



NuHami1 : A numerical code to generate nuclear two- and three-body matrix elements from chiral effective field theory

Takayuki Miyagi^{1,2,3,a}

¹ Department of Physics, Technische Universität Darmstadt, 64289 Darmstadt, Germany

² ExtreMe Matter Institute EMMI, GSI Helmholtzzentrum für Schwerionenforschung GmbH, 64291 Darmstadt, Germany

³ Max-Planck-Institut für Kernphysik, Saupfercheckweg 1, 69117 Heidelberg, Germany

Received: 17 February 2023 / Accepted: 23 May 2023

© The Author(s) 2023

Communicated by Vittorio Somá

Abstract The applicability of nuclear ab initio calculations has rapidly extended over the past decades. However, starting research projects is still challenging due to the required numerical expertise in the generation of underlying nuclear interaction matrix elements and many-body calculations. To ease the first issue, in this paper we introduce the numerical code NuHami1 to generate the nucleon-nucleon (NN) and three-nucleon (3N) matrix elements expressed in a spherical harmonic-oscillator basis, inputs of many-body calculations. The ground-state energies for the selected doubly closed shell nuclei are calculated with the no-core shell-model (NCSM) and in-medium similarity renormalization group (IMSRG). The code is written in modern Fortran, and OpenMP+MPI hybrid parallelization is available for the 3N matrix-element calculations.

Program summary

Program title: NuHami1

Licensing provisions: GPLv3

Programming language: Modern Fortran

Repository and DOI: <https://github.com/Takayuki-Miyagi/NuHami1-public> DOI:<https://doi.org/10.5281/zenodo.7529481>

Description of problem: Nucleon-nucleon (NN) and three-nucleon (3N) matrix elements are essential inputs in nuclear ab initio calculations. However, developing a numerical code to generate the matrix elements is a demanding task. Prepar-

ing the input matrix elements is one of the main barriers to begin studies.

Method of solution: The NuHami1 code has the capability to generate both NN and 3N matrix elements expressed in a single-particle harmonic-oscillator (HO) basis, which can be used as inputs for most of the ab initio calculation methods. The jobs can be managed by a simple Python script.

Additional comments: For other open-source software, one can use the computational environment for nuclear structure (CENS) [1] and recently published NuclearToolkit code [2].

1 Introduction

The dynamics of the atomic nucleus are governed by the strong interaction, whose fundamental theory is described by quantum chromodynamics (QCD). Since the quarks are tightly confined in a nucleon, it is well established that nuclear Hamiltonians associated with the interactions between nucleons are a good starting point to understand nuclear structure and reactions. The study of nuclear interaction models has a long story starting from the pion-exchange theory [3]. The quantitative understanding of nuclear interactions is an open problem. In the past decade, interactions based on chiral effective-field theory (EFT) [4,5] have become the standard starting point of ab initio many-body calculations. Chiral EFT-based interactions have several advantages over other interactions such as AV18 [6] and CD-Bonn potentials [7]. For example, a systematic expansion is possible by ordering the diagrams according to the power counting, suggesting the possibility of an uncertainty quantification due to the truncation in the expansion [8–11]. Further, many-nucleon interactions naturally appear at higher order, explaining the hierarchy of many-body terms. In

^a e-mail: miyagi@theorie.ikp.physik.tu-darmstadt.de (corresponding author)

nuclear physics, it is well known that 3N interactions play an important role (see for example Ref. [12] as a recent review). With the progress in nuclear interactions and methodological developments in many-body problems, nuclear ab initio studies are well motivated. Nowadays, the applicability extends over the nuclear chart [13] and recently reached to the heaviest known doubly-magic system ²⁰⁸Pb [14], and further applications are expected.

To perform ab initio calculations, the matrix elements of nuclear Hamiltonians (and relevant operators) are essential. However, developing a numerically efficient code for the matrix-element generation requires expert knowledge and can be a barrier for those entering the field. The goal of the NuHamil code is to provide a simple way to generate NN and 3N matrix elements expressed in a spherical HO basis, applicable for basis-expansion methods, such as the no-core shell model (NCSM) [15], coupled-cluster method [16], self-consistent Green’s function method [17], in-medium similarity renormalization group [18] approach, and many-body perturbation theory [19]. The code currently supports the input formats of the open-source BIGSTICK [20] and imsr++ [21] codes for the NCSM and IMSRG calculations, respectively.

This paper is organized as follows. In Sect. 2, we clarify the input NN and 3N matrix elements for the many-body calculations and briefly show how to compute them with the single-particle product state in the HO basis. Also, we discuss the free-space similarity renormalization group (SRG) prescription to soften the nuclear interactions in Sect. 3. We show some benchmark many-body calculation results with NCSM and IMSRG calculations in Sect. 4. The usage of the code and the conclusion are given in Sect. 5 and Sect. 6, respectively.

2 Matrix elements of Hamiltonian

Here, we review how the NN and 3N matrix elements enter in the many-body problem. Our numerical goal is to solve the non-relativistic many-body Schrödinger equation $H|\Psi_n\rangle = E_n|\Psi_n\rangle$, with the intrinsic Hamiltonian with up to 3N terms

$$H = \frac{A-1}{A} \sum_i \frac{\mathbf{p}_i^2}{2m} + \sum_{i<j} (V_{ij}^{NN} - T_{ij}^{NN}) + \sum_{i<j<k} V_{ijk}^{3N}. \tag{1}$$

Here, \mathbf{p}_i is the momentum vector of i th nucleon, and m is the nucleon mass. The factor $(A-1)/A$ in the first term and T_{ij}^{NN} is from the subtraction of the center-of-mass (cm) kinetic term. A is the nucleon number of the system. The terms V_{ij}^{NN} and V_{ijk}^{3N} are the NN and 3N interactions, respectively.

2.1 Second-quantized representation

To proceed with many-body calculations with basis-expansion methods, we begin with the expression by the second quantization. To this end, we first define creation and annihilation operators for a nucleon in an HO orbit \tilde{p} : $c_{\tilde{p}}^\dagger$ and $c_{\tilde{p}}$. The subscript \tilde{p} is a collective index specifying the HO orbit and defined as $\tilde{p} = \{n_p, l_p, j_p, m_p, t_{z,p}\}$. Here, n_p, l_p, j_p, m_p , and $t_{z,p}$ are the nodal quantum number, orbital angular momentum, total angular momentum, z -component of j_p , and the z -component of the isospin distinguishing protons and neutrons, respectively. Note that proton (neutron) states are labeled as $t_z = -1/2$ ($1/2$). In nuclear physics, the use of an HO basis is particularly useful since the coordinate transformation coefficient is well known [22–24], as will be shown in later. The creation and annihilation operators satisfy the anticommutation relations

$$\{c_{\tilde{p}}, c_{\tilde{q}}\} = 0, \{c_{\tilde{p}}^\dagger, c_{\tilde{q}}^\dagger\} = 0, \{c_{\tilde{p}}, c_{\tilde{q}}^\dagger\} = \delta_{\tilde{p}\tilde{q}}. \tag{2}$$

The object $\delta_{\tilde{p}\tilde{q}}$ is defined by products of Kronecker’s delta and is written as

$$\delta_{\tilde{p}\tilde{q}} = \delta_{n_p n_q} \delta_{l_p l_q} \delta_{j_p j_q} \delta_{m_p m_q} \delta_{t_{z,p} t_{z,q}}. \tag{3}$$

Applying the creation operators to the nucleon vacuum state $|0\rangle$, one can define antisymmetrized states. For example, one-, two-, and three-nucleon states can be written as

$$|\tilde{p}\rangle = c_{\tilde{p}}^\dagger |0\rangle, \tag{4}$$

$$|\tilde{p}\tilde{q}\rangle = c_{\tilde{p}}^\dagger c_{\tilde{q}}^\dagger |0\rangle, \tag{5}$$

$$|\tilde{p}\tilde{q}\tilde{r}\rangle = c_{\tilde{p}}^\dagger c_{\tilde{q}}^\dagger c_{\tilde{r}}^\dagger |0\rangle. \tag{6}$$

Using the creation and annihilation operators, an arbitrary n -body operator $O^{[n]}$ can be expressed as

$$O^{[n]} = \left(\frac{1}{n!}\right)^2 \sum_{\tilde{p}'_1 \dots \tilde{p}'_n} \sum_{\tilde{p}_1 \dots \tilde{p}_n} O_{\tilde{p}'_1 \dots \tilde{p}'_n \tilde{p}_1 \dots \tilde{p}_n} c_{\tilde{p}'_1}^\dagger \dots c_{\tilde{p}'_n}^\dagger c_{\tilde{p}_n} \dots c_{\tilde{p}_1}. \tag{7}$$

