Large-scale ab initio configuration interaction calculations for light nuclei

This content has been downloaded from IOPscience. Please scroll down to see the full text.
2012 J. Phys.: Conf. Ser. 403 012019
(http://iopscience.iop.org/1742-6596/403/1/012019)

View the table of contents for this issue, or go to the journal homepage for more

Download details:
IP Address: 207.241.229.51
This content was downloaded on 11/07/2017 at 20:46

Please note that terms and conditions apply.

You may also be interested in:

No Core CI calculations for light nuclei with chiral 2- and 3-body forces
Pieter Maris, H Metin Aktulga, Sven Binder et al.

Ab Initio Nuclear Structure Calculations of Light Nuclei
Pieter Maris

Ab initio calculations for Be-isotopes with JISP16
P Maris

Recent development of Monte Carlo shell model and its application to no-core calculations
T Abe, P Maris, T Otsuka et al.

The no-core shell model with general radial bases
M A Caprio, P Maris and J P Vary

Monte Carlo shell model studies with massively parallel supercomputers
Noritaka Shimizu, Takashi Abe, Michio Honma et al.

Symmetry-Guided Ab Initio Approach to Light and Medium-Mass Nuclei
T Dytrych, J P Draayer, K D Launey et al.

Coupled-cluster computations of atomic nuclei
G Hagen, T Papenbrock, M Hjorth-Jensen et al.
Large-scale ab initio configuration interaction calculations for light nuclei

Pieter Maris¹, H Metin Aktulga², Mark A Caprio³, Ümit V Çatalyürek¹,⁵, Esmond G Ng², Dossay Oryspayev⁶, Hugh Potter¹, Erik Saule¹, Masha Sosonkina⁷, James P Vary¹, Chao Yang² and Zheng Zhou⁸

¹ Department of Physics and Astronomy, Iowa State University, Ames, IA 50011, USA
² Computational Research Div, Lawrence Berkeley National Lab, Berkeley, CA 94720, USA
³ Department of Physics, University of Notre Dame, Notre Dame, IN 46556, USA
⁴ Dept. of Biomedical Informatics, The Ohio State University, Columbus, OH 43210, USA
⁵ Dept. of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210, USA
⁶ Dept. of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA
⁷ Dept. of Modeling, Simulation and Visualization Eng., Old Dominion University, Norfolk, VA 23529, USA and Ames Laboratory, Ames, IA, 50011, USA
⁸ School of Computer, Wuhan University, P. R. China

Abstract. In ab-initio Configuration Interaction calculations, the nuclear wavefunction is expanded in Slater determinants of single-nucleon wavefunctions, and the many-body Schrödinger equation becomes a large sparse matrix problem. The challenge is to reach numerical convergence to within quantified numerical uncertainties for physical observables using finite truncations of the infinite-dimensional basis space. We discuss strategies for constructing and solving the resulting large sparse matrix eigenvalue problems on current multicore computer architectures. Several of these strategies have been implemented in the code MFDn, a hybrid MPI/OpenMP Fortran code for ab-initio nuclear structure calculations that can scale to 100,000 cores and more. Finally, we will conclude with some recent results for ¹²C including emerging collective phenomena such as rotational band structures using SRG evolved chiral N3LO interactions.

1. No-Core Configuration Interaction calculations

Solving for nuclear properties with the best available nucleon-nucleon (2NF) potentials [1, 2], supplemented by three-nucleon forces (3NF) as needed [3, 4, 5], using a quantum many-particle framework that respects all the known symmetries of the potentials is referred to as an “ab initio” problem and is recognized to be computationally hard.

A commonly used approach in nuclear physics is the Configuration Interaction (CI) method for solving the many-body nuclear Hamiltonian in a (sufficiently large) single-particle basis space. In this approach, the many-body Schrödinger equation becomes a large sparse matrix eigenvalue problem. In a No-Core CI (NCCI) approach [6], the wavefunction $\Psi$ of a nucleus consisting of $A$ nucleons (protons and neutrons) is expanded in an $A$-body basis of Slater determinants $\Phi_k$ of single-particle wavefunctions $\phi_i$

