathbf{R}_a|}, \quad (14)$$

parameterizing the Coulomb attraction of the electrons to the protons of each atom a . The function $u(\mathbf{r}, \mathbf{r}')$ parameterizes the Coulomb repulsion between electrons

$$u(\mathbf{r}, \mathbf{r}') = \frac{1}{|\mathbf{r} - \mathbf{r}'|}. \quad (15)$$

This Hamiltonian is known as the *electronic structure* Hamiltonian, and solving for its ground state or other properties known as the *electronic structure problem*.

In general, the electronic structure problem is very difficult. One reason is that it involves many strongly interacting fermions. It is also a three dimensional problem in the continuum, raising the challenging issue of how best to discretize it for methods which operate in a discrete Hilbert space, with DMRG being one such method.

A wide variety of methods have been devised to study electronic structure. Density functional theory is one common approach, especially when the electronic structure problem must be solved at a relatively low cost within other algorithms such as molecular dynamics simulations. Another method is coupled cluster, which often serves as the standard for high-accuracy studies of small molecules. Other important methods include variants of configuration-interaction [14, 15] and quantum Monte Carlo within a fixed-node approximation [16, 17]. And of course there is DMRG which is the focus of this chapter [7, 8].

One common approach in electronic structure is to divide valence electrons from core electrons—those whose orbitals are occupied with probability essentially equal to one—then remove the core electrons and appropriately modify the one-electron potential $v(\mathbf{r})$ by adding a so-called pseudopotential. But in what follows, we will only consider an all-electron approach for simplicity, meaning we will give every electron a fully quantum mechanical treatment.

3.1 Basis set discretization

A very common way to discretize the electronic structure problem is to project the single-particle basis onto a finite set of functions. The set of functions used is called the basis set. If

the initial set of functions are not normalized and orthogonal to each other (under the 2-norm), which is often the case for standard basis sets, then one typically expresses the discretized Hamiltonian in a set of orthonormal functions obtained by an transformation of the original functions. Here by basis set, we will usually mean the final set of orthonormal functions obtained after the transformation.

Consider a basis set given by N orthonormal functions $\{\phi_i(\mathbf{r})\}$ where $i = 1, \dots, N$. Then this basis set can be used to discretize the electronic structure Hamiltonian by defining the integrals t_{ij} and V_{ijkl} , where

$$t_{ij} = \int_{\mathbf{r}\mathbf{r}'} \phi_i(\mathbf{r}) \left(-\frac{1}{2}\nabla^2 + v(\mathbf{r}) \right) \phi_j(\mathbf{r}') \quad (16)$$

$$V_{ijkl} = \int_{\mathbf{r}\mathbf{r}'} \frac{\phi_i(\mathbf{r})\phi_j(\mathbf{r}')\phi_k(\mathbf{r}')\phi_l(\mathbf{r})}{|\mathbf{r} - \mathbf{r}'|}. \quad (17)$$

One also defines orbital annihilation operators

$$\hat{c}_i = \int_{\mathbf{r}} \phi_i(\mathbf{r}) \hat{\psi}_{\mathbf{r}} \quad (18)$$

and conjugate orbital creation operators \hat{c}_i^\dagger .

With these definitions, the electronic structure Hamiltonian projected into the basis set is

$$\hat{H} = \frac{1}{2} \sum_{ij\sigma} t_{ij} \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} + \frac{1}{2} \sum_{ijkl\sigma\sigma'} V_{ijkl} \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma'}^\dagger \hat{c}_{k\sigma'} \hat{c}_{l\sigma}. \quad (19)$$

In this form, the Hamiltonian is in principle straightforwardly treatable by lattice methods such as DMRG, though it is crucial to make sure that the very large number of terms involving the V_{ijkl} interaction integrals are treated efficiently. To apply DMRG to a complicated lattice Hamiltonian such as the one above, one must choose some one-dimensional ordering of the basis. Within this ordering, the system can be viewed as a one-dimensional chain with non-local interactions. Which ordering to choose is not obvious, but different heuristics can be used to determine very good orderings, such as an analysis of the entanglement of the Hartree-Fock approximation to the ground state.

Regarding scaling, observe there are N^4 interaction integrals V_{ijkl} for a basis set of size N . Thus quantum chemistry calculations using basis sets have a cost that grows at least as N^4 , and often much more rapidly, with the total number of basis functions N . However we will see in Section 4 that one can obtain much better scaling using local bases within DMRG for certain classes of systems.

3.1.1 Gaussian basis sets

Because integrals involving Gaussian functions admit exact closed-form expressions, even when the integrand involves Gaussians, derivatives of Gaussians, and additional polynomial factors, basis sets built from three-dimensional Gaussians are a very common choice in quantum chemistry due to the efficiency they offer when performing integrals, Eqs. (16) and (17), to construct the Hamiltonian. For computational scientists used to working with lattice models such as the

Hubbard model, it can be surprising to learn that just constructing the Hamiltonian for a quantum chemistry calculation can take significant time, but note that for $N = 100$ basis functions, there are a hundred million V_{ijkl} terms, each defined by a six-dimensional integral!

Despite their name, the basis functions in a Gaussian basis set are not necessarily themselves Gaussians, but are made by summing (or “contracting”) a small number of Gaussian functions together, then possibly multiplying them by integer powers of x , y , and z for those functions designated as capturing P or D orbitals. The purely Gaussian functions which get summed together are called primitive Gaussians.

Standard and popular Gaussian basis sets are designated by acronyms such as STO-3G or cc-pVDZ (standing for “Slater-type orbitals fit to three Gaussians” or “correlation-consistent, polarized valence double-zeta”). These standard basis sets differ not only in the specific Gaussian functions they contain, but also how these functions were chosen and for what purpose. For example, the correlation consistent (cc-) basis sets were constructed for the purpose of extrapolating energies smoothly to the continuum limit, which in chemistry parlance is called the *complete basis set* limit.

The typical procedure for using a Gaussian basis set, which generally consists of groups of non-orthogonal functions centered on each atom, is to compute a set of orthogonal functions $\phi_j(\mathbf{r})$ from each of the functions $b_i(\mathbf{r})$ in the initial non-orthogonal basis. For example, one can diagonalize the overlap matrix $\mathcal{O}_{ij} = \int_{\mathbf{r}} b_i(\mathbf{r})b_j(\mathbf{r})$ and use the resulting orthogonal matrix of eigenvectors to define the $\phi_j(\mathbf{r})$ basis. Then the same transformation can be used to transform the integrals t_{ij} and V_{ijkl} to the orthonormal basis. It is important to note that although the original $b_i(\mathbf{r})$ are somewhat local and are centered on one of the atoms, the orthonormal $\phi_j(\mathbf{r})$ functions typically end up being much less localized. Though various strategies can be used to ensure the ϕ_j are as local as possible, in general they will have significant overlap with many other basis functions ϕ_k , which has negative consequences when used as a starting point for methods based on locality such as DMRG.

3.1.2 Quantum chemistry DMRG with Gaussian basis sets

The use of DMRG for quantum chemistry originates from the proposal of White and Martin [7] to use DMRG in combination with standard Gaussian basis sets, as well as an earlier study [18]. Since then, DMRG has become a powerful technique for certain quantum chemistry problems, in large part due to its ability to handle strongly correlated systems, as it does not rely on an expansion of the wavefunction in Slater determinants unlike many other quantum chemistry methods. Some notable examples of applying DMRG within Gaussian basis are an accurate study of the strongly-correlated Mn_4CaO_5 cluster in the photosystem II protein complex [19], and calculations of the challenging Cr_2 dimer [20].

Many technical improvements to the DMRG algorithm have been developed in the setting of chemistry with Gaussians or adapted to this setting. These include using the “complementary operator” technique to handle very large Hamiltonians with non-local interactions [21, 7], exploiting $SU(2)$ symmetries [22], and using matrix product operators (MPO) to compress

Hamiltonians and simplify calculations [13]. Another important step often used in conjunction with DMRG is the selection of an active space of orbitals, which are a subset of the full basis set most important for estimating chemical properties [19]. Advanced DMRG and MPS techniques for going beyond ground-state calculations have also been developed and refined in the quantum chemistry setting, such as tangent-space linear response methods [23]. For review articles about using DMRG in quantum chemistry with Gaussian basis sets, see Refs. [24], [8], and [25].