The object $O_{\tilde{p}'_1 \dots \tilde{p}'_n \tilde{p}_1 \dots \tilde{p}_n}$ is a shorthand notation for the operator matrix element $\langle \tilde{p}'_1 \dots \tilde{p}'_n | O^{[n]} | \tilde{p}_1 \dots \tilde{p}_n \rangle$. In the same way, the Hamiltonian in Eq. (1) can be quantized as

$$H = \sum_{\tilde{p}'\tilde{p}} T_{\tilde{p}'\tilde{p}} c_{\tilde{p}'}^\dagger c_{\tilde{p}} + \left(\frac{1}{2!}\right)^2 \sum_{\tilde{p}'\tilde{q}'\tilde{p}\tilde{q}} (V_{\tilde{p}'\tilde{q}'\tilde{p}\tilde{q}}^{NN} - T_{\tilde{p}'\tilde{q}'\tilde{p}\tilde{q}}^{NN}) c_{\tilde{p}'}^\dagger c_{\tilde{q}'}^\dagger c_{\tilde{q}} c_{\tilde{p}} + \left(\frac{1}{3!}\right)^2 \sum_{\tilde{p}'\tilde{q}'\tilde{r}'\tilde{p}\tilde{q}\tilde{r}} V_{\tilde{p}'\tilde{q}'\tilde{r}'\tilde{p}\tilde{q}\tilde{r}}^{3N} c_{\tilde{p}'}^\dagger c_{\tilde{q}'}^\dagger c_{\tilde{r}'}^\dagger c_{\tilde{r}} c_{\tilde{q}} c_{\tilde{p}}, \tag{8}$$

using the matrix elements of one-body kinetic term $T_{\tilde{p}'\tilde{p}}$, two-body kinetic term $T_{\tilde{p}'\tilde{q}'\tilde{p}\tilde{q}}^{\text{NN}}$, NN interaction $V_{\tilde{p}'\tilde{q}'\tilde{p}\tilde{q}}^{\text{NN}}$, and 3N interaction $V_{\tilde{p}'\tilde{q}'\tilde{r}'\tilde{p}\tilde{q}\tilde{r}}^{\text{3N}}$.

2.2 J -coupled scheme

The number of matrix elements defined in Eq. (8) is greatly reduced by exploiting the rotational symmetry of the Hamiltonian. To introduce a smaller set of matrix elements, we define the J -coupled two- and three-body states using the Clebsch–Gordan coefficient $C_{m_1 m_2 m_1+m_2}^{j_1 j_2 J}$:

$$|pq : JM\rangle = \sqrt{\frac{1}{1 + \delta_{pq}}} \sum_{m_p m_q} C_{m_p m_q M}^{j_p j_q J} |\tilde{p}\tilde{q}\rangle, \tag{9}$$

$$|pqr : J_{pq} JM\rangle = \sum_{m_p m_q m_r} C_{m_p m_q m_r M}^{j_p j_q j_r J} |\tilde{p}\tilde{q}\tilde{r}\rangle. \tag{10}$$

Here, $p, q,$ and r are the quantum number set without m , i.e. $p = \{n_p, l_p, j_p, t_{z,p}\}$, and an additional Kronecker’s delta product is introduced as

$$\delta_{pq} = \delta_{n_p n_q} \delta_{l_p l_q} \delta_{j_p j_q} \delta_{t_{z,p} t_{z,q}}. \tag{11}$$

Note that the factor $\sqrt{1/(1 + \delta_{pq})}$ in the two-body state is for the normalization so that we have $\langle pq : J'M' | pq : JM \rangle = \delta_{J'J} \delta_{M'M}$. On the other hand, such normalization factor is not usually included in the three-body state. For the three-body state, one can define another state by a different angular momentum coupling order, which should be related with the Wigner’s $6j$ -symbol. In this paper, the first and second indices are always coupled first, and then the third index is coupled. In practical applications, we need permutations of the indices of the states to further reduce the storage requirement. For the two-body state, it is given by

$$|qp : JM\rangle = -(-1)^{j_p+j_q-J} |pq : JM\rangle. \tag{12}$$

Likewise, permutations of the indices in the three-body state are given by

$$|qrp : J_{qr} JM\rangle = - \sum_{J_{pq}} (-1)^{j_q+j_r+J_{qr}} \sqrt{[J_{pq}][J_{qr}]} \times \left\{ \begin{matrix} j_p & j_q & J_{pq} \\ j_r & J & J_{qr} \end{matrix} \right\} |pqr : J_{pq} JM\rangle, \tag{13}$$

$$|rpq : J_{pr} JM\rangle = - \sum_{J_{pq}} (-1)^{j_p+j_q+J_{pq}} \sqrt{[J_{pq}][J_{pr}]} \times \left\{ \begin{matrix} j_p & j_q & J_{pq} \\ J & j_r & J_{pr} \end{matrix} \right\} |pqr : J_{pq} JM\rangle, \tag{14}$$

$$|qpr : J_{pq} JM\rangle = -(-1)^{j_p+j_q-J_{pq}} |pqr : J_{pq} JM\rangle, \tag{15}$$

$$|rqp : J_{qr} JM\rangle = \sum_{J_{pq}} \sqrt{[J_{pq}][J_{qr}]}$$

$$\times \left\{ \begin{matrix} j_p & j_q & J_{pq} \\ j_r & J & J_{qr} \end{matrix} \right\} |pqr : J_{pq} JM\rangle, \tag{16}$$

$$|prq : J_{pr} JM\rangle = - \sum_{J_{pq}} (-1)^{j_q+j_r+J_{pr}+J_{pq}} \sqrt{[J_{pq}][J_{pr}]} \times \left\{ \begin{matrix} j_p & j_q & J_{pq} \\ J & j_r & J_{pr} \end{matrix} \right\} |pqr : J_{pq} JM\rangle. \tag{17}$$

Here, Wigner’s $6j$ -symbol with the standard notation [25] and $[x] = 2x + 1$ are introduced. With the J -coupled states, the two- and three-body matrix elements can be introduced as $\langle p'q' : J'M' | O^{[2]} | pq : JM \rangle$ and $\langle p'q'r' : J'_{p'q'} J'M' | O^{[3]} | pqr : J_{pq} JM \rangle$, respectively. Because of the rotational invariance of the Hamiltonian, the Hamiltonian matrix is M -independent and diagonal with respect to J and M . Therefore, we introduce a shorthand notation for the matrix elements.

$$T_{p'p} = \langle p | T | q \rangle, \tag{18}$$

$$T_{p'q'pq}^{\text{NN},J} = \langle p'q' : JM | T^{\text{NN}} | pq : JM \rangle, \tag{19}$$

$$V_{p'q'pq}^{\text{NN},J} = \langle p'q' : J'M' | V^{\text{NN}} | pq : JM \rangle, \tag{20}$$

$$V_{p'q'r'pqr}^{\text{3N},J_{p'q'} J_{pq} J} = \langle p'q'r' : J_{p'q'} J'M' | V^{\text{3N}} | pqr : J_{pq} JM \rangle. \tag{21}$$

Since the uncoupled matrix elements with tilde indices can be computed from the J -coupled matrix elements, only calculating the J -coupled matrix elements is sufficient for many-body calculations. The relation between the uncoupled and J -coupled matrix elements are the following.

$$T_{\tilde{p}'\tilde{p}} = T_{p'p} \delta_{m_{p'} m_p}, \tag{22}$$

$$T_{\tilde{p}'\tilde{q}'\tilde{p}\tilde{q}}^{\text{NN}} = \sum_J C_{m_{p'} m_{q'} M}^{j_{p'} j_{q'} J} C_{m_p m_q M}^{j_p j_q J} T_{p'q'pq}^{\text{NN},J}, \tag{23}$$

$$V_{\tilde{p}'\tilde{q}'\tilde{p}\tilde{q}}^{\text{NN}} = \sum_J C_{m_{p'} m_{q'} M}^{j_{p'} j_{q'} J} C_{m_p m_q M}^{j_p j_q J} V_{p'q'pq}^{\text{NN},J}, \tag{24}$$

$$V_{\tilde{p}'\tilde{q}'\tilde{r}'\tilde{p}\tilde{q}\tilde{r}}^{\text{3N}} = \sum_{J J_{p'q'} J_{pq}} C_{m_{p'} m_{q'} M_{p'q'}}^{j_{p'} j_{q'} J_{p'q'}} C_{m_p m_q M_{pq}}^{j_p j_q J_{pq}} \times C_{M_{p'q'} m_{r'} M}^{j_{p'q'} j_{r'} J} C_{M_{pq} m_r M}^{j_{pq} j_r J} V_{p'q'r'pqr}^{\text{3N},J_{p'q'} J_{pq} J}. \tag{25}$$