$$\Psi(r_1, \ldots, r_A) = \sum c_k \Phi_k(r_1, \ldots, r_A),$$

Published under licence by IOP Publishing Ltd
with \( \Phi_k(r_1, \ldots, r_A) = A[\phi_{n_1l_1j_1m_1}(r_1) \phi_{n_2l_2j_2m_2}(r_2) \ldots \phi_{n_Al_Aj_Am_A}(r_A)] \). Conventionally, one uses a harmonic oscillator (HO) basis for the single-particle wavefunctions, but it is straightforward to extend this approach to a more general single-particle basis \([7, 8]\). The single-particle wavefunctions are labelled by the quantum numbers \( n, l, j, \) and \( m \), where \( n \) and \( l \) are the radial and orbital HO quantum numbers, with \( N = 2n + l \) the number of HO quanta; \( j \) is the total single-particle spin, and \( m \) its projection along the \( z \)-axis. The many-body basis states have well-defined total spin-projection, which is simply the sum of \( m \) of the single-particle states, \( M = \sum m \), hence the name \( M \)-scheme. However, the many-body basis states do not have a well-defined total spin \( J \). Some of the benefits of this scheme is that it is very simple to implement, and that in two runs (one for positive and one for negative parity), we get the complete low-lying spectrum, including the ground state, even if the spin of the ground state is unknown.

In a complete basis, this method would give exact results for a given input interaction \( V \). However, practical calculations can only be done in a finite-dimensional truncation of a complete basis. Different truncation schemes have different convergence rates: Full Configuration Interaction or FCI employs a widely-know truncation in which all many-body basis states are retained that can be constructed from a finite set of single-particle states. This method is commonly used in atomic and molecular physics, but has a rather slow convergence rate in typical ab initio nuclear structure calculations \([9, 10]\). The so-called \( N_{\text{max}} \) truncation, in which the total number of HO quanta in a basis state is limited: \( \sum N_k \leq N_0 + N_{\text{max}} \), is generally much more efficient for NCCI calculations. Here, \( N_k \) is the number of quanta of each of the single-particle states in the many-body basis state; \( N_0 \) is the minimal number of quanta for that nucleus; and \( N_{\text{max}} \) is the truncation parameter. For HO single-particle states, or more general, for any basis in which the single-particle states have radial and orbital quantum numbers \( n \) and \( l \), we have \( N_k = 2n_k + l_k \). Furthermore, in combination with a HO basis for the single-particle basis, this truncation leads to an exact factorization of the center-of-mass wavefunction and the relative wavefunction.

For a very light nucleus like \(^4\text{He}\) or \(^6\text{Li}\) one can achieve convergence of the ground state energy and other observables for many modern 2NFs by simply going to a sufficiently large basis, but for larger nuclei and/or with 3NFs that is not practical. In order to reliably extrapolate to the complete basis, we need results in finite bases up to at least \( N_{\text{max}} = 8 \), and preferably \( N_{\text{max}} = 10 \) or even higher \([11, 12]\), depending on the interaction. This leads to a computational challenge, because the basis space dimension increases rather rapidly with \( N_{\text{max}} \) and with the number of particles. In the left panel of Fig. 1 we show the basis space dimensions for a number of \( N = Z \) nuclei in the \( p \)-shell (solid curves), as well as for a few nuclei in the \( sd \)-shell. The basis space dimension for even the first couple of nuclei in the \( sd \)-shell (more than 8 protons and 8 neutrons) is of the order of 10 billion or more at \( N_{\text{max}} = 8 \). Furthermore, the sparsity of the matrix depends on the ‘rank’ of the potential as shown in the right panel of Fig. 1: a two-body potential leads to a much sparser many-body matrix than a three-body potential \([13]\). As the matrix size increases, there is a clear need for high-performance computing: obtaining the lowest 10 to 20 eigenvalues and eigenvectors of matrices with a dimension of the order of several billion is a nontrivial task, even if the matrix is extremely sparse.

2. Computational Challenges and Solutions

For our calculations we use the code MFDn, which is a hybrid MPI/OpenMP parallel Fortran 90 code for nuclear structure calculations, that has been in development for over two decades. Significant improvements in its performance have been made over the last five years \([14, 15, 16]\) under the UNEDF SciDAC program. MFDn constructs the many-body basis states in \( M \)-scheme and the corresponding many-body matrix, and solves for the lowest eigenstates using an iterative Lanczos algorithm. Each Lanczos iteration consists of a matrix-vector multiplication followed by orthogonalization. At the end of a run MFDn writes the eigenvectors to file, and evaluates
selected physical observables which can be compared to experimental data (the wavefunctions themselves are not experimentally observable). It also writes out the one-body density matrix elements (in the underlying single-particle basis), which can be used for further analysis.