3.2 Grid discretization

An alternative to using a basis set to discretize the continuum Hamiltonian Eq. (13) is to use a *grid discretization*. Within a grid approximation to the continuum, one introduces grid points $\mathbf{r}_\mathbf{n}$ indexed by a set of integers $\mathbf{n} = (n_x, n_y, n_z)$ where $n_x = 1, 2, \dots, N_x$ and similar for y and z . Rather than associating individual functions to these grid points, one thinks of them as locations at which to sample from any smooth function. Operators in the continuum Hamiltonian are replaced by operators defined only on grid points, such that for any sufficiently smooth wavefunction, the expected values of the grid operators accurately approximate that of the continuum operators.

For terms not involving derivatives, the replacement of continuum operators $\hat{\psi}_{\mathbf{r}\sigma}$ with grid operators $\hat{c}_{\mathbf{n}\sigma}$ is

$$\hat{\psi}_{\mathbf{r}\sigma} \rightarrow \sum_{\mathbf{n}} \delta(\mathbf{r} - \mathbf{r}_\mathbf{n}) \hat{c}_{\mathbf{n}\sigma}. \quad (20)$$

Thus for terms such as the one-body potential energy, the transformation from continuum to grid form is

$$\frac{1}{2} \sum_{\sigma} \int_{\mathbf{r}} v(\mathbf{r}) \hat{\psi}_{\mathbf{r}\sigma}^{\dagger} \hat{\psi}_{\mathbf{r}\sigma} \rightarrow \frac{1}{2} \sum_{\mathbf{nn}'\sigma} \int_{\mathbf{r}} v(\mathbf{r}) \delta(\mathbf{r} - \mathbf{r}_\mathbf{n}) \delta(\mathbf{r} - \mathbf{r}'_{\mathbf{n}'}) \hat{c}_{\mathbf{n}\sigma}^{\dagger} \hat{c}_{\mathbf{n}'\sigma} = \frac{1}{2} \sum_{\mathbf{n}\sigma} v(\mathbf{r}_\mathbf{n}) \hat{c}_{\mathbf{n}\sigma}^{\dagger} \hat{c}_{\mathbf{n}\sigma} \quad (21)$$

from which we see that the basic replacement is just to evaluate the coefficient functions at grid points and replace continuum with grid operators.

For the Coulomb interaction term, the transformation to the grid is

$$\frac{1}{2} \sum_{\sigma\sigma'} \int_{\mathbf{r}\mathbf{r}'} u(\mathbf{r}, \mathbf{r}') \hat{\psi}_{\mathbf{r}\sigma}^{\dagger} \hat{\psi}_{\mathbf{r}'\sigma'}^{\dagger} \hat{\psi}_{\mathbf{r}'\sigma'} \hat{\psi}_{\mathbf{r}\sigma} \rightarrow \frac{1}{2} \sum_{\mathbf{n}\neq\mathbf{n}'} u(\mathbf{r}_\mathbf{n}, \mathbf{r}_{\mathbf{n}'}) \hat{n}_\mathbf{n} \hat{n}_{\mathbf{n}'} \quad (22)$$

which gives a particularly simple and compact form for this computationally expensive term. Crucially, unlike the basis set approach which results in N_b^4 discrete Coulomb terms for a basis set consisting of N_b functions, the number of Coulomb terms in the grid approach is just N^2 where $N = N_x N_y N_z$ is the number of grid points. So the scaling of the number of Coulomb integrals is much more favorable in the grid approach than in the basis-set approach, though this is not the entire story as we will see just below.

Finally, for the kinetic energy term of the Hamiltonian, in the grid approach one replaces derivatives by finite-difference approximations over neighboring grid points. There are many finite-difference approximations for the second derivative operator, which can be obtained by approaches such as fitting low-order polynomials or using wavelet techniques. For a regular grid

with grid spacing a , the resulting transformation of the kinetic energy takes the form

$$-\frac{1}{2} \sum_{\sigma} \int_{\mathbf{r}} \hat{\psi}_{\mathbf{r}\sigma}^{\dagger} \nabla^2 \hat{\psi}_{\mathbf{r}\sigma} \rightarrow -\frac{1}{2a^2} \sum_{\mathbf{nn}'\sigma} \Delta_{\mathbf{nn}'} \hat{c}_{\mathbf{n}\sigma}^{\dagger} \hat{c}_{\mathbf{n}'\sigma} \quad (23)$$

where the $\Delta_{\mathbf{nn}'}$ are finite-difference coefficients which fall exactly to zero once \mathbf{n} and \mathbf{n}' are more than a certain distance apart, depending on the order of the approximation.

3.3 Comparison of basis set and grid discretization

Having just reviewed two of the major approaches to discretizing continuum Hamiltonians—Gaussian basis set discretization and grid discretization—let us contrast the key aspects of each approach.

One important consideration is whether the resulting approximation is variational, meaning whether the ground-state energy of the approximate Hamiltonian is always greater than or equal to the continuum ground-state energy. A key advantage of the basis set approach is that it is variational. Grid discretization is typically not variational.

The other key consideration is the overall computational cost, with the number of Coulomb interaction terms being a major driver of cost. The number of Coulomb terms scales much better within the grid approach: just N_g^2 Coulomb terms for N_g grid points, versus N_b^4 Coulomb terms when using a basis set of N_b functions. However, Gaussian basis sets often need many fewer functions N_b compared to the number of grid points N_g needed to reach a similar accuracy. For example, approximating a small molecule with about $N_b = 100$ Gaussian basis functions requires a grid of linear size roughly $N_{x,y,z} = 100$ giving $N_g = 10^6$ grid points in total, making the grid approach more costly overall, despite its better scaling.

Thus at least for the study of small molecules using high-accuracy quantum chemistry techniques, Gaussian basis set discretization is typically the preferred approach due to its variational nature and relatively low costs.

But as we will see in the next section, one can successfully combine the best features of the grid and basis set approaches—low cost and good scaling—for use within methods such as DMRG.

4 Local bases for quantum chemistry DMRG

We turn now to a promising approach of using basis sets made from *local* functions for DMRG and tensor network methods for quantum chemistry. These bases are distinct from standard Gaussian basis sets, though the first one we will discuss uses Gaussian bases as an ingredient. Although choosing a local basis often results in the basis being larger than the number of Gaussians needed to resolve the continuum to the same accuracy, we will see that the tradeoff is worthwhile because tensor networks very naturally exploit real-space locality.

The motivation for developing alternative bases for DMRG calculations is that calculations using Gaussian basis sets scale poorly compared to DMRG calculations of lattice systems, such as of the 1D Hubbard model, which have local Hamiltonian terms only. For example, quantum

chemistry DMRG calculations of one-dimensional chains of 3D hydrogen atoms can only handle about a hundred atoms using Gaussian basis sets, whereas 1D Hubbard DMRG calculations can scale to thousands of sites. Yet both quantum chemistry calculations and Hubbard model calculations use the same local single-site or single-orbital basis: $\{|0\rangle, |\uparrow\rangle, |\downarrow\rangle, |\uparrow\downarrow\rangle\}$. Thus it follows that scaling is not about whether the degrees of freedom are electrons, but instead must originate from either the range or complexity of the Hamiltonian.

To be more precise, there are three significant factors which determine the cost of DMRG calculations for quantum chemistry:

1. the *size of the Hamiltonian* within the particular basis and format used within DMRG
2. the *bond dimension* of the MPS needed to accurately represent the ground state wavefunction in the basis used
3. the *number of sweeps*, or passes, over the system necessary to reach convergence

Crucially, factors number (1) and (2) are both tied very strongly to the size and locality of the basis one uses. For example, if one uses a basis set which is local in the sense that any function ϕ_i only overlaps with a small, finite number of other functions ϕ_j , then the number of non-zero interaction integrals V_{ijkl} scales only as N^2 , not as N^4 . This better scaling has immediate implications for the costs (1) coming from the size of the Hamiltonian. Regarding the costs (2) associated with the MPS bond dimension, a ground state in a spatially local basis can be accurately captured by an MPS of a bond dimension growing only logarithmically with the largest direction of a system extended along one dimension. In contrast, a basis that is not spatially local can result in MPS bond dimensions which are orders of magnitude larger, with the most extreme example being that of an interacting system in a non-local plane-wave basis, for which an accurate MPS representation must have a bond dimension growing exponentially with system size!