2.3 Matrix elements of kinetic terms

The one-body kinetic element $T_{p'p}$ is given by

$$T_{p'p} = \frac{A-1}{A} \delta_{l_{p'} l_p} \delta_{j_{p'} j_p} \delta_{t_{z,p'} t_{z,p}} \frac{\hbar\omega}{2} \times \left[\left(2n_p + l_p + \frac{3}{2} \right) \delta_{n_{p'} n_p} + \sqrt{n_p \left(n_p + l_p + \frac{1}{2} \right)} \delta_{n_{p'} n_p - 1} \right]$$

$$+ \sqrt{(n_p + 1) \left(n_p + l_p + \frac{3}{2} \right)} \delta_{n_p, n_p+1} \Big]. \quad (26)$$

Note that the one-body kinetic operator takes the tridiagonal form. Also, the two-body kinetic matrix element can be computed through the non-antisymmetrized J -coupled matrix element $\bar{T}_{p'q'pq}^{NN,J}$

$$\begin{aligned} \bar{T}_{p'q'pq}^{NN,J} &= (-1)^{j_{q'}+j_p+J} \frac{\hbar^2}{Am} \\ &\times \left\{ \begin{matrix} j_{p'} & j_{q'} & J \\ j_q & j_p & 1 \end{matrix} \right\} \langle p' \| \nabla \| p \rangle \langle q' \| \nabla \| q \rangle, \end{aligned} \quad (27)$$

with the reduced matrix element of the gradient operator, which is given by

$$\begin{aligned} \langle p' \| \nabla \| p \rangle &= (-1)^{l'_p+j_p+1/2} \frac{1}{b} \sqrt{[j_{p'}][j_p]} \left\{ \begin{matrix} j_{p'} & j_p & 1 \\ l_p & l_{p'} & 1/2 \end{matrix} \right\} \\ &\times \left[\sqrt{(l_p + 1)(n_p + l_p + 3/2)} \delta_{n_p, n_p} \delta_{l_p, l_p+1} \right. \\ &+ \sqrt{(l_p + 1)n_p} \delta_{n_p, n_p-1} \delta_{l_p, l_p+1} \\ &+ \sqrt{l_p(n_p + l_p + 1/2)} \delta_{n_p, n_p} \delta_{l_p, l_p-1} \\ &\left. + \sqrt{l_p(n_p + 1)} \delta_{n_p, n_p+1} \delta_{l_p, l_p-1} \right]. \end{aligned} \quad (28)$$

Here, the HO length parameter $b^2 \equiv \hbar/m\omega$ is introduced with the HO frequency ω . The antisymmetrized matrix element is obtained as

$$T_{p'q'pq}^{NN,J} = \sqrt{\frac{1}{(1 + \delta_{p'q'}) (1 + \delta_{pq})}} \left[\bar{T}_{p'q'pq}^{NN,J} - (-1)^{j_{p'}+j_q-J} \bar{T}_{p'q'qp}^{NN,J} \right], \quad (29)$$

with

$$\delta_{pq} = \delta_{n_p n_q} \delta_{l_p l_q} \delta_{j_p j_q} \delta_{t_z, p t_z, q}. \quad (30)$$

The main tasks remaining are to compute the matrix elements $V_{p'q'pq}^{NN,J}$ and $V_{p'q'r'pqr}^{3N, J_{p'q'} J_{pq} J}$.

2.4 Nucleon–nucleon matrix elements

We begin with the NN matrix element. One might think that the matrix element can be calculated directly from the integral using the single-particle HO wave function. It is actually done in quantum chemistry. However, this would be a computationally expensive task since functional forms of NN interactions are complicated. Instead, the Talmi–Moshinsky transformation is widely used in nuclear physics:

$$\begin{aligned} V_{p'q'pq}^{NN,J} &= \sum_{N_{cm}^{NN} L_{cm}^{NN} J_{rel}^{NN}} \sum_{S n' l' n l} T_{N_{cm}^{NN} L_{cm}^{NN} n' l' S J_{rel}^{NN}}^{p' q' J} \\ &\times V_{n' l' n l}^{S J_{rel}^{NN}} T_{N_{cm}^{NN} L_{cm}^{NN} n l S J_{rel}^{NN}}^{pq J}. \end{aligned} \quad (31)$$

The quantum numbers introduced for the transformation N_{cm}^{NN} , L_{cm}^{NN} , n , l , S , and J_{rel}^{NN} are the NN cm radial quantum number, NN cm orbital angular momentum, relative radial quantum number, relative orbital angular momentum, total spin, and total angular momentum of the relative motion, respectively. The transformation coefficient $T_{N_{cm}^{NN} L_{cm}^{NN} n l S J_{rel}^{NN}}^{pq J}$ is

$$\begin{aligned} T_{N_{cm}^{NN} L_{cm}^{NN} n l S J_{rel}^{NN}}^{pq J} &= (-1)^{L_{cm}^{NN}+l+S+J} \sqrt{[j_p][j_q][S][J_{rel}^{NN}]} \\ &\times \sum_{\Lambda} [\Lambda] \left\{ \begin{matrix} l_p & 1/2 & j_p \\ l_q & 1/2 & j_q \\ \Lambda & S & J \end{matrix} \right\} \left\{ \begin{matrix} L_{cm}^{NN} & l & \Lambda \\ S & J & J_{rel}^{NN} \end{matrix} \right\} \\ &\times \langle N_{cm}^{NN} L_{cm}^{NN} n l : \Lambda | n_p l_p n_q l_q : \Lambda \rangle_1. \end{aligned} \quad (32)$$

In the above equation, $9j$ -symbol is used with the standard notation [25]. The symbol $\langle N L n l : \Lambda | n_1 l_1 n_2 l_2 : \Lambda \rangle_d$ is the HO bracket defined with the notation in Ref. [24]. The inner summations in Eq. (31) can be performed with an efficient matrix multiplication. Note that the antisymmetrization is not taken into account here. However, it is trivial and can be done by multiplying the factor f_{pq} to Eq. (32):

$$f_{pq} = \begin{cases} 1, & t_{z,p} \neq t_{z,q} \\ \sqrt{\frac{1}{2(1+\delta_{pq})}} [1 + (-1)^{l+S}], & t_{z,p} = t_{z,q} \end{cases}. \quad (33)$$

The NN matrix element in the relative HO basis $V_{n' l' n l}^{S J_{rel}^{NN}}$ can be obtained through the integral:

$$\begin{aligned} V_{n' l' n l}^{S J_{rel}^{NN}} &= \int d\pi'_1 d\pi_1 \pi_1'^2 \pi_1^2 R_{n' l'}(\pi'_1) R_{n l}(\pi_1) \\ &\times V_{l' l}^{S J_{rel}^{NN}}(\pi'_1, \pi_1), \end{aligned} \quad (34)$$

with the radial HO wave function:

$$\begin{aligned} R_{n l}(\pi_1) &= (-1)^n b \sqrt{\frac{2b\Gamma(n+1)}{\Gamma(n+l+3/2)}} \\ &\times (\pi_1 b)^l e^{-\pi_1^2 b^2/2} L_n^{(l+1/2)}(\pi_1^2 b^2). \end{aligned} \quad (35)$$

The gamma function $\Gamma(x)$ and associated Laguerre polynomial $L_n^{(\alpha)}(x)$ are introduced. The momentum π_1 is $\pi_1 = |(\mathbf{p}_1 - \mathbf{p}_2)|/\sqrt{2}$, consistent with the definition of the HO bracket. Note that π_1 is different from the usual relative momentum definition $p = |\mathbf{p}_1 - \mathbf{p}_2|/2$.

The NuHam1 code requires the input file for $V_{l' l}^{S J_{rel}^{NN}}(p', p)$ stored as a function of p' and p . Some selected interactions are given in the `input_nn_files` directory. The available NN interactions are LO–N⁴LO with 500 MeV regulator cutoff by Entem–Machleidt–Nosyk [26], N³LO with 500 MeV regulator cutoff by Entem–Machleidt [27], N²LO_{opt} [28], N²LO_{sat} [29], and Δ -full EFT series by the Gothenburg–Oak Ridge collaboration [30].

2.5 Three-nucleon matrix elements

For computational reasons, the 3N matrix elements are calculated within the isospin formalism. The matrix elements with the proton-neutron basis can be obtained through the JT -coupled matrix element $V_{p'q'r'pqr}^{3N, J_{p'q'} J_{pq} J, T_{p'q'} T_{pq} T}$:

$$V_{p'q'r'pqr}^{3N, J_{p'q'} J_{pq} J} = \sum_{T_{p'q'} T_{pq} T} C_{t_z, p' t_z, q' T_z, p' q'}^{t_{p' t_q} T_{p' q'}} C_{t_z, p t_z, q T_z, pq}^{t_{p t_q} T_{pq}} \times C_{T_z, p' q' t_z, r' T_z}^{T_{p' q' r} T} C_{T_z, pq t_z, r T_z}^{T_{pq r} T} V_{p'q'r'pqr}^{3N, J_{p'q'} J_{pq} J, T_{p'q'} T_{pq} T} \quad (36)$$

The recoupling should be done in many-body calculations, and the goal here is to obtain the JT -coupled matrix element. Note that recoupling coefficients from isospin structure have to be considered for the permutation of indices, similar to Eqs. (13)–(17).