Each vector is distributed over \( n \) processors, and the matrix itself is distributed in a 2-dimensional fashion over all available processors. We work with the lower triangle only, because the matrix is symmetric, so we need \( n(n+1)/2 \) processors. For a sparse matrix-vector multiplication (SpMV) we need to broadcast the input vector along the columns and along the rows, then perform a local SpMV and transpose SpMV on each processor, and finally reduce the local output vectors along both the rows and the columns to get the global output vector, as
Figure 4. Speedup of the Lanczos iterations for different processors layout for $^{10}$Be at $N_{\text{max}} = 8$ using NN-only interactions (left) and with LBCM MPI+OMP for $^7$Li (red, $D = 6.2 \cdot 10^6$ and NNZ = $10^{11}$) and $^{10}$Be (blue, $D = 1.6 \cdot 10^8$ and NNZ = $5.2 \cdot 10^{12}$) at $N_{\text{max}} = 8$ using 3NF (right), where $D$ is the basis dimension, and NNZ the number of nonzero’s in half the matrix.

illustrated in Fig. 2. The local work load of the SpMV is proportional to the number of nonzero matrix elements on each processor. With a round-robin distribution of the $M$-scheme basis states over the $n$ processors, we achieve almost perfect local load balancing, both in memory usage and the (local) CPU usage.

However, the overall performance also depends on the communication overhead, and it turns out that the mapping of the matrix blocks onto the MPI processors has a large influence on the communication time [16]. The layout that was initially implemented in MFDn was Diagonal-Major (DM), as indicated in the left panel of Fig. 3. However, this mapping leads to poor scaling and increasing communication time as the number of processors increases: the communication volume is not balanced among different communicator groups (e.g. the first column communicator group contains $n$ processors whereas the last column communicator group contains only one processor). Furthermore, on modern large-scale high performance computer systems, job-schedulers often use as default a space-filling Z curve (or a similar) algorithm for ranking processing units in order to ensure that physically nearby processing units are ranked consecutively. With the DM mapping of the left panel of Fig. 3, this means that the ‘diagonal’ processors are physically close to each other, creating local communication hot-spots, whereas members of the same column communicator group, as well as members of the same row communicator group, tend to be spread out over the system.

On the other hand, with a Load-Balanced Column-Major (LBCM) mapping as indicated in the right panel of Fig. 3, the communication volume is balanced, and members of the same column communicator group are physically close together, leading to much lower communication times. The overall performance of this mapping is indeed significantly improved over the original DM mapping [16], see the left panel of Fig. 4. We may be able to further optimize these mappings for specific hardware and interconnection networks.

For large runs on multi-core architectures such as the Cray XT4/5, it is significantly more efficient to run the code in a hybrid mode, using OpenMP threading within a node, and MPI between nodes [15]. As the number of threads per node increases on modern multiprocessor systems, the main memory system on a node is often split into multiple memory banks with each bank associated with a set of cores. This is referred to as Nonuniform Memory Access (NUMA), and accesses to remote memory banks (but still within the same node) reduce scalability when
using one MPI processor node, e.g. using one MPI processor per node with 24 threads on Hopper (Cray XE6) at NERSC, or one MPI processor per node with 16 threads on Jaguar (Cray XK6) at ORNL; performance is significantly better using one MPI processor per NUMA node instead, which is what we used in Fig. 4. We are currently working on improving the performance on NUMA architectures by focussing on the data locality [17].

The Lanczos vectors are stored in memory, distributed over all processors; this allows for fast and efficient re-orthogonalization after every SpMV. The mapping of the Lanczos vectors on the processors also influences the communication overhead significantly. The hierarchical 1-dimensional mapping of the right panel of Fig. 3 seems to be most efficient [16]. Note that the re-orthogonalization is done in double precision, even though the matrix and vectors themselves are stored in single precision. This turns out to be essential in order to maintain numerical accuracy over hundreds of Lanczos iterations with matrix dimensions in the billions.