Yet one does not immediately gain from just choosing a basis set of local functions because such local bases can require many more functions N to capture the continuum than a non-local basis. But with a smart choice of local functions (Sections 4.1 and 4.2), and technology for representing the electronic structure in a compressed form (Section 4.3.2), local basis sets can be a very beneficial choice.

4.1 Approach 1: hybrid grid and basis set, or “sliced basis” approach

One idea to develop a local basis for DMRG is to combine grid discretization with Gaussian basis set discretization, with the goal of obtaining the best aspects of each. A successful way to combine a grid and basis set is to use a grid only along one spatial direction, such as the z direction, then a Gaussian basis set along the other two directions x and y . This idea, called *sliced-basis*, was recently developed for use with DMRG in Ref. [26] and successfully applied to DMRG calculations of one-dimensional, strongly correlated chains of hydrogen atoms in Refs. [26, 27].

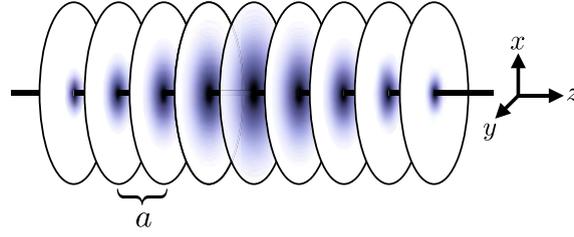


Fig. 1: Visualization of a sliced-basis representation of the continuum, where the z direction is discretized using a grid spacing a and the (x, y) directions are captured using a set of functions which can be derived from a 3D Gaussian basis.

To carry out the grid discretization along just the z direction, define a regular grid spacing a and grid points $z_n = n \cdot a$. Each value of n defines an infinite two-dimensional plane (x, y, z_n) we will call a *slice* of three-dimensional space. Next, within each continuum slice, discretize using a basis set of functions $\{\varphi_{nj}(x, y)\}$ with $j = 1, 2, \dots, N_o$. The number of functions N_o used on each slice could differ from slice to slice in principle, but here we consider just the case where N_o the same for every slice. Fig. 1 shows schematically how a sliced basis might represent a three-dimensional continuum function such as an electron orbital as linear combinations of two-dimensional functions arranged on planes separated by a small spacing a in the z direction. Although such a discretization does not strictly speaking correspond to using a basis set, it is still pedagogically useful to think of it as involving a basis set given by

$$\phi_{nj}(\mathbf{r}) = \varphi_{nj}(x, y) \delta^{1/2}(z - a \cdot n) \quad (24)$$

such that these ‘‘sliced basis functions’’ are ultra-local along the z -direction. The $1/2$ power on the delta function simply indicates that the ϕ are square normalized. The reason the sliced-basis approach is not technically a basis set approach is that it uses a finite-difference approximation for the z -direction kinetic energy as we will discuss below. Thus it lacks the guarantee of a variational energy. But it can still be useful to picture it in terms of the functions Eq. (24).

After introducing the grid and basis set approximations, the discrete electronic structure Hamiltonian takes the form

$$\hat{H} = \frac{1}{2} \sum_{nn'ij\sigma} t_{ij}^{nn'} \hat{c}_{ni\sigma}^\dagger \hat{c}_{n'j\sigma} + \frac{1}{2} \sum_{nn'ijkl\sigma\sigma'} V_{ijkl}^{nn'} \hat{c}_{ni\sigma}^\dagger \hat{c}_{j\sigma'}^\dagger \hat{c}_{k\sigma'} \hat{c}_{l\sigma} . \quad (25)$$

where the slice or grid indices run over $n, n' = 1, 2, \dots, N_z$ and the orbital indices over $i, j, k, l = 1, 2, \dots, N_o$. Note the resemblance to Eq. (19), except that the one-body integrals $t_{ij}^{nn'}$ and two-body interaction integrals $V_{ijkl}^{nn'}$ carry extra indices n, n' labeling pairs of slices along the z direction. The one-body integrals are defined as

$$t_{ij}^{nn'} = \delta_{nn'} \int_{\rho} \varphi_{ni}(\rho) \left(-\frac{1}{2} \nabla_{\rho}^2 + v(\rho, z_n) \right) \varphi_{n'j}(\rho) - \delta_{ij} \frac{1}{2a^2} \Delta_{nn'} , \quad (26)$$

where $\rho = (x, y)$ is a convenient shorthand, and $\Delta_{nn'}$ are second-derivative finite-difference coefficients. The two-body interaction integrals are defined as

$$V_{ijkl}^{nn'} = \int_{\rho, \rho'} \frac{\varphi_{ni}(\rho) \varphi_{n'j}(\rho') \varphi_{n'k}(\rho') \varphi_{nl}(\rho)}{\sqrt{|\rho - \rho'|^2 + (z_n - z_{n'})^2}} . \quad (27)$$

4.1.1 Scaling of the sliced-basis approach

Despite the resemblance to the basis-set form of the electronic structure Hamiltonian Eq. (19), a key distinctive of the sliced-basis approach is that the number of interaction integrals $V_{ijkl}^{nm'}$ scales as $N_z^2 N_o^4$, with N_z the number of slices and N_o the number of orbitals on each slice. When treating quasi-1D, chainlike molecules it is only the number of slices N_z that grows with the number of atoms, not N_o . Therefore the overall scaling of the method as a function of the number of atoms is quadratic, which is a huge improvement over the quartic scaling with the number of atoms incurred by the basis set approach. We discuss in Section 4.3.2 how the z -locality of the sliced-basis approach allows its quadratic scaling to be improved to just *linear* scaling when taking advantage MPO compression techniques. Thus the sliced-basis approach realizes some of the best aspects of a grid while avoiding the pitfall of the huge number of grid points arising from a fully 3D grid. By using a grid only along one direction, the overall number of interaction integrals that must be treated remains tractable.

4.1.2 Deriving the transverse basis functions

A crucial step in setting up a sliced basis for a chemistry calculation is selecting or deriving the transverse functions $\{\varphi_{nj}(x, y)\}$ which define the basis of each slice. While many approaches could be conceived to do this, the approach taken in Ref. [26] derives the transverse functions from a standard, atom-centered Gaussian basis set.

In the simplest setting, all of the atoms are of the same type (such as hydrogen atoms) and the “parent” Gaussian basis set consists purely of Gaussians (that is, the basis functions are “uncontracted”). This can be made to be true by just using the primitive Gaussians from a set without contracting them. Because we assumed the atoms are the same, the entire basis consists of the same set of Gaussian functions centered on each atom. Then for a given basis function $b_i(x, y, z)$ the associated transverse function intersecting a slice is just $\tilde{\varphi}_{ni}(x, y) = b_i(x, y, z_n)$. Because in this case $b_i(\mathbf{r})$ is a Gaussian $b_i(\mathbf{r}) \propto \exp(-\zeta_i |\mathbf{r} - \mathbf{r}_i|^2)$, its restriction to slice n will also be a Gaussian with the same length scale ζ_i . Thus although every basis function for every atom intersects with a given slice, there will only be as many unique Gaussian functions as in the basis set of just a single atom. The final transverse basis of each slice is formed by just creating an orthonormal linear combination of the $\tilde{\varphi}_{ni}$.