Since the antisymmetrization of the 3N basis is more complicated than that of the NN basis, the 3N matrix element is cumbersome. The antisymmetrized basis is expressed as the linear combination of the non-antisymmetrized basis:

$$|Ei J_{rel}^{3N} T\rangle = \sum_{\beta} c_{i\beta} |E\beta J_{rel}^{3N} T\rangle, \quad (37)$$

where E , i , and J_{rel}^{3N} are the HO principle quantum number, label distinguishing the states, and total Jacobi angular momentum, respectively. The collective index β ,

$$\beta = \{n_{12}, l_{12}, s_{12}, j_{12}, t_{12}, n_3, l_3, j_3\}, \quad (38)$$

specifies the non-antisymmetrized basis. The quantum numbers with the subscript ‘12’, $n_{12}, l_{12}, s_{12}, j_{12}$, and t_{12} are for the relative motion of nucleons 1 and 2, i.e., the nodal, orbital angular momentum, spin, total angular momentum, and total isospin quantum numbers, respectively. Likewise, n_3, l_3 , and j_3 are the quantum numbers for the nucleon 3 with respect to the cm of the nucleons 1 and 2. Note that the principle quantum number is defined as $E = 2n_{12} + l_{12} + 2n_3 + l_3$. The coefficient in the linear combination $c_{i\beta}$ can be obtained by the diagonalization of the antisymmetrizer [31, 32]:

$$\mathcal{A}|Ei J_{rel}^{3N} T\rangle = A_i |E\beta J_{rel}^{3N} T\rangle, \quad (39)$$

with $\mathcal{A} = (1 + \mathcal{T}_{13}\mathcal{T}_{12} + \mathcal{T}_{12}\mathcal{T}_{23} - \mathcal{T}_{12} - \mathcal{T}_{13} - \mathcal{T}_{23})/6$ defined with the exchange operator \mathcal{T}_{ij} and the eigenvalue A_i . The matrix element of the antisymmetrizer is [31, 32]

$$\begin{aligned} &\langle E'\beta' J_{rel}^{3N} T' | \mathcal{A} | E\beta J_{rel}^{3N} T \rangle \\ &= \left[\frac{\delta_{\beta'\beta}}{3} - \frac{2}{3} (-1)^{s'_{12} + t'_{12} + s_{12} + t_{12}} \sum_{\Lambda S} [\Lambda][S] \right. \\ &\quad \left. \times \sqrt{[s'_{12}][j'_{12}][j'_3][t'_{12}]} \sqrt{[s_{12}][j_{12}][j_3][t_{12}]} \right] \end{aligned}$$

$$\begin{aligned} &\times \left\{ \begin{matrix} l'_{12} & s'_{12} & j'_{12} \\ l_3 & 1/2 & j'_3 \\ \Lambda & S & J \end{matrix} \right\} \left\{ \begin{matrix} l_{12} & s_{12} & j_{12} \\ l_3 & 1/2 & j_3 \\ \Lambda & S & J \end{matrix} \right\} \\ &\times \left\{ \begin{matrix} 1/2 & 1/2 & s'_{12} \\ 1/2 & S & s_{12} \end{matrix} \right\} \left\{ \begin{matrix} 1/2 & 1/2 & t'_{12} \\ 1/2 & T & t_{12} \end{matrix} \right\} \\ &\times \langle n'_{12} l'_{12} n'_3 l'_3 : \Lambda | n_{12} l_{12} n_3 l_3 \Lambda \rangle_{1/3} \Big] \\ &\times \delta_{E'E} \delta_{J_{rel}^{3N} J_{rel}^{3N}} \delta_{T'T}. \quad (40) \end{aligned}$$

The eigenvalue problem can be separated into $\{E, J_{rel}^{3N}, T\}$ blocks. Due to the overcompleteness of the non-antisymmetrized basis, the eigenvalue A_i is either 0 or 1, and A_i for the physical state has to be 1. Therefore, we always see $N_A \leq N_{NA}$ where N_A and N_{NA} are the basis numbers in the antisymmetrized and non-antisymmetrized bases within the $\{E, J_{rel}^{3N}, T\}$ block, respectively. In the code, all the 3N operators are stored with the antisymmetrized basis rather than the non-antisymmetrized basis, as it is computationally easier to handle. As another option for the 3N antisymmetrization, one may apply the permutator operator to the 3N momentum state as introduced in Ref. [33].

Similarly to the NN case, the 3N matrix elements can be obtained through the three-body Talmi–Moshinsky transformation:

$$V_{p'q'r'pqr}^{3N, J_{p'q'} J_{pq} J, T_{p'q'} T_{pq} T} = 6 \sum_{N_{cm}^{3N} L_{cm}^{3N} E'i' Ei} \sum_{E'i' Ei} \times T_{N_{cm}^{3N} L_{cm}^{3N} E'i' J_{rel}^{3N}}^{p' q' r' J_{p' q'} J T_{p' q'} T} V_{E'i' Ei}^{J_{rel}^{3N} T} T_{N_{cm}^{3N} L_{cm}^{3N} E'i J_{rel}^{3N}}^{pqr J_{pq} J T_{pq} T}, \quad (41)$$

where N_{cm}^{3N} and L_{cm}^{3N} denote the 3N cm radial quantum number and 3N cm orbital angular momentum, respectively. The matrix element $V_{E'i' Ei}^{J_{rel}^{3N} T}$ is a shorthand notation of $\langle E'i' J_{rel}^{3N} T | V | E i J_{rel}^{3N} T \rangle$. The transformation coefficient is given by

$$T_{N_{cm}^{3N} L_{cm}^{3N} E'i J_{rel}^{3N}}^{pqr J_{pq} J T_{pq} T} = \sum_{\beta} c_{i\beta} T_{N_{cm}^{3N} L_{cm}^{3N} E\beta J_{rel}^{3N}}^{pqr J_{pq} J T_{pq} T}, \quad (42)$$

with

$$\begin{aligned} &T_{N_{cm}^{3N} L_{cm}^{3N} E\beta J_{rel}^{3N}}^{pqr J_{pq} J T_{pq} T} = \delta_{T_{pq} t_{12}} (-1)^{L_{cm}^{3N} + j_3 + 3/2} \sqrt{[J_{pq}][j_r][J_{rel}^{3N}][j_3]} \\ &\times \sum_{N_{cm}^{NN} L_{cm}^{NN}} T_{N_{cm}^{NN} L_{cm}^{NN} n_{12} l_{12} s_{12} j_{12}}^{pq J_{pq}} \sum_{\Lambda} (-1)^{\Lambda} [\Lambda] \\ &\times \langle N_{cm}^{3N} L_{cm}^{3N} n_3 l_3 : \Lambda | N_{cm}^{NN} L_{cm}^{NN} n_r l_r : \Lambda \rangle_2 \\ &\times \left\{ \begin{matrix} j_{12} & L_{cm}^{NN} & \Lambda & L_{cm}^{3N} \\ J_{pq} & l_r & l_3 & J_{rel}^{3N} \\ J & j_r & 1/2 & j_3 \end{matrix} \right\}. \quad (43) \end{aligned}$$

The $12j$ -symbol of the first kind [25] is used. Note that $T_{N_{cm}^{NN} L_{cm}^{NN} n_{12} l_{12} s_{12} j_{12}}^{pq J_{pq}}$ is defined in Eq. (32), and one can find a recursive relation for the N -body Talmi–Moshinsky transformation with the $12j$ -symbol and HO bracket.

A typical limit of the three-body Talmi–Moshinsky transformation is $E_{3\max} = 16$, where $E_{3\max}$ is defined as $\max(2n_p + l_p + 2n_q + l_q + 2n_r + l_r)$. This limit does not allow us to obtain converged results for heavier systems with $A \gtrsim 100$. Recently, the limit was extended to $E_{3\max} = 28$ [34], by only computing the matrix elements relevant to the normal-ordered two-body (NO2B) approximation, which is widely used in basis expansion methods. For further details about the NO2B matrix elements, see Ref. [34]. The code supports this format as well.