With these recent improvements, we obtain very good scaling characteristics. With 3NF we can scale on Jaguar at ORNL up to the full machine (300k cores), as illustrated in the right panel of Fig. 4. The minimal number of processors needed for a run is determined by the amount of memory needed to store all nonzero matrix elements in compressed-column format (CCF), or alternatively, in compressed-row format (CRF). On currently available machines, the largest many-body basis dimension that MFDn can handle is about 10 to 15 billion with 2NFs only, and about one billion with 2NFs plus 2NFs.

Despite the significant performance improvement, for the current algorithm and implementation the main challenge remains the amount of memory needed to store the matrix in core. An obvious alternative would be to use an 'out-of-core' implementation for the SpMV, reading the matrix from disk as needed. Indeed, such a simple out-of-core implementation for MFDn exists, and has been quite useful on clusters with a local disk associated with each processor, e.g. for the calculations presented in Ref. [12]. However, on modern large-scale high performance computer systems with a parallel filesystem, this simple out-of-core implementation performs rather poorly.

On the other hand, the emergence of high performance computing platforms equipped with solid state drives (SSD) presents an opportunity to dramatically increase the efficiency of out-of-core calculations. In Ref. [18], we explore the advantages and challenges associated with performing the SpMV on a small SSD testbed. We use a novel distributed out-of-core linear algebra framework, called Distributed Out-of-Core (DOoC) plus Linear Algebra Frontend (LAF). This DOoC+LAF framework provides an easy-to-use high-level application interface for linear algebra operations such as SpMV, while performing efficient execution by orchestrating pipelined execution of computation, communication and I/O. Preliminary experiments indicate that the out-of-core implementation on the SSD testbed can compete with an in-core implementation in terms of total CPU hours and energy efficiency.

Another recent hardware development is the advent of GPUs and accelerators, which are now employed in several of the top-ten of the fastest supercomputers. However, it is far from trivial to achieve the speedup potentially provided by GPUs. In particular for a heavily memory-bound application such as MFDn, the bottleneck is not floating point operations, but aggregate memory and memory access. Nevertheless, there may be some parts of the MFDn that could benefit from GPUs. We are currently exploring the use of GPUs for the decoupling of the 3NF matrix elements in coupled JT-format. For a simple test we find a speedup of a factor of 20 to 50 on a GPU using CUDA, as compared to a serial code on a CPU. These speedup values include the transfer times between the host CPU and the GPU. Integrating this in a production code for nuclear structure, however, is still a challenge.

In addition, there may be opportunities to use GPUs in combination with a 'matrix-free' SpMV algorithm, in which the necessary matrix elements are constructed as needed, rather than all stored in memory. Obviously, such an implementation would have a strongly reduced
3. Recent results for $^{12}$C with chiral 2-body interactions

In Fig. 5, we show new results for $^{12}$C using different 2NFs: JISP16 and chiral N3LO interactions, SRG renormalized [12] to three different values of the evolution parameter $\lambda$. We did not include the Coulomb interaction in the chiral N3LO calculations, though it is included in the JISP16 calculations. Fig. 5 clearly shows the difference in convergence; furthermore, the actual (converged or extrapolated) binding energy depends strongly on the SRG parameter $\lambda$. This effect is also seen in lighter nuclei and arises because we have neglected induced many-body forces in the SRG evolution. Inclusion of induced 3NFs reduces this $\lambda$ dependence to less than a percent over this $\lambda$ range [20].

For the extrapolation we use the empirical behavior [11, 12] that the nuclear binding energies seem to converge exponentially with $N_{\text{max}}$

$$E_{\text{binding}}^N = E_{\text{binding}}^\infty + a_1 \exp(-a_2 N_{\text{max}}).$$ (2)

Here we use three consecutive $N_{\text{max}}$ values at a fixed basis space parameter $h\omega$ in order to estimate a converged binding energy at that $h\omega$. The error estimate is based on the difference with the extrapolated value obtained from the next-smaller basis spaces at the same $h\omega$ value. An extrapolated result that is within the error estimates of the previous $N_{\text{max}}$ extrapolation, together with a consistent decrease and reduced dependence on $h\omega$ as $N_{\text{max}}$ increases, indicates the optimal $h\omega$ range for this extrapolation method. For $\lambda = 1.5$ fm$^{-1}$ the ground state is extremely well converged, and there is no need to extrapolate to the infinite basis limit, and we find a binding energy of 120.03(5) MeV. For $\lambda = 2.0$ fm$^{-1}$ the extrapolation seems to be reasonably independent of $h\omega$ and with decreasing error bars over a region of 20 MeV $\leq h\omega \leq 30$ MeV, and an extrapolated result of 113.9(6) MeV. At $\lambda = 2.5$ fm$^{-1}$ the convergence is much slower. Nevertheless, the extrapolation gives consistent results, with only a very small dependence on $h\omega$, in the region region 24 MeV $\leq h\omega \leq 36$ MeV, which suggests an extrapolated
result of $106.5\pm5.0$ MeV. Finally, for JISP16 the extrapolation works very well, with extrapolated results almost independent of $\hbar\omega$, and a binding energy of $94.9(4)$ MeV, that is, JISP16 overbinds $^{12}$C by about 2.5 MeV.