More generally, though, the atoms will be of different types and some of the basis functions will be contracted, so their restrictions to slice will result in large set of functions, much larger than was needed to represent any one atom in the original Gaussian basis. Using all of these functions would be costly and would not give enough extra accuracy relative to the other approximations being made to be worthwhile. So for this more general case one must truncate the functions on a slice in some sensible way. A very reasonable and effective approach is to use *principal component analysis* (PCA). Here this just means the following: say that we obtain a set of functions $\tilde{b}_{ni}(x, y) = b_i(x, y, z_n)$ by restricting each function b_i from a Gaussian basis set to the

n^{th} slice. Compute the overlap matrix

$$\mathcal{O}_{ij} = \int_{x,y} \tilde{b}_{ni}(x,y) \tilde{b}_{nj}(x,y) . \quad (28)$$

Now diagonalize this matrix and sort its eigenvalues from largest to smallest. Keep the eigenvectors v_k^j corresponding to the largest $k = 1, 2, \dots, N_o$ eigenvalues, where for the case of molecules consisting of only one atom type, N_o can be chosen to be the number of functions in the basis set of a single atom. The resulting eigenfunctions take the form

$$\tilde{\varphi}_{nk}(x,y) = \sum_j v_k^j \tilde{b}_{nj}(x,y) . \quad (29)$$

The final transverse basis $\{\varphi_{ni}(x,y)\}_{i=1}^{N_o}$ of slice n is then found by normalizing the $\tilde{\varphi}$.

4.1.3 The sliced-basis approach in practice

To test the sliced-basis approach outlined above, Refs. [26, 27] applied sliced bases to finding the ground state of one-dimensional chains of hydrogen atoms with their nuclei evenly spaced by a distance R . Although hydrogen atoms lack core orbitals, solving hydrogen chains still involves reckoning with most of the issues that make quantum chemistry challenging. These issues include scaling to large numbers of atoms, converging to the continuum or complete basis set limit, and dealing with strongly correlated wavefunctions.

One goal of studying hydrogen molecules was to compare to results obtained with standard Gaussian basis sets such as those in Ref. [27]. For this purpose, it is sufficient to study finite hydrogen chains of ten atoms or H_{10} —see the results in Fig. 2. The energy obtained using a sliced basis derived from a given Gaussian basis is similar to the energy obtained just using that Gaussian basis without slicing. This is encouraging to see, since it demonstrates that the sliced-basis maintains favorable aspects of the parent Gaussian basis like the ability to smoothly extrapolate to the continuum or complete basis set limit. On the other hand, a sliced basis is much more scalable to long hydrogen chains as we will see below. Another observation about the results in Fig. 2 is that the sliced-basis energies are generally slightly lower than the energy of the corresponding Gaussian basis, at least for larger basis sets such as cc-pVDZ (double ζ) and cc-pVTZ (triple ζ). This can be readily understood as a consequence of finer resolution of a sliced basis along the z direction, allowing electrons more freedom to avoid costly Coulomb interactions.

Another goal of studying hydrogen chains with the sliced-basis approach was to test that it can scale to very long systems extended along the z direction, with a cost that is only linear in system length. Here a crucial technical step for achieving good scaling is the compression of the Coulomb interaction using matrix product operator techniques, which we describe later in Section 4.3.2 below. Taking a fixed inter-atomic spacing $R = 3.6$ and working with a sliced basis derived from the STO-6G basis set, Fig. 3 shows results for the timing and energy of calculations up to 1000 hydrogen atoms. From the inset of the figure, one can see very close to linear scaling of the method with number of atoms, while getting consistently accurate energies across all system sizes.

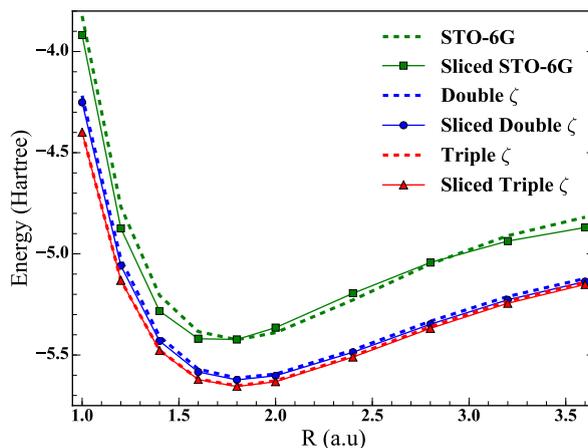


Fig. 2: Ground-state energies of H_{10} chains as a function of inter-atomic spacing R calculated using DMRG within standard Gaussian basis sets (dashed curves) and sliced basis sets (solid curves and points) using a uniform grid spacing of $a = 0.1$ atomic units [26].

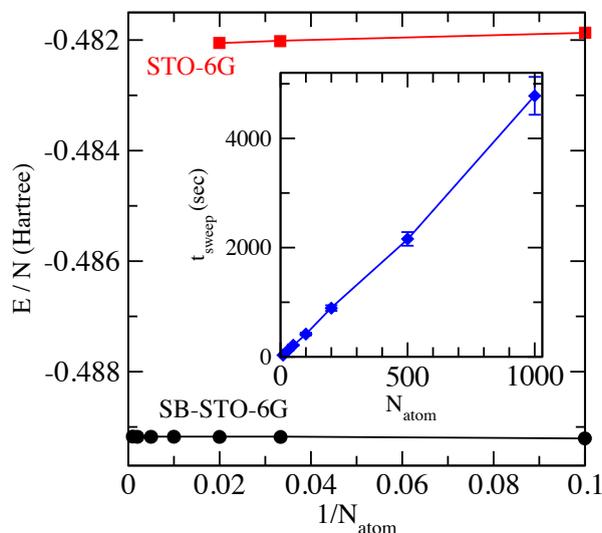


Fig. 3: Scaling with number of atoms of sliced-basis calculations up to 1000 hydrogen atoms. The inter-atomic spacing is fixed to $R = 3.6$ and a sliced basis derived from the STO-6G Gaussian basis was used. The outer plot shows the ground state energy from DMRG using the standard STO-6G basis and the sliced version (SB-STO-6G). The inset shows the average time per DMRG sweep, taking a bond dimension of $m = 100$.

4.2 Approach 2: multi-sliced gausslet basis

The sliced basis approach to discretizing the electronic structure Hamiltonian demonstrates a successful marriage of the grid and basis set approaches to quantum chemistry. Counterintuitively, it demonstrates that using *more* functions to represent the Hamiltonian can result in a more affordable calculation overall, by choosing the functions to be local (at least along one direction), so that the DMRG algorithm and MPO methods for compressing the Hamiltonian (Section 4.3.2) can perform to their full potential.

There are some drawbacks to the sliced-basis approach though. It is very oriented toward one-dimensional systems, and only scales well along the z direction: the direction discretized with a grid. The cost of including more transverse functions within each (x, y) slice is high. The sliced-basis also requires a small grid spacing a to get good accuracy. Finally, because sliced-basis transverse functions are derived from standard Gaussian bases, they sometimes inherit their weaknesses when dealing with certain molecules for which they were not designed. This can include molecules where the nuclei are extremely close together or for which electrons occupy very spatially extended orbitals.

Because these issues mostly stem from the use of Gaussian bases in making the transverse functions, a better approach would be to start from a completely different set of basis functions. Ideally such functions would behave like grid points in terms of the simplicity of the Hamiltonian formed from them, yet a relatively small number of them would be able to capture continuum details. Fortunately such functions have been recently developed for quantum chemistry applications and are called *gausslets* [28]. Gausslet basis sets have been extended to real 3D chemistry calculations based on DMRG through an approach called *multi-slicing* [29]. We will discuss both gausslets and multi-slicing below, with examples of the improvements they give over the sliced-basis approach. A key reason for these improvements is not only that multi-sliced gausslets capture the continuum with relatively few functions, but that their properties enable *diagonal approximations* for the costly Coulomb interaction terms.

4.2.1 Gausslet functions and their properties

In the one-dimensional case, a gausslet is a function $G(x)$ which is symmetric about $x = 0$ and which is orthonormal $\int_{x,x'} G(x) G(x') = \delta(x - x')$. Gausslets are also smooth, in the precise sense of being orthogonal to a certain hierarchy of oscillatory functions; they are local in the sense of falling rapidly to zero past a certain length scale; they have excellent completeness properties, meaning that linear combinations of neighboring gausslets can represent any polynomial up to a certain very high order (such as order 10); and finally gausslets have an important property of integrating like a delta function when integrated with sufficiently smooth functions. What this means is that for any polynomial $p(x)$ that is not too high-order

$$\int_{-\infty}^{\infty} dx G(x - x') p(x) = p(x'). \quad (30)$$

Thus integration against a gausslet “plugs in” the coordinate where the gausslet is centered. All of these desirable properties of gausslets resemble those of wavelets (technically wavelet scaling functions), yet underneath a gausslet is defined as a weighted sum of Gaussian functions, hence the name gausslet. Defining gausslets in terms of underlying Gaussians makes them very convenient and efficient for the integrations necessary to perform when constructing quantum chemistry Hamiltonians.