The 3N matrix element in the Jacobi HO basis $V_{E'i'Ei}^{J3NT}$ can be obtained from the matrix element expressed with the non-antisymmetrized basis:

$$V_{E'i'Ei}^{J3NT} = \sum_{\beta'\beta} c_{i'\beta'} c_{i\beta} V_{\beta'\beta}^{J3NT}, \tag{44}$$

with

$$V_{\beta'\beta}^{J3NT} = \int d\pi'_1 d\pi'_2 d\pi_1 d\pi_2 \pi_1'^2 \pi_2'^2 \pi_1^2 \pi_2^2 \times R_{n_1' l_1' l_2'}(\pi_1') R_{n_1 l_1 l_2}(\pi_1) R_{n_3 l_3}(\pi_2) \times V_{\alpha'\alpha}^{J3NT}(\pi_1', \pi_2', \pi_1, \pi_2). \tag{45}$$

Here, the collective index α is introduced as

$$\alpha = \{l_{12}, s_{12}, j_{12}, t_{12}, l_3, j_3\}. \tag{46}$$

The momentum π_2 is defined as $\pi_2 = \sqrt{2/3}|(\mathbf{p}_1 + \mathbf{p}_2)/2 - \mathbf{p}_3|$. The momentum-space matrix element includes a regulator function:

$$V_{\alpha'\alpha}^{J3NT}(\pi_1', \pi_2', \pi_1, \pi_2) = f_\Lambda V_{\chi\text{EFT},\alpha'\alpha}^{J3NT}(\pi_1', \pi_2', \pi_1, \pi_2). \tag{47}$$

The regulator function takes either a non-local form

$$f_{\Lambda\text{nonlocal}} = \exp\left[-\left(\frac{\pi_1'^2 + \pi_2'^2}{2\Lambda_{\text{nonlocal}}^2}\right)^n\right] \exp\left[-\left(\frac{\pi_1^2 + \pi_2^2}{2\Lambda_{\text{nonlocal}}^2}\right)^n\right], \tag{48}$$

a local form

$$f_{\Lambda\text{local}} = \exp\left[-\left(\frac{|\mathbf{p}'_1 - \mathbf{p}_1|^2}{\Lambda_{\text{local}}^2}\right)^n\right] \exp\left[-\left(\frac{|\mathbf{p}'_2 - \mathbf{p}_2|^2}{\Lambda_{\text{local}}^2}\right)^n\right], \tag{49}$$

or a semilocal form [35] that is not supported in the code. The code fully supports the locally regulated matrix elements at N²LO in chiral EFT based on Ref. [36]. Also, a newly introduced local-non-local regularized form, $f_\Lambda = f_{\Lambda\text{local}} f_{\Lambda\text{nonlocal}}$ [37], is supported.

For non-local matrix elements, we have tried to implement along Ref. [38]. However, we found a numerical instability in the higher angular momentum partial waves, which

would be related to the discussion made in Ref. [39]. For this reason, non-local matrix elements are not fully supported, and external input files are required. The code has the capability to read the momentum-space matrix element $V_{\chi\text{EFT},\alpha'\alpha}^{J3NT}(p', q', p, q)$ ¹ from the HDF files by Hebeler et al. [12,40]. In the integral (45), the cubic b-spline interpolation is used to capture the oscillating nature of the HO wave functions.

3 Similarity renormalization group

The momentum scales of chiral EFT interactions are significantly lower than those of the other potential models such as AV18 [6] and CD-Bonn [7]. However, the momentum scale is not sufficiently low to obtain converged results in the many-body calculations. To accelerate convergence, one sometimes softens nuclear interactions. Softening procedures are well summarized for example in Ref. [43]. Here, we briefly review the widely used similarity renormalization group (SRG) approach.

In the SRG, we consider a unitary transformation depending on a continuous parameter α :

$$H(\alpha) = U^\dagger(\alpha) H U(\alpha). \tag{50}$$

The SRG flow equation can be obtained by differentiating both sides:

$$\frac{dH(\alpha)}{d\alpha} = [\eta(\alpha), H(\alpha)], \quad \eta(\alpha) = \frac{dU^\dagger(\alpha)}{d\alpha} U(\alpha). \tag{51}$$

The antihermitian operator $\eta(\alpha)$ is known as the generator of the flow equation and can be chosen flexibly [44]. The most widely used choice is $\eta(\alpha) = [T_{\text{kin}}, H(\alpha)]$ with the kinetic operator T_{kin} , which guarantees the suppression of the coupling between low and high momenta. The flow equation (51) is integrated until the coupling is sufficiently suppressed. Since the unitary transformation does not change the eigenvalues, the SRG can be regarded as a reshuffling of the NN, 3N, and many-body sectors. In other words, the many-body interactions are induced by the SRG evolution even if the original interaction includes only NN interactions.

In practical applications, we extract the SRG-evolved interactions with the subtraction method (see Ref. [41] for

¹ In the HDF files, the matrix elements are given in terms of the Jacobi variables p and q [12], defined as

$$p = \frac{1}{2} |\mathbf{p}_1 - \mathbf{p}_2| = \sqrt{\frac{1}{2}} \pi_1,$$

and

$$q = \frac{1}{3} |-\mathbf{p}_1 - \mathbf{p}_2 + 2\mathbf{p}_3| = \sqrt{\frac{2}{3}} |\pi_2|.$$

Table 1 The ground-state energies of selected doubly magic nuclei computed with the NCSM and IMSRG(2). The resolution scale λ is related with the end point of the SRG evolution: $\lambda = \alpha^{-1/4}$ in units of fm^{-1} . The N_{max} and e_{max} truncations are employed in the NCSM and IMSRG(2), respectively. The truncation defined by $E_{3\text{max}} = \max(e_1 + e_2 + e_3)$ is applied for the input 3N matrix elements. The $E_{3\text{max}}$ values with asterisk indicates that the 3N matrix elements are computed within the half-precision floating-point numbers. Also, $E_{3\text{max}} = \text{“none”}$ indicates that there are no input 3N matrix elements.

The entry $\hbar\omega$ is the basis frequency used in the many-body calculations, and the numbers in the parentheses are the parent frequency adopted in the frequency conversion for the 3N matrix elements [41]. For ${}^4\text{He}$, ${}^{16}\text{O}$, and ${}^{40}\text{Ca}$, the 3N SRG evolution was done in the ramp A space defined in Ref. [41]. For ${}^{132}\text{Sn}$, the evolution was done in the $N_{\text{max}} = 48$ space only for $J_{\text{rel}}^{3N} \leq 13/2$ channels. The star symbol in the $E_{3\text{max}}$ column indicates that the 3N matrix elements are in half-precision floating point numbers