In addition to the ground state energy, we also looked at the lowest $2^+$ and $4^+$ states. With the chiral N3LO interactions, these are the two lowest excited states, and their excitation energies are extremely well converged. These three states have energies that are indicative of a rotational band: the ratio $E^{(4)}_{\text{exc}}/E^{(2)}_{\text{exc}}$ is 3.6, independent of $\lambda$, compared to 3.33 for a pure rotational band [21]. Note the different scale for the JISP16 results in Fig. 6; furthermore, with JISP16 there is a $0^+$ and a $1^+$ state with slightly smaller excitation energies than the $4^+$ state which are not shown in this figure.

Also the quadrupole moments and B(E2) transitions of the $2^+$ and $4^+$ states are in reasonable agreement with a rotational model. At $\lambda = 1.5$ fm$^{-1}$ both the quadrupole moments and the B(E2) transitions show convergence, and we find $Q^{(4)}/Q^{(2)} = 1.30$, compared to 1.27 for a rotor. Our calculations suggest an intrinsic quadrupole of about $Q_0 = -16.1$ e fm$^2$. This in turn would lead us to expect B(E2) values of 5.16 e$^2$ fm$^4$ and 7.4 e$^2$ fm$^4$ for the $2^+ \to 0^+$ and $4^+ \to 2^+$ transitions respectively, if the rotational relations for E2 matrix elements hold. Our NCCI calculations indicate a B(E2) for the $2^+ \to 0^+$ transition that is about 10% below this value, and that for the $4^+ \to 2^+$ transition is about 25% below that of the rotational model. However, our results for that last transition are not quite converged yet. It would be very interesting to look in more detail at these states with the Symmetry-Adapted No-Core Shell Model [22].

Of course, these numbers are not (yet) to be compared to experimental numbers, since we incorporated neither induced nor explicit 3NFs in these calculations. However, such calculations
are currently underway on Jaguar at Oak Ridge National Laboratory, and would not have been possible without the recent improvements in the performance of our codes [16].

4. Taming the scale explosion

Despite the tremendous increase in available computing resources and the significant improvements in the performance of our codes, the limiting factor for our calculations remains the combinatorial scale explosion of the dimension, see Fig. 1. As we have just seen for the calculations of $^{12}\text{C}$, $N_{\text{max}} = 10$ results are highly desirable – however, the corresponding basis dimension is 8 billion. Hence there is an ongoing effort to reduce the basis dimension while keeping the relevant physics.

One way to reduce the basis dimension is by exploiting additional symmetries. Instead of using an $M$-scheme, one could construct many-body basis states with well-defined total spin $J$, that is, construct a basis out of eigenstates of the $J^2$ operator. This can be done by starting out with an $M$-scheme basis, and diagonalizing the $J^2$ operator in this $M$-scheme basis numerically. It is easy to see that the $J^2$ operator becomes a block-diagonal matrix in an $M$-scheme basis, provided that basis states with the same quantum numbers $n_i$, $l_i$, $j_i$ (but different $m_i$) are grouped together. Thus diagonalizing the $J^2$ operator reduces to diagonalizing a block-diagonal matrix, that is, repeatedly diagonalizing much smaller (sparse) matrices. The caveat is that some of the blocks that need to be diagonalized become rather large. Nevertheless, this method has been implemented and has been demonstrated to work [23, 24]. In order to achieve good parallel performance and scaling, the largest blocks are diagonalized first, using all processors; next, blocks of medium size are diagonalized, using a subset of processors; and finally the smallest blocks are diagonalized on single processors [23].