Figure 4 shows an array of one-dimensional gausslet functions with a length scale of 1.0, such that their centers are arranged on a grid with a 1.0 spacing. Gausslets can be constructed in various ways involving trade-offs in their favorable properties listed above. For more information

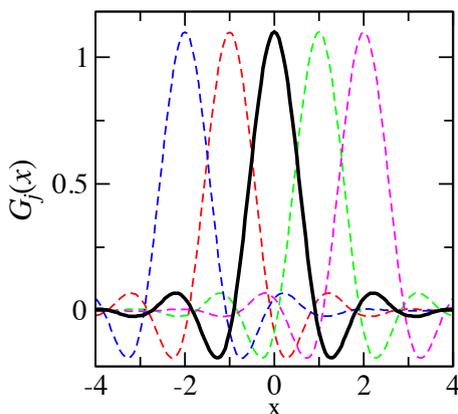


Fig. 4: One-dimensional array of gausslet functions with a length scale of 1.0. The gausslet centered at the origin is highlighted in solid black to emphasize details.

about the details of constructing them, see Ref. [28] which proposes and constructs gausslets, building upon the development in Ref. [30] of compact and symmetric families of orthogonal wavelets.

Importantly, after constructing a grid of gausslets like in Fig. 4 and using them to discretize quantum chemistry Hamiltonians, one can obtain very accurate continuum results using a gausslet spacing of about 1.0, in contrast to grid discretization which requires about an order of magnitude smaller spacing to obtain similar accuracy [28]. Better yet, we will see next that by adapting the grid on which the gausslets are centered, one can even better resolve the continuum using small numbers of gausslet functions.

4.2.2 Adapted grid of gausslets

Using an even-spaced grid of gausslet functions, as in Fig. 4 to discretize quantum chemistry Hamiltonians is more efficient than using a simple grid, yet still requires more functions than are actually needed. The reason is that while high resolution is required to capture details of the electronic wavefunction near atomic nuclei, much less resolution is needed away from nuclei. A straightforward way to reduce the number of functions needed while preserving high resolution near nuclei is to perform a coordinate mapping on the gausslet functions so that they form an adapted grid, with a finer spacing near nuclei and a coarser spacing otherwise.

Such a coordinate mapping may be defined via a function $x(u)$ which maps from a fictitious space u where the gausslets are defined to have a regular grid spacing into the actual space x used for the quantum chemistry calculation. Let $u(x)$ be defined as the inverse of the mapping $x(u)$. For the case of a single atom, a sensible coordinate mapping is

$$u(x) = \frac{1}{s} \sinh^{-1}(x/a) \quad (31)$$

defined by a *scale* parameter s and a *core cutoff* parameter a . Figure 5(a) shows how this mapping takes an evenly spaced grid along the y direction of the plot into a variable spaced grid along the x direction.

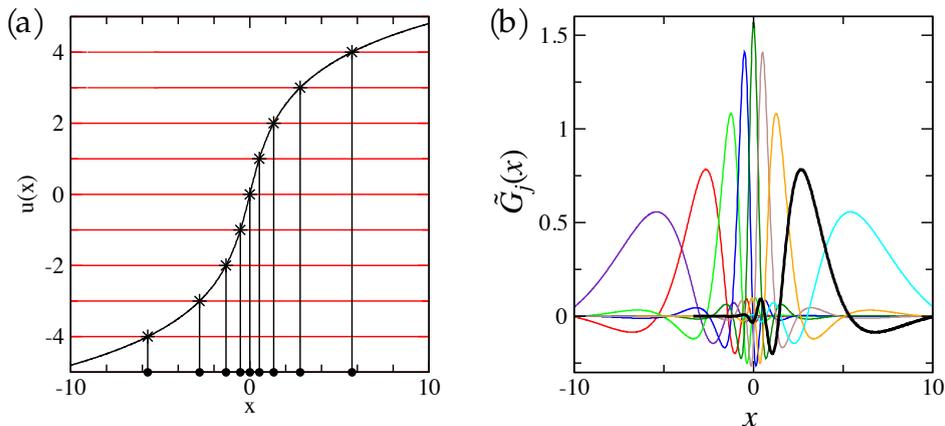


Fig. 5: Illustration of (a) a coordinate mapping Eq. (31) which maps a regularly spaced grid in the u coordinates to an adapted grid in the x coordinates. The resulting adapted gauslets shown in (b) remain orthonormal, but achieve a higher resolution near the origin. One gauslet is shown with a bold line to highlight details.

Having chosen a coordinate mapping, the transformation of the gauslets which moves their centers onto the adapted, variable-spacing grid while preserving their orthogonality and other good properties is

$$\tilde{G}_j(x) = G_j(u(x)) \sqrt{u'(x)} \quad (32)$$

where $G_j(u)$ is the gauslet centered at the integer grid point j in the u space. Figure 5(b) shows the adapted gauslets $\tilde{G}_j(x)$ resulting from using the coordinate mapping $u(x)$ defined in Eq. (31) above. One of the adapted gauslets is highlighted with a bolder line, and you can observe that it takes a distorted shape compared to the unadapted gauslets in Fig. 4. The placement of more and finer-sized gauslets near the origin gives better resolution there.

Note that when adapting gauslets for systems of multiple atoms, there are modifications of the transformation Eq. (31) which make it better suited for treating molecules. The supplemental information of Ref. [29] discusses such multi-atom coordinate transformations.

4.2.3 Multi-sliced grid

To apply the above ideas of gauslet basis sets to 3D systems, the most straightforward approach is to define basis functions as products $G_i(x) G_j(y) G_k(z)$ of 1D gauslets. But how to maintain this product form while also adapting the gauslet spacing near atomic nuclei is less obvious; for example, performing the coordinate transformations in a radially symmetric way destroys the product form of the 3D functions, resulting in integrals which are too costly when constructing the discrete Hamiltonian.

Fortunately, there is a simple way around this problem that only incurs a modest overhead in the total number of functions needed. This workaround is called *multi-slicing* and is just the idea of performing the coordinate transformation sequentially: first in the z direction (the direction along the greatest extent of the system), then in the y direction, and finally the in x direction.

In more detail, one starts by first defining a coordinate transformation $u^z(z)$ to determine an

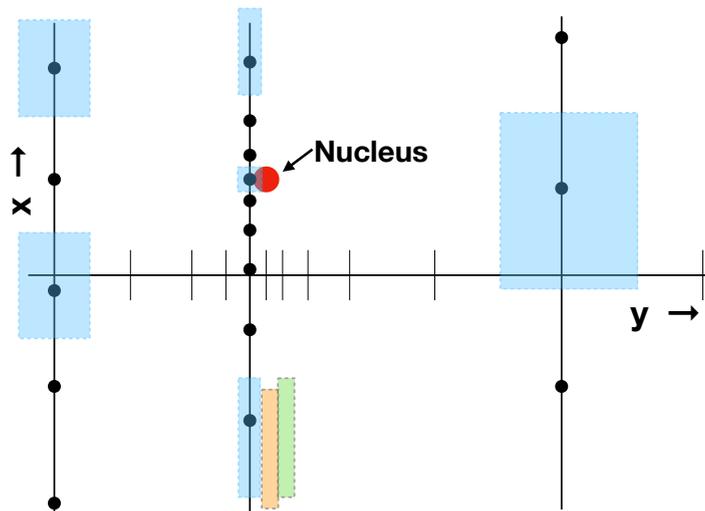


Fig. 6: Illustration of a fixed- z cut through a multi-sliced grid, showing the primary non-zero support of selected gausslet functions as colored rectangles. The vertical lines are selected x -slices and points are centers of adapted gausslets. The position of a nucleus is shown, illustrating how the multi-sliced grid bunches more gausslets of a smaller size nearby.