Nucleus	Interaction	λ (fm^{-1})	Method	$N_{\text{max}}/e_{\text{max}}$	$E_{3\text{max}}$	$\hbar\omega$ (MeV)	$E_{\text{g.s.}}$ (MeV)
${}^4\text{He}$	EM500 [27]	1.8	NCSM	16	None	16	-28.45
${}^4\text{He}$	EM500 [27]	2.0	NCSM	16	None	16	-28.23
${}^4\text{He}$	EM500 [27]	2.2	NCSM	16	None	16	-27.88
${}^4\text{He}$	EM500 [27]	1.8	IMSRG(2)	14	None	16	-28.43
${}^4\text{He}$	EM500 [27]	2.0	IMSRG(2)	14	None	16	-28.28
${}^4\text{He}$	EM500 [27]	2.2	IMSRG(2)	14	None	16	-28.08
${}^4\text{He}$	EM500 [27]	1.8	IMSRG(2)	14	16	16(30)	-25.61
${}^4\text{He}$	EM500 [27]	2.0	IMSRG(2)	14	16	16(30)	-25.76
${}^4\text{He}$	EM500 [27]	2.2	IMSRG(2)	14	16	16(30)	-25.96
${}^4\text{He}$	$\text{N}^3\text{LO}_{\text{nl}}$ [37]	1.8	IMSRG(2)	14	16	16(30)	-28.63
${}^4\text{He}$	$\text{N}^3\text{LO}_{\text{nl}}$ [37]	2.0	IMSRG(2)	14	16	16(30)	-28.69
${}^4\text{He}$	$\text{N}^3\text{LO}_{\text{nl}}$ [37]	2.2	IMSRG(2)	14	16	16(30)	-28.76
${}^4\text{He}$	1.8/2.0 (EM) [42]	None	IMSRG(2)	14	16	16	-29.23
${}^4\text{He}$	$\Delta\text{N}^2\text{LO}_{\text{Go}}$ (394) [30]	None	IMSRG(2)	14	16	16	-28.67
${}^{16}\text{O}$	EM500 [27]	1.8	IMSRG(2)	14	None	16	-173.3
${}^{16}\text{O}$	EM500 [27]	2.0	IMSRG(2)	14	None	16	-165.7
${}^{16}\text{O}$	EM500 [27]	2.2	IMSRG(2)	14	None	16	-158.5
${}^{16}\text{O}$	EM500 [27]	1.8	IMSRG(2)	14	16	16(30)	-121.6
${}^{16}\text{O}$	EM500 [27]	2.0	IMSRG(2)	14	16	16(30)	-122.9
${}^{16}\text{O}$	EM500 [27]	2.2	IMSRG(2)	14	16	16(30)	-124.2
${}^{16}\text{O}$	$\text{N}^3\text{LO}_{\text{nl}}$ [37]	1.8	IMSRG(2)	14	16	16(30)	-128.6
${}^{16}\text{O}$	$\text{N}^3\text{LO}_{\text{nl}}$ [37]	2.0	IMSRG(2)	14	16	16(30)	-127.2
${}^{16}\text{O}$	$\text{N}^3\text{LO}_{\text{nl}}$ [37]	2.2	IMSRG(2)	14	16	16(30)	-126.0
${}^{16}\text{O}$	1.8/2.0 (EM) [42]	None	IMSRG(2)	14	16	16	-127.2
${}^{16}\text{O}$	$\Delta\text{N}^2\text{LO}_{\text{Go}}$ (394) [30]	None	IMSRG(2)	14	16	16	-126.1
${}^{40}\text{Ca}$	EM500 [27]	1.8	IMSRG(2)	14	None	16	-639.4
${}^{40}\text{Ca}$	EM500 [27]	2.0	IMSRG(2)	14	None	16	-595.8
${}^{40}\text{Ca}$	EM500 [27]	2.2	IMSRG(2)	14	None	16	-554.1
${}^{40}\text{Ca}$	EM500 [27]	1.8	IMSRG(2)	14	16	16(30)	-352.3
${}^{40}\text{Ca}$	EM500 [27]	2.0	IMSRG(2)	14	16	16(30)	-360.0
${}^{40}\text{Ca}$	EM500 [27]	2.2	IMSRG(2)	14	16	16(30)	-366.7
${}^{40}\text{Ca}$	$\text{N}^3\text{LO}_{\text{nl}}$ [37]	1.8	IMSRG(2)	14	16	16(30)	-347.1
${}^{40}\text{Ca}$	$\text{N}^3\text{LO}_{\text{nl}}$ [37]	2.0	IMSRG(2)	14	16	16(30)	-341.9
${}^{40}\text{Ca}$	$\text{N}^3\text{LO}_{\text{nl}}$ [37]	2.2	IMSRG(2)	14	16	16(30)	-336.7
${}^{40}\text{Ca}$	1.8/2.0 (EM) [42]	None	IMSRG(2)	14	16	16	-344.4
${}^{40}\text{Ca}$	$\Delta\text{N}^2\text{LO}_{\text{Go}}$ (394) [30]	None	IMSRG(2)	14	16	16	-339.1

Table 1 continued

Nucleus	Interaction	λ (fm ⁻¹)	Method	N_{\max}/e_{\max}	$E_{3\max}$	$\hbar\omega$ (MeV)	$E_{g.s.}$ (MeV)
¹³² Sn	N ³ LO _{nl} [37]	2.0	IMSRG(2)	14	24	12 (30)	-1064
¹³² Sn	1.8/2.0 (EM) [42]	None	IMSRG(2)	14	24	12	-1109
¹³² Sn	Δ N ² LO _{GO} (394) [30]	None	IMSRG(2)	14	24	12	-1098
²⁰⁸ Pb	1.8/2.0 (EM) [42]	None	IMSRG(2)	14	28*	12	-1660
²⁰⁸ Pb	Δ N ² LO _{GO} (394) [30]	None	IMSRG(2)	14	28*	12	-1628

example). For NN, the Hamiltonian $H_{\text{NN}} = T_{\text{NN}} + V_{\text{NN}}$ is evolved, and the evolved NN interaction is given as $V_{\text{NN}}(\alpha) = H_{\text{NN}}(\alpha) - T_{\text{kin}}$. Note that T_{kin} and V_{NN} are the NN kinetic and interaction operators, respectively. From the definition of the flow equation, $V_{\text{NN}}(\alpha)$ reproduces the two-body observables obtained through the original interaction V_{NN} . However, this is not true for three- and many-body observables due to missing induced many-body forces, and the many-body observables have an artificial α dependence, showing how much the unitarity of the transformation is broken in the many-body space. To obtain more α -independent result, one has to include induced 3N interaction extracted from the 3N evolution.

For 3N, the starting Hamiltonian is $H_{3\text{N}} = T_{3\text{N}} + V_{\text{NN}}^{[3]}$, where $T_{3\text{N}}$ is the 3N kinetic operator, and $V_{\text{NN}}^{[3]}$ is the NN interaction embedded into the 3N space. Note that one can add an initial 3N term if required. The induced 3N term can be obtained as $V_{3\text{N,ind}}(\alpha) = H_{3\text{N}}(\alpha) - T_{3\text{N}} - V_{\text{NN}}^{[3]}(\alpha)$. Here, $V_{\text{NN}}^{[3]}(\alpha)$ is $V_{\text{NN}}(\alpha)$ obtained from the NN evolution and embedded into 3N space. As seen in the NN evolution, the 3N evolution preserves three-body observables. The same procedure can be applied for many-body terms, and the many-body evolution is needed until the α -dependence of the many-body observables becomes weak enough. However, in practice, even the four-body SRG evolution is too expensive to do due to the resulting basis dimension and the cost of antisymmetrizing the basis. In the NuHam11 code, the SRG evolution can be performed in the NN and 3N sectors, the current state-of-the-art.

The unitary transformation can be obtained from the flow equation for the transformation operator:

$$\frac{dU(\alpha)}{d\alpha} = -U(\alpha)\eta(\alpha). \quad (52)$$

However, a computationally more moderate way is used in practice, and the unitary transformation is obtained as

$$U(\alpha) = \sum_k |\psi_k\rangle\langle\psi_k(\alpha)|, \quad (53)$$

with the eigenstates of the original and evolved Hamiltonians:

$$H|\psi_k\rangle = E_k|\psi_k\rangle, \quad H(\alpha)|\psi_k(\alpha)\rangle = E_k|\psi_k(\alpha)\rangle. \quad (54)$$

Note that the relative phase of $|\psi_k\rangle$ and $|\psi_k(\alpha)\rangle$ cannot be determined in general, which affect the sign of the matrix element of the transformation operator. Since the SRG transformation does not change the wave function drastically, the relative phase is fixed such that $\langle\psi_k|\psi_k(\alpha)\rangle \geq 0$. In the same way as for the Hamiltonian, the induced three-body term of an operator can be computed in the NuHam11 code as done in Ref. [45]. The end point of the flow equation is usually parametrized by the momentum scale $\lambda = \alpha^{-1/4}$ instead of α , and we follow this convention.

In the code, the SRG evolution is done in a relative-coordinate HO space because the consistent evolution of the other operators is straightforward. This means that the evolution is done in the truncated HO space, and the N_{\max} truncation is employed in the code. The N_{\max} is defined as

$$N_{\max} = \begin{cases} \max(2n + l), & \text{NN system} \\ \max(2n_{12} + l_{12} + 2n_3 + l_3), & \text{3N system} \end{cases}. \quad (55)$$

The UV momentum scale in the employed N_{\max} space is roughly estimated as $p_{\text{UV}} \sim \sqrt{2N_{\max}m\omega}$ [46], and we expect that the N_{\max} should be increased until p_{UV} is sufficiently larger than the cutoff scale of the interaction, typically 500 MeV. Although we can take sufficiently large N_{\max} for the NN evolution,² the 3N N_{\max} can be an issue especially in heavy nuclei calculations [34] even if the frequency conversion technique [41] is used.

4 Many-body results

Here, we show the ground-state energies for the selected doubly magic nuclei computed with the NCSM and two-body approximated IMSRG [IMSRG(2)] as a benchmark. We do not introduce the theoretical details of the many-body calculation methods. The details can be found in Refs. [15, 18, 47] and references therein. The numerical codes used here are open source; the NCSM and IMSRG are done with the BIGSTICK [20] and `imsrg++` [21] codes, respectively.

² We observed that HO- and momentum-space evolutions provide almost the same results. The code supports the momentum-space evolution only for NN interactions, and users can verify it.

In Table 1, the ground-state energies are shown, using the SRG-softened $N^3\text{LO}$ NN interaction [27], labeled by “EM500”, with and without induced 3N interaction, $N^3\text{LO}_{\text{int}}$ [37], 1.8/2.0 (EM) [42], and $\Delta N^2\text{LO}_{\text{GO}}(394)$ [30].

5 Program summary and specifications

The NuHamil code is written in modern Fortran. It requires a set of libraries, BLAS, LAPACK, GNU scientific library (gsl), zlib, and hdf5.