Once the $J^2$ operator is diagonalized, we can project the many-body Hamiltonian onto the invariant subspace for a given value of $J$, and obtain the spectrum for that $J$-value, see Fig. 8. Note, however, that the projected many-body matrix is not as sparse as the original matrix in $M$-scheme. The typical sparsity of the many-body matrix in coupled-$J$ scheme with 2NF only is therefore similar to that of a many-body matrix in $M$-scheme with 3NF, see Fig. 9. In fact, although the dimension of this subspace is much smaller than that of the corresponding $M$-scheme basis, the number of nonzero matrix elements in the projected many-body matrix can be (significantly) larger than that of the many-body matrix in $M$-scheme. Furthermore, the construction of the many-body matrix, as well as the construction of the basis itself, is much more involved than in $M$-scheme. Nevertheless, for certain applications, in particular when a lot of states with the same spin (and parity) are required, a basis with well-defined total spin $J$...
is advantageous over an $M$-scheme basis.

Instead of, or in addition to, exploiting symmetries, one could also reduce the basis dimension by keeping only the most important basis states. The question then becomes: How to select the most important basis states? At least three different methods are currently being pursued, namely the Importance-Truncated No-Core Shell Model (IT-NCSM) [25], the no-core Monte-Carlo Shell Model (MCSM) [9], and the Symmetry-Adapted No-Core Shell Model (SA-NCSM) [22, 27].

The IT-NCSM works with an $M$-scheme basis, and typically works with basis dimensions that are an order of magnitude (or more) smaller than the corresponding complete $M$-scheme basis. The sparsity of the resulting matrix is similar to that of the complete $M$-scheme basis and thus the number of nonzero matrix elements is typically several orders of magnitude smaller than of the complete $M$-scheme basis. One can make an estimate of the numerical error introduced by truncating the complete $M$-scheme basis [25].

The no-core MCSM [9] is an extension of the original MCSM [26], and is based on a truncation of the single-particle states, rather than on the many-body states. The many-body basis is constructed from linear combinations of non-orthogonal angular momentum and parity projected deformed Slater determinants with good total angular momentum projection $M$ using a stochastic Monte-Carlo algorithm. Although the actual many-body basis in the no-core MCSM consists often of only a (few) hundred Monte-Carlo basis states, in contrast to the millions or billions of basis states in the corresponding FCI basis, the results from the no-core MCSM are in very good agreement with the corresponding FCI results [9].

The SA-NCSM is based on the observation that nuclear wavefunctions are often dominated by many-body configurations of specific shape, spatial, and spin deformations [22]. By restricting the many-body basis to the dominant symplectic or SU(3) structures up to an $N_{\text{max}}$ cutoff, one obtains almost the same results as with the complete $M$-scheme basis, but in a much smaller basis [27]. Note, however, that the number of nonzero matrix elements could be larger than in the corresponding complete $M$-scheme calculation. In other words, the resulting many-body matrices are less sparse than the corresponding matrices in coupled-$J$ scheme, which in turn are less sparse than the corresponding matrices in $M$-scheme.

Last but not least, is the use of a more general single-particle wavefunctions than the HO basis. Although there are certain advantages to using a HO basis, one is not restricted to this basis. A drawback of the HO basis is that the falloff (Gaussian asymptotics) of the HO functions at large radius makes them rather poorly suited for the description of the asymptotic properties of the nuclear wavefunction. In Ref. [7] we have established the foundations for carrying out NCCI

---

**Figure 9.** Comparison of the sparsity of the many-body matrix in $M$-scheme (solid symbols) and in coupled-$J$ scheme (open symbols).
calculations with an alternative many-body basis built from Coulomb–Sturmian functions. They provide a complete, discrete set of functions with a realistic exponential falloff. Initial results for $^6$Li in this basis are encouraging. In particular, the convergence properties of observables such as the root-mean-square radius are promising.

Note that neither the IT-NCSM nor the no-core MCSM lead to an exact factorization of the center-of-mass motion and the relative motion. Also with the Coulomb–Sturmian basis there is no exact factorization. However, in practice, the spurious center-of-mass dynamics seems to be tractable [7, 28]. On the other hand, the SA-NCSM truncation, in combination with a HO single-particle basis, does lead to an exact factorization of the center-of-mass wavefunction and the relative wavefunction.