adapted grid of discrete z values z_k , $k = 1, 2, \dots, N_z$ which are more closely spaced whenever planes $z = z_k$ pass nearby atomic nuclei. This transformation defines planes $z = z_k$ with adapted functions $\tilde{G}_k(z)$ centered on them, forming what is called a z -slice. Now within each z -slice, a coordinate transformation is applied to the y coordinates, defining an adapted grid through a function $u^y(y)$ which yields discrete y values y_{kj} with $j = 1, 2, \dots, N_y$. The transformation $u^y(y)$ is also chosen to make the y_{kj} values more closely bunched whenever lines of fixed (y_{kj}, z_k) pass nearby an atomic nucleus. This second step defines y -slices as lines of fixed (y_{kj}, z_k) with functions $\tilde{G}_{kj}(y) \tilde{G}_j(z)$ centered on them. Finally discrete x points x_{kji} with $i = 1, 2, \dots, N_x$ are defined through a transformation $u^x(x)$ such that points (x_{kji}, y_{kj}, z_k) are more densely spaced the closer they are to nuclei. All these transformations taken together define 3D functions $\tilde{G}_{kji}(x) \tilde{G}_{kj}(y) \tilde{G}_j(z)$ centered on points (x_{kji}, y_{kj}, z_k) . To make the basis finite, only functions whose centers lie within a certain distance of at least one of the atoms are kept in the basis. By construction these final basis functions maintain a product form and orthonormality, while being adapted for higher resolution near nuclei. For more technical discussion on how to make a good choice for the u^z , u^y , and u^x coordinate transformations, see the supplemental information section of Ref. [29]. Essentially these coordinate transformation functions are chosen to have a form like that of Eq. (31), but with the s and a parameters varying according to the 3D distance of a particular z or y slice from the nearest atomic nucleus.

Figure 6 shows a 2D cut through a multi-sliced grid, with boxes illustrating the non-zero support of selected gausslet functions. From the figure, one can observe that a downside of multi-slicing is that it results in many long and thin functions far away from any nucleus which provide more resolution than is actually needed. But by using an approximate wavefunction as a guide, such as a wavefunction from a Hartree-Fock calculation, one can take the additional step of combining such redundant functions together into single functions to reduce the basis size.

4.2.4 Diagonal approximations

A final and very important step when using multi-sliced gausslet bases is to make a *diagonal approximation* to the Coulomb repulsion term when discretizing the Hamiltonian. Being able to use a diagonal form for this very expensive term is a crucial way to reduce calculation costs. A diagonal approximation to the Coulomb term means a discretization of the form

$$\frac{1}{2} \sum_{\sigma\sigma'} \int_{\mathbf{r}\mathbf{r}'} u(\mathbf{r}, \mathbf{r}') \hat{\psi}_{\mathbf{r}\sigma}^\dagger \hat{\psi}_{\mathbf{r}'\sigma'}^\dagger \hat{\psi}_{\mathbf{r}'\sigma'} \hat{\psi}_{\mathbf{r}\sigma} \rightarrow \frac{1}{2} \sum_{\mathbf{m} \neq \mathbf{m}'} V_{\mathbf{m}, \mathbf{m}'} \hat{n}_{\mathbf{m}} \hat{n}_{\mathbf{m}'} \quad (33)$$

where $\mathbf{m} = (i, j, k)$ are the grid points on which the gausslets are centered and the $\hat{n}_{\mathbf{m}} = \hat{c}_{\mathbf{m}}^\dagger \hat{c}_{\mathbf{m}}$ operators measure the occupation of the gausslet basis functions $\tilde{G}_{kji}(x)\tilde{G}_{ji}(y)\tilde{G}_k(z)$ at position $\mathbf{r}_{\mathbf{m}} = \mathbf{r}_{ijk}$. Such a form is called diagonal because it only involves N^2 terms, with N being the total number of 3D gausslet functions, rather than N^4 terms as in the most general form of the discrete Coulomb interaction Eq. (19).

The justification for using a diagonal approximation with multi-sliced gausslets is that gausslets have the ability to represent a wide variety of smooth functions while also integrating like a delta function, meaning:

$$\int_x \tilde{G}_i(x) f(x) = w_i f(x_i) \quad (34)$$

for any smooth function f where x_i is the center of the adapted gausslet $\tilde{G}_i(x)$ and $w_i = \int_x \tilde{G}_i(x)$. Note that this delta-function relation differs from Eq. (30) by the inclusion of the weight w_i . This is because, although adapted gausslets remain square-normalized, they generally no longer integrate to 1.0.

To see how the representability and delta function properties of gausslets lead to a diagonal approximation, consider just the single-particle potential term $v(x)$ for a 1D Hamiltonian in the first-quantization formalism. The action of this Hamiltonian term on a single-particle wavefunction $\psi(x)$ is $v(x)\psi(x)$. Define the resulting function to be $\phi(x) = v(x)\psi(x)$. Now if we assume that both $\psi(x)$ and $\phi(x)$ are smooth enough that they can be approximated by gausslets, that means there exist coefficients ψ_i and ϕ_i such that

$$\psi(x) = \sum_i \psi_i \tilde{G}_i(x) \quad (35)$$

$$\phi(x) = \sum_i \phi_i \tilde{G}_i(x), \quad (36)$$

where in fact

$$\psi_i = \int_x \tilde{G}_i(x) \psi(x) = w_i \psi(x_i) \quad (37)$$

$$\phi_i = \int_x \tilde{G}_i(x) \phi(x) = w_i \phi(x_i) \quad (38)$$

because of the delta-function integration property Eq. (30) of gausslets. The discrete form of the Hamiltonian we seek are the coefficients v_{ij} which are defined as mapping

$$\phi_i = \sum_j v_{ij} \psi_j. \quad (39)$$

Now observe that

$$\phi_i = \int_x \tilde{G}_i(x) \phi(x) = \int_x \tilde{G}_i(x) (v(x) \psi(x)) = w_i v(x_i) \psi(x_i) = v(x_i) \psi_i, \quad (40)$$

where we used the delta-function property of the gausslets and Eq. (37) above to obtain the last expression. By inspection of the above equation and from the definition of v_{ij} in Eq. (39), we finally see that $v_{ij} = \delta_{ij} v(x_i)$ which is the diagonal approximation we seek (in a first-quantized form). Note that the most general expression for v_{ij} in a gausslet basis would be $v_{ij} = \int_x \tilde{G}_i(x) v(x) \tilde{G}_j(x) = w_i v(x_i) \tilde{G}_j(x_i)$ which is non-zero for $i \neq j$ and thus non-diagonal. So for the diagonal approximation to be justified, one is making additional smoothness assumptions about the functions being transformed by the Hamiltonian and not just invoking properties of the Hamiltonian itself. By similar arguments one can make a diagonal approximation to the two-body Coulomb interaction of the form $V_{ijkl} = \delta_{il} \delta_{jk} u(x_i, x_j)$ when using gausslet basis functions.

The form of diagonal approximation we just outlined is called the point-wise approximation. Another type of diagonal approximation that can be derived is the *integral approximation*:

$$V_{ijkl} = \frac{\delta_{il} \delta_{jk}}{w_i w_j} \int_{x,x'} \tilde{G}_i(x) u(x, x') \tilde{G}_j(x'), \quad (41)$$

which is much more accurate than the point-wise approximation. There is also a summed diagonal approximation which is discussed in Ref. [28]. A straightforward generalization of one of these diagonal approximations to the case of 3D gausslet bases leads to the expression Eq. (33) at the beginning of this section.

4.2.5 The multi-sliced gausslet approach in practice

To test multi-sliced gausslet bases for quantum chemistry, Ref. [29] considered systems of hydrogen atoms, much like in the Ref. [26] studies of sliced-bases. But whereas Ref. [26] primarily emphasizes scalability to very long systems, Ref. [29] emphasizes the ability of multi-sliced gausslets to reach the complete basis set limit (or continuum limit). Yet multi-sliced gausslets are also scalable to very long systems when used in DMRG.