5.1 Installation

The source code can be downloaded from GitHub:

```
$ cd
$ git clone https://github.com/Takayuki-Miyagi/NuHamil-public.git
```

Note that downloading the code in the home directory is not mandatory, but it is recommended. One needs to download the submodules, linear algebra wrapper and b-spline interpolation:

```
$ cd NuHamil
$ git submodule init
$ git submodule update
```

The compilation can be done with the make command. The default compiler is GCC Fortran. If a user needs to use another compiler, the Makefile has to be edited appropriately. The symbolic link will be created by the make install command.³ Once the directory is added to PATH, the code is ready to run.

```
$ make
$ make install
$ echo 'export PATH=$PATH:$HOME/bin' >>
  $HOME/.bashrc
$ source $HOME/.bashrc
```

5.2 How to run

A job submission can be controlled by a Python script, and some sample scripts are prepared in the exe directory. For example, the NN and 3N matrix elements can be generated with NuHamil_2BME.py and NuHamil_3BME.py, respectively. A Python script generates the corresponding input file for NuHamil.exe and submits the job. If an user needs to run a job manually, it can be done with

```
$ NuHamil.exe input.txt
```

The “input.txt” is the input file based on the Fortran namelist functionality, and the file format is given the following.

```
&input
  variable1=value
  variable2=value
  ...
&end
```

5.3 Major parameters

In the NuHamil code, there are a number of input parameters. Here, we list some of the basic input parameters that the users might need to change depending on their requirements.⁴

- rank: integer, particle number of the system.
- hw: frequency of the HO basis in the unit of MeV. The typical range is $10 \lesssim hw \lesssim 40$.
- hw_target: target frequency for the frequency conversion technique [41]. The parameter is valid if $rank > 2$. To turn off the frequency conversion, set $hw_target = -1$.
- emax: $e_{\text{max}} = \max(2n + l)$ truncation for the output lab-frame HO matrix element file.
- e2max: $e_{2\text{max}} = \max(2n_p + l_p + 2n_q + l_q)$ truncation for the output lab-frame HO matrix element file. It is recommended to use $e2\text{max} = 2 \times e\text{max}$.
- e3max: $E_{3\text{max}} = \max(2n_p + l_p + 2n_q + l_q + 2n_r + l_r)$ truncation for the output 3N lab-frame HO matrix element file. A typical limit is $e3\text{max} = 16$, and it will not work for the larger e3max because of the memory requirements. If only the matrix elements relevant for the NO2B approximation are needed [34], $e3\text{max} = 24$ would be a typical choice without MPI parallelization.
- file_name_nn: file name of the output lab-frame NN HO matrix elements.
- file_name_3n: file name of the output lab-frame 3N HO matrix elements.
- renorm: renormalization method; “bare”, “srg”, and “Vlowk” are available. Note that the code does not support the 3N evolution for “Vlowk” option.
- renorm_space2: NN interaction renormalization space; “ho” and “mom” are available. The NN renormalization procedure is done in HO- (“ho”) or momentum- (“mom”)space. The default is “ho”.
- input_nn_file: file name of the NN interaction represented in the relative momentum space. The files are in the input_nn_files directory.
- NNInt: name of the NN interaction.
- N2max: N_{max} truncation for the NN system.
- only_no2b_element: If it is set True, only the matrix elements relevant for the NO2B approximation will be computed [34].

³ By default the link will be created in HOME/bin.

⁴ Users can contact the author for the parameters not listed here.

- `jmax3`: maximum value of the 3N Jacobi angular momentum taken into account. This has to be an integer and twice the actual angular momentum.
- `genuine_3bf`: set True if the bare 3N interaction needs to be included. If it is set False and `renorm="srg"`, the SRG induced 3N interaction will be computed.
- `Regulator`: 3N regulator functional form. One can choose “Local” (49), “NonLocal” (48), or “LNL” [37].
- `RegulatorPower`: power of the regulator function, i.e., n in Eqs. (48) and (49).
- `LECs`: 5 dimensional array providing the low-energy constants appear in N^2LO 3N interaction in the chiral EFT, $\{c_1, c_3, c_4, c_D, c_E\}$. Note that c_1, c_3 , and c_4 are in units of GeV^{-1} , while c_D and c_E are dimensionless. For more details, see Ref. [48].
- `lambda_3nf_nonlocal`: cutoff of the 3N non-local regulator, $\Lambda_{\text{nonlocal}}$ in Eq. (48), in the unit of MeV.
- `lambda_3nf_local`: cutoff of the 3N local regulator, Λ_{local} in Eq. (49), in units of MeV.

5.4 File format of input NN interactions in relative momentum space

As mentioned in Sect. 2.4, some selected NN interaction files are prepared in the `input_nn_files` directory. Furthermore, one can use their own momentum-space NN interaction. The file needs to be written with the binary.⁵ The file should begin with listing the following variables:

```
Number_of_mesh_points
J_max
Number_of_relative_coordinate_channels
```

where the 32-bit integers `Number_of_mesh_points`, `J_max`, and `Number_of_relative_coordinate_channels` are the size of momentum mesh points, maximum total angular momentum in the relative coordinate, i.e., $\max(J_{\text{rel}}^{\text{NN}})$, and the number of $[J_{\text{rel}}^{\text{NN}}, (-1)^l, S, t_{z,p} + t_{z,q}]$ combinations written in the file, respectively. Then, one needs to write the momentum mesh points and corresponding weights for a quadrature method, which are `number_of_mesh_points`-dimensional arrays with the 64-bit float:

```
momentum_mesh_points
weights
```

Finally, the momentum-space matrix for each $[J_{\text{rel}}^{\text{NN}}, (-1)^l, S, t_{z,p} + t_{z,q}]$ block should be written in the following way:

```
Angular_momentum
Parity
Spin
Isospin_z_component
Matrix_dimension
Momentum_space_matrix
```

⁵ For Fortran users, the file should not include any delimiters.

```
... (The same structure is repeated
Number_of_relative_coordinate_channels
times)
```

Here, the 32-bit integers `Angular_momentum`, `Parity`, `Spin`, `Isospin_z_component`, and `Matrix_dimension` correspond to $J_{\text{rel}}^{\text{NN}}$, $(-1)^l$, S , $t_{z,p} + t_{z,q}$, and the size of momentum-space matrix, respectively. Note that $t_{z,p} + t_{z,q}$ can take either of -1 (proton-proton), 0 (proton-neutron), or 1 (neutron-neutron). The `Momentum_space_matrix` is the flattened (“`Matrix_dimension`”)2-dimensional array with the 64-bit float, corresponding to $V_{ll}^{S, J_{\text{rel}}^{\text{NN}}}(p', p)$.⁶ Notice that `Number_of_mesh_points` and `Matrix_dimension` are not always the same because `Matrix_dimension` is twice of `Number_of_mesh_points` for spin-triplet coupled channels.

5.5 File format of non-local 3N matrix elements in Jacobi momentum space

As mentioned in Sect. 2.5, external input HDF5 files are needed for the non-local 3N matrix elements. The HDF5 files need to be prepared for each $[J_{\text{rel}}^{3N}, (-1)^{l_{12}+l_3}, T]$ partial waves and placed in the directory (`directory path`)/`T3_2T/J3_2J_{rel}^{3N}/PAR_(-1)^{l_{12}+l_3}`, e.g., `HOME/3NF_matrix_elements_nonlocal_V/T3_1/J3_1/PAR_1` for the ^3H and ^3He ground-state channel. Each HDF5 file must include the entries `Nalpha`, `Np`, `Nq`, `p mesh`, `q mesh`, `pw channels`, and `matrix elements`, written as the dataset type supported in the HDF5 format. The `Nalpha` is the 32-bit integer corresponding to the number of α channels. The objects `Np` and `Nq` are also 32-bit integers and the number of momentum mesh points for p and q , respectively. Note that α is defined in Eq. (46). The dataset `p mesh` should at least have `mesh point` and `mesh weight` entries, corresponding to the p -momentum mesh points and associated weights for a quadrature method, respectively. The objects `mesh point` and `mesh weight` should be `Np`-dimensional array with the 64-bit float. The dataset `q mesh` is the same as `p mesh` except that it is for q . Regarding the dataset `pw channels`, it should at least include `L_12`, `S_12`, `J_12`, `T_12`, `l_3`, and `2*j_3` entries, which are l_{12} , s_{12} , j_{12} , t_{12} , l_3 , and $2j_3$, respectively. Each entry has to be “`Nalpha`”-dimensional array with the 32-bit integer. Finally, the “`matrix elements`” corresponds to the 3N matrix stored as `[Nalpha, Nq, Np, Nalpha, Nq, Np]`-dimensional array with the 32-bit float.⁷

⁶ For further details, one can see the subroutine `read_nn_mom` in `NNForce.F90`.

⁷ Further details can be found in `NNFFromFile.F90`.