Acknowledgments
This work was supported in part by U.S. Department of Energy Grants DE-FC02-09ER41582 (SciDAC/UNEDF), DE-FG02-87ER40371, DE-FG02-95ER-40934, DE-FC02-06ER2775, and by the U.S. NSF grants 0904782, CNS-0643969, OCI-0904809, OCI-0904802, and by the Research Corporation for Science Advancement under a Cottrell Scholar Award. Computational resources were provided by the National Energy Research Supercomputer Center (NERSC), which is supported by the DOE Office of Science, and by an INCITE award, “Nuclear Structure and Nuclear Reactions”, from the DOE Office of Advanced Scientific Computing. This research used resources of the Oak Ridge Leadership Computing Facility at ORNL, which is supported by the DOE Office of Science under Contract DE-AC05-00OR22725.

References
[1] Wiringa R B, Stoks V G J and Schiavilla R 1995 Phys. Rev. C51 38
[2] Entem D R and Machleidt R 2003 Phys. Rev. C68 041001
[3] Pieper S C, Pandharipande V R, Wiringa R B and Carlson J 2001 Phys. Rev. C64 014001
[4] Epelbaum E, Hammer H-W and Meissner U G 2009 Rev. Mod. Phys. 81 1773
[5] Navratil P, Gueorguiev V G, Vary J P, Ormand W E and Nogga A 2007 Phys. Rev. Lett. 99 042501
[6] Navratil P, Vary J P and Barrett B R 2000 Phys. Rev. Lett. 84 5728
[7] Caprio M, Maris P and Vary J P 2012 Phys. Rev. C86 034312
[8] Caprio M, Maris P and Vary J P 2012 these proceedings (Preprint 1208.6055)
[9] Abe T, Maris P, Otsuka T, Shimizu N, Utsuno Y and Vary J 2012 Phys. Rev. C in press
[10] Maris P 2012 accepted for publication in CCP2011
[11] Maris P, Vary J P and Shirokov A M 2009 Phys. Rev. C79 014308
[12] Bogner S K, Furnstahl R J, Maris P, Perry R J, Schwenk A, and Vary J P 2008 Nucl. Phys. A801 21
[13] Vary, J P, Maris, P, Ng, E G, Yang, C and Sosonkina, M 2009 J. Phys.: Conf. Ser. 180 012083
[14] Sternberg P, Ng E G, Yang C, Maris P, Vary J P, Sosonkina M and Le H V 2008 in Proc. of the 2008 ACM/IEEE conf. on Supercomputing SC ’08 (Piscataway, NJ: IEEE Press) 15:1
[15] Maris P, Sosonkina M, Vary J P, Ng E G and Yang C 2010 Procedia Computer Science 1 97
[16] Aktulga H M, Yang C, Ng E G, Maris P and Vary J P 2012 in Lecture Notes in Computer Science 7484 (Springer) 830
[17] Srinivasa S, Sosonkina M, Maris P, and Vary J P 2012 Procedia Computer Science 9 256
[18] Zhou Z, Saule E, Aktulga H M, Yang C, Ng E G, Maris P, Vary J P and Catalyurek U V 2012 in Proc. of IEEE Cluster to be published
[19] Maris P Vary J P, Navratil P, Ormand W E, Nam H, and Dean D J 2011 Phys. Rev. Lett. 106 202502
[20] Roth R, Langhammer J, Calci A, Binder S and Navratil P 2012 Phys. Rev. Lett. 107 072501
[21] Rowe D J 1970 Nuclear Collective Motion (London: Methuen, London)
[22] Dytrych T, Tsviratcheva K D, Bahri C, Draayer J P and Vary J P 2008 J. Phys. G35 095101
[23] Aktulga H M, Yang C, Ng E G, Maris P and Vary J P 2011 in proceedings of HPCS2011 176
[24] Aktulga H M, Yang C, Catalyurek U V, Maris P, Vary J P and Ng E G 2011 in Lecture Notes in Computer Science 7155 Euro-Par workshops (1) (Springer) 305
[25] Roth R 2009 Phys. Rev. C79, 064324
[26] Otsuka T, Honma M, Mizusaki T, Shimizu N and Utsuno Y 2001 Prog. Part. Nucl. Phys. 47 319
[27] Dytrych T, Draayer J P, Launey K D, Caprio M A, Langr D 2012 these proceedings
[28] Hagen, G, Papenbrock, T and Dean, D J 2009 Phys. Rev. Lett. 103 062503