To study the effect of the scale parameter s controlling the typical spacing between neighboring gausslets, Fig. 7 shows Hartree-Fock calculations of 10-atom hydrogen chains with inter-atomic spacing $R = 1$ in standard Gaussian basis sets (horizontal lines) such as quadruple-zeta (cc-pVQZ) and 5-zeta (cc-pV5Z), as well as continuum extrapolations of Gaussian bases. In contrast, the more jagged set of points shows results of converged multi-sliced gausslet (MSG) calculations as a function of the scale or gausslet spacing (x axis of plot). For a scale below $s = 0.6$ in atomic units, the MSG results converge smoothly and systematically until reaching close agreement with the best Gaussian basis extrapolation to the complete basis set limit.

To test the multi-sliced gausslet approach within high-accuracy DMRG calculations, Fig. 8 shows DMRG calculations of 10-atom hydrogen chains as a function of the inter-atomic spacing R , including strongly correlated stretched or larger- R chains. The figure shows results

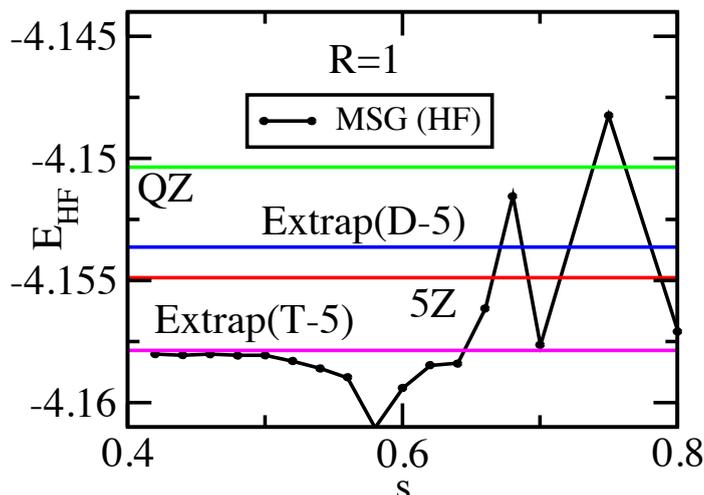


Fig. 7: Hartree-Fock energies (in units of Hartree) of 10-atom hydrogen chains with interatomic spacing $R = 1$. The points are results from multi-sliced gausslet (MSG) bases with varying scale parameter controlling the spacing between gausslets. The lines labeled QZ and 5Z are energies obtained with the *cc-pVQZ* and *cc-pv5Z* Gaussian basis sets, while the lines labeled Extrap are extrapolations to the complete basis set limit using either double-zeta or triple-zeta up through 5-zeta.

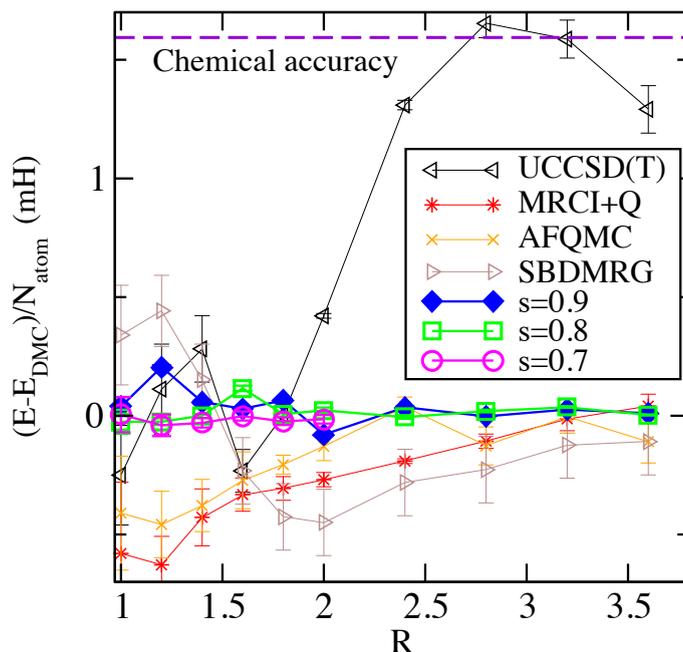


Fig. 8: Energies of 10-atom hydrogen chains computed by various methods relative to those obtained by diffusion Monte Carlo (E_{DMC}). Energy differences are shown in milli-Hartree. Results are from Ref. [27].

from competitive quantum chemistry methods including coupled cluster (UCCSD(T)), multi-reference configuration interaction (MRCI+Q), auxiliary field quantum Monte Carlo (AFQMC), DMRG using a sliced-basis (SBDMRG), and finally diffusion quantum Monte Carlo (DMC), which is used as the reference energy for the figure. These results were first obtained and dis-

cussed in much more detail in Ref. [27]. Multi-sliced gausslet DMRG (MSG-DMRG) results are labeled by the scale factor s they used. The figure shows that for scale factors $s = 0.8$ and especially $s = 0.7$, the MSG-DMRG results have the closest agreement with the DMC results on a scale of less than 0.1 milli-Hartree. Because the sources of error in DMC versus MSG-DMRG have a very different origin, their close agreement strongly suggests they are both resolving the continuum limit to the highest accuracy of the methods shown.

To obtain such accurate results, it should be noted for completeness that additional technical steps are used in Ref. [29] which are beyond the scope of the discussion here. These include a delta-function correction to the single-particle potential and combining the single-particle part of the energy with Hartree-Fock energies on a finer scale to reduce finite-scale errors in the MSG-DMRG calculations.

4.3 DMRG with sliced-basis or multi-sliced discretization

Having introduced two types of local bases which allow quantum chemistry DMRG to scale to larger numbers of atoms and better resolve the continuum limit, let us now discuss some of the technical steps involved in using these bases in actual DMRG calculations.

4.3.1 Splitting terms into multiple MPOs

One simple but very effective optimization of DMRG when treating discrete quantum chemistry Hamiltonians of the form Eq. (19) is to split different types of Hamiltonian terms into separate matrix product operators (MPOs). The reason for doing this is that even after making the most efficient possible MPO representation of the entire Hamiltonian as a single MPO, one can observe that this MPO consists of disjoint blocks of terms which do not mix with one another, such that its bond dimension is the sum of the dimensions of each of the blocks.

To illustrate why splitting the terms associated with each block into separate MPOs is more efficient, consider an example where there are four different blocks, each contributing a size k to the bond dimension. Because DMRG scales as the sum of squares of the bond dimensions of each MPO used, storing these blocked terms in separate MPOs will have a cost $4k^2$ within DMRG whereas combining them into a single MPO will have a much higher cost of $(4k)^2 = 16k^2$ in DMRG.

An example of a possible splitting of terms into separate Hamiltonians which are summed to make the total Hamiltonian could be

$$H = H_{\uparrow} + H_{\downarrow} + H_V \quad (42)$$

$$H_{\uparrow} = \frac{1}{2} \sum_{ij} t_{ij} \hat{c}_{i\uparrow}^{\dagger} \hat{c}_{j\uparrow} \quad (43)$$

$$H_{\downarrow} = \frac{1}{2} \sum_{ij} t_{ij} \hat{c}_{i\downarrow}^{\dagger} \hat{c}_{j\downarrow} \quad (44)$$

$$H_V = \frac{1}{2} \sum_{ijkl\sigma\sigma'} V_{ijkl} \hat{c}_{i\sigma}^{\dagger} \hat{c}_{j\sigma'}^{\dagger} \hat{c}_{k\sigma'} \hat{c}_{l\sigma} \cdot \quad (45)$$

This is only one possible splitting because additional structure may be present in the V_{ijkl} terms resulting from different choices of local bases. This structure can suggest additional further splittings, such as separating terms which connect orbitals entirely in one slice versus across two different slices within the sliced-basis approach.

4.3.2 Compressing long-range interactions with MPOs

A crucial optimization that can be applied when using spatially local bases in DMRG is the compression of the two-particle Coulomb terms in the electronic structure Hamiltonian using MPO techniques. In the discussion of the sliced-basis and multi-sliced gausslet bases above, we have alluded to this compression a few times since it is so important, but delayed discussing it to this section since it is also very technical and requires a detailed familiarity with the construction of MPOs. A very detailed discussion of the compression algorithm and MPO structure is given in the appendix of Ref. [26].

Due to the limited technical scope of this chapter, let us first describe what this MPO compression accomplishes, treating the method itself as just a “black box” algorithm. In the most general case, given a set of Coulomb interaction integrals V_{ijkl} running over N orbitals, meaning $i, j, k, l = 1, 2, \dots, N$, the minimum size MPO which exactly represents the discrete Hamiltonian Eq. (19) has a bond dimension which scales as N^2 [13]. Because the DMRG algorithm scales quadratically in the bond dimension of the Hamiltonian MPO, using an uncompressed MPO for quantum chemistry results in an N^4 scaling which is typical for quantum chemistry but nevertheless very costly. However, for systems extended primarily along one dimension (taken to be the z direction) and represented using either a sliced-basis or multi-sliced gausslet basis which have the important property of consisting of local functions, exploiting both sparsity and an off-diagonal low-rank structure in the Coulomb integrals V_{ijkl} allows one to numerically construct an MPO approximation of the Coulomb interaction terms whose bond dimension grows at most logarithmically with the number of atoms in the system. In practice a compressed bond dimension of a few hundreds in size gives very good accuracy even for very large systems consisting of hundreds or thousands of atoms.

To briefly describe how the MPO compression is accomplished, consider a purely diagonal form of the Coulomb interaction $\frac{1}{2} \sum_{ij} V_{ij} \hat{n}_i \hat{n}_j$ which could result from the diagonal approximation within a gausslet basis, for example. Empirically, one can observe that *all* of the blocks $V_{ij}^{(p)}$ of the symmetric matrix V defined by restricting $i \leq p$ and $j \geq i$ are approximately low rank, implying they can be approximated well by a truncated singular value decomposition (SVD). This block-low-rank property can be understood as resulting from the smoothness of the Coulomb interaction for electrons far apart from one another. Furthermore, the unitary matrices computed in the SVD factorizations of each block $V^{(p)}$ can be related to each other by auxiliary unitary maps of a fixed size, related to the number of singular values kept. By defining MPO tensors which implement these maps, the MPO can reconstruct any of the matrix elements V_{ij} while having a bond dimension determined only by the number of singular values kept in the factorization of V , which depends very weakly on the size of the system [26].

5 Conclusions and future directions

In this chapter, we reviewed the DMRG algorithm for optimizing many-body wave functions in matrix product state form and the application of DMRG to quantum chemistry calculations. A key consideration in making DMRG efficient is choosing a basis that allows the Hamiltonian, typically represented as a matrix product operator (MPO) tensor network within DMRG, to have a manageable size. This issue becomes extremely important when applying DMRG to the electronic structure Hamiltonian used in quantum chemistry.

Although the approach of discretizing the electronic structure Hamiltonian using Gaussian basis sets has been very successful for applying DMRG to chemistry, here we reviewed two recent alternative basis constructions which are much more spatially local than standard 3D Gaussian basis sets. Though locality makes these bases much larger than Gaussian bases, these local bases can nevertheless be very advantageous for DMRG calculations whose costs are tied much more strongly to the spatial locality or smooth spatial decay of Hamiltonian terms than the number of sites or total size of the Hilbert space used in the calculation.

Looking ahead, it would be extremely welcome if a basis such as multi-sliced gausslets could be used for PEPS tensor network calculations, which are tensor networks that are scalable along two dimensions in contrast to MPS which are only scalable in one dimension. One of the key challenges is developing efficient Coulomb interaction representations suitable for PEPS, though significant progress was recently made in Ref. [31]. Another interesting direction would be to use the sliced-basis or multi-sliced gausslet bases in quantum chemistry methods very different from DMRG, such as auxiliary field quantum Monte Carlo (AFQMC). Ideas similar to the compression of the Coulomb terms (Section 4.3.2) would have to be adapted to AFQMC, but then it could benefit from the high continuum resolution of gausslets, for example.

Acknowledgements

Thanks to Steve R. White for many helpful discussions and supporting information regarding both the sliced-basis and multi-sliced Gausslet approach to quantum chemistry with DMRG.

References

- [1] S.R. White, *Phys. Rev. Lett.* **69**, 2863 (1992)
- [2] S. White, *Phys. Rev. B* **48**, 10345 (1993)
- [3] U. Schollwöck, *Rev. Mod. Phys.* **77**, 259 (2005)
- [4] U. Schollwöck, *Annals of Physics* **326**, 96 (2011)
- [5] S. Östlund and S. Rommer, *Phys. Rev. Lett.* **75**, 3537 (1995)
- [6] E. Stoudenmire and S.R. White, *Ann. Rev. Condens. Matter Phys.* **3**, 111 (2012)
- [7] S.R. White and R.L. Martin, *J. Chem. Phys.* **110**, 4127 (1999)
- [8] G.K.-L. Chan and S. Sharma, *Ann. Rev. Phys. Chem.* **62**, 465 (2011)
- [9] D. Perez-Garcia, F. Verstraete, M.M. Wolf, and J.I. Cirac, *Quantum Inf. Comput.* **7**, 401 (2007)
- [10] R. Penrose, *Combinatorial mathematics and its applications* **1**, 221 (1971)
- [11] S.R. White, *Phys. Rev. B* **72**, 180403 (2005)
- [12] C. Hubig, I.P. McCulloch, U. Schollwöck, and F.A. Wolf, *Phys. Rev. B* **91**, 155115 (2015)
- [13] G.K.-L. Chan, A. Keselman, N. Nakatani, Z. Li, and S.R. White, *J. Chem. Phys.* **145**, 014102 (2016)
- [14] O.L. Polyansky, A.G. Császár, S.V. Shirin, N.F. Zobov, P. Barletta, J. Tennyson, D.W. Schwenke, and P.J. Knowles, *Science* **299**, 539 (2003)
- [15] N.M. Tubman, J. Lee, T.Y. Takeshita, M. Head-Gordon, and K.B. Whaley, *J. Chem. Phys.* **145**, 044112 (2016)
- [16] D.M. Ceperley, *Rev. Mod. Phys.* **67**, 279 (1995)
- [17] J. Shee, B. Rudsteyn, E.J. Arthur, S. Zhang, D.R. Reichman, and R.A. Friesner, *J. Chem. Theory Comput.* **15**, 2346 (2019)
- [18] G. Fano, F. Ortolani, and L. Ziosi, *J. Chem. Phys.* **108**, 9246 (1998)
- [19] Y. Kurashige, G.K.-L. Chan, and T. Yanai, *Nat. Chem.* **5**, 660 (2013)
- [20] Y. Kurashige and T. Yanai, *J. Chem. Phys.* **135**, 094104 (2011)
- [21] T. Xiang, *Phys. Rev. B* **53**, R10445 (1996)
- [22] S. Sharma and G.K.-L. Chan, *J. Chem. Phys.* **136**, 124121 (2012)

- [23] N. Nakatani, S. Wouters, D. Van Neck, and G.K.-L. Chan, *J. Chem. Phys.* **140**, 024108 (2014)
- [24] G.K.-L. Chan and M. Head-Gordon, *J. Chem. Phys.* **116**, 4462 (2002)
- [25] R. Olivares-Amaya, W. Hu, N. Nakatani, S. Sharma, J. Yang, and G.K.-L. Chan, *J. Chem. Phys.* **142**, 034102 (2015)
- [26] E.M. Stoudenmire and S.R. White, *Phys. Rev. Lett.* **119**, 046401 (2017)
- [27] M. Motta, D.M. Ceperley, G.K.-L. Chan, J.A. Gomez, E. Gull, S. Guo, C.A. Jiménez-Hoyos, T.N. Lan, J. Li, F. Ma, A.J. Millis, N.V. Prokof'ev, U. Ray, G.E. Scuseria, S. Sorella, E.M. Stoudenmire, Q. Sun, I.S. Tupitsyn, S.R. White, D. Zgid, and S. Zhang, *Phys. Rev. X* **7**, 031059 (2017)
- [28] S.R. White, *J. Chem. Phys.* **147**, 244102 (2017)
- [29] S.R. White and E.M. Stoudenmire, *Phys. Rev. B* **99**, 081110 (2019)
- [30] G. Evenbly and S.R. White, *Phys. Rev. A* **97**, 052314 (2018)
- [31] M.J. O'Rourke, Z. Li, and G.K.-L. Chan, *Phys. Rev. B* **98**, 205127 (2018)