6 Summary and future perspective

We introduce the NuHamil code to generate NN and 3N matrix elements. The jobs can be managed with a simple Python script. The available NN interactions are LO – N^4 LO with 500 MeV regulator cutoff by Entem–Machleidt–Nosyk [26], N^3 LO with 500 MeV regulator cutoff by Entem–Machleidt [27], N^2 LO_{opt} [28], N^2 LO_{sat} [29], and Δ -full EFT series by Gothenburg–Oak Ridge collaboration [30]. The code can generate locally regulated 3N interactions [36]. Additional input files are needed for non-locally regulated 3N interactions. The code also supports the free-space NN and 3N SRG evolution, and the consistent evolution of the other operators is implemented. The output files can be used for the NCSM calculations with the BIGSTICK code [20] and IMSRG calculations with the `imsrg++` code [21].

For a comprehensive understanding of nuclear structure, interactions between a nucleus and external field should be addressed. For example the electromagnetic observables are results of the nucleus-photon interaction and are related with the multipole components of the electromagnetic current operators. Although we know that higher-order contributions are essential, see Refs. [49,50] for example, the LO current is used in most calculations due to the complexity of the matrix element calculations. As a future development, we plan to implement the higher-order current operators, including the two-body contributions.

Acknowledgements The NuHamil code is greatly inspired by the `manyeff` code by P. Navrátil and `VRenormalize` in the Computational Environment for Nuclear Structure (CENS) project. The code uses VODE library by G. D. Byrne and S. Thompson. The author thanks P. Navrátil, N. Shimizu, and N. Tsunoda for the discussions, optimizations, and parallelizations. The author also thanks P. Arthuis, A. Belly, M. Heinz, B. S. Hu, S. R. Stroberg, and A. Tichai for testing the code and useful feedback. This work was in part supported by JSPS KAKENHI Grant No. JP16J05707, the Program for Leading Graduate Schools, MEXT, Japan, the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 279384907 – SFB 1245, and the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (Grant Agreement No. 101020842). Also, the author was affiliated with TRIUMF receiving funding via a contribution through the National Research Council of Canada. The code was in part developed and tested with an allocation of computing resources on Cedar at WestGrid and Compute Canada, at the Jülich Supercomputing Center, and the Oak Cluster at TRIUMF managed by the University of British Columbia department of Advanced Research Computing (ARC).

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability Statement This manuscript has associated data in a data repository. [Authors’ comment: The numerical code developed in this work is available in the github repository <https://github.com/Takayuki-Miyagi/NuHamil-public>.]

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation,

distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. M. Hjorth-Jensen, T.T. Kuo, E. Osnes, Phys. Rep. 261, 125 (1995); CENS, <https://github.com/ManyBodyPhysics/CENS>
2. S. Yoshida, J. Open Source Softw. 7, 4694 (2022). <https://doi.org/10.21105/joss.04694>
3. R. Machleidt, Int. J. Mod. Phys. E 26, 1730005 (2017). <https://doi.org/10.1142/S0218301317300053>. arXiv:1710.07215v2
4. E. Epelbaum, H.-W. Hammer, U.-G. Meißner, Rev. Mod. Phys. 81, 1773 (2009). <https://doi.org/10.1103/RevModPhys.81.1773>. arXiv:0811.1338
5. R. Machleidt, D. Entem, Phys. Rep. 503, 1 (2011). <https://doi.org/10.1016/j.physrep.2011.02.001>. arXiv:1105.2919
6. R.B. Wiringa, V.G.J. Stoks, R. Schiavilla, Phys. Rev. C 51, 38 (1995). <https://doi.org/10.1103/PhysRevC.51.38>. arXiv:nucl-th/9408016 [nucl-th]
7. R. Machleidt, Phys. Rev. C 63, 024001 (2001). <https://doi.org/10.1103/PhysRevC.63.024001>. arXiv:nucl-th/0006014 [nucl-th]
8. E. Epelbaum, H. Krebs, U.-G. Meißner, Eur. Phys. J. A 51, 53 (2015). <https://doi.org/10.1140/epja/i2015-15053-8>. arXiv:1412.0142
9. R.J. Furnstahl, N. Klco, D.R. Phillips, S. Wesolowski, Phys. Rev. C 92, 024005 (2015). <https://doi.org/10.1103/PhysRevC.92.024005>
10. J.A. Melendez, S. Wesolowski, R.J. Furnstahl, Phys. Rev. C 96, 024003 (2017). <https://doi.org/10.1103/PhysRevC.96.024003>. arXiv:1704.03308
11. J.A. Melendez, R.J. Furnstahl, D.R. Phillips, M.T. Pratala, S. Wesolowski, Phys. Rev. C 100, 044001 (2019). <https://doi.org/10.1103/PhysRevC.100.044001>. arXiv:1904.10581
12. K. Hebeler, Phys. Rep. 890, 1 (2021). <https://doi.org/10.1016/j.physrep.2020.08.009>. arXiv:2002.09548
13. H. Hergert, Front. Phys. 8, 1 (2020). <https://doi.org/10.3389/fphy.2020.00379>
14. B.S. Hu, W. Jiang, T. Miyagi, Z.H. Sun, A. Ekström, C. Forssén, G. Hagen, J.D. Holt, T. Papenbrock, S.R. Stroberg, I. Vernon, Nat. Phys. 18, 1196 (2022). <https://doi.org/10.1038/s41567-022-01715-8>. arXiv:2112.01125
15. B.R. Barrett, P. Navrátil, J.P. Vary, Prog. Part. Nucl. Phys. 69, 131 (2013). <https://doi.org/10.1016/j.pnpnp.2012.10.003>
16. G. Hagen, T. Papenbrock, M. Hjorth-Jensen, D.J. Dean, Rep. Prog. Phys. 77, 096302 (2014). <https://doi.org/10.1088/0034-4885/77/9/096302>. arXiv:1312.7872
17. V. Somà, Front. Phys. 8, 1 (2020). <https://doi.org/10.3389/fphy.2020.00340>
18. H. Hergert, S.K. Bogner, T.D. Morris, A. Schwenk, K. Tsukiyama, Phys. Rep. 621, 165 (2016). <https://doi.org/10.1016/j.physrep.2015.12.007>. arXiv:1512.06956
19. A. Tichai, R. Roth, T. Duguet, Front. Phys. 8, 1 (2020). <https://doi.org/10.3389/fphy.2020.00164>
20. C.W. Johnson, W.E. Ormand, K.S. McElvain, H. Shan, arXiv:1801.08432
21. S.R. Stroberg, <https://github.com/ragnarstroberg/imsrg>

22. M. Moshinsky, Nucl. Phys. **13**, 104 (1959). [https://doi.org/10.1016/0029-5582\(59\)90143-9](https://doi.org/10.1016/0029-5582(59)90143-9)
23. L. Trlifaj, Phys. Rev. C **5**, 1534 (1972). <https://doi.org/10.1103/PhysRevC.5.1534>
24. G. Kamuntavičius, R. Kalinauskas, B. Barrett, S. Micevičius, D. Germanas, Nucl. Phys. A **695**, 191 (2001). [https://doi.org/10.1016/S0375-9474\(01\)01101-0](https://doi.org/10.1016/S0375-9474(01)01101-0). arXiv:nucl-th/0105009 [nucl-th]
25. D.A. Varshalovich, A.N. Moskalev, V.K. Khersonskii, Quantum Theory of Angular Momentum (World Scientific, Singapore (1988)). <https://doi.org/10.1142/0270>
26. D.R. Entem, R. Machleidt, Y. Nosyk, Phys. Rev. C **96**, 024004 (2017). <https://doi.org/10.1103/PhysRevC.96.024004>. arXiv:1703.05454
27. D.R. Entem, R. Machleidt, Phys. Rev. C **68**, 041001 (2003). <https://doi.org/10.1103/PhysRevC.68.041001>. arXiv:nucl-th/0304018 [nucl-th]
28. A. Ekström, G. Baardsen, C. Forssén, G. Hagen, M. Hjorth-Jensen, G.R. Jansen, R. Machleidt, W. Nazarewicz, T. Papenbrock, J. Sarich, S.M. Wild, Phys. Rev. Lett. **110**, 192502 (2013). <https://doi.org/10.1103/PhysRevLett.110.192502>. arXiv:1303.4674
29. A. Ekström, G.R. Jansen, K.A. Wendt, G. Hagen, T. Papenbrock, B.D. Carlsson, C. Forssén, M. Hjorth-Jensen, P. Navrátil, W. Nazarewicz, Phys. Rev. C **91**, 051301 (2015). <https://doi.org/10.1103/PhysRevC.91.051301>. arXiv:1502.04682
30. W.G. Jiang, A. Ekström, C. Forssén, G. Hagen, G.R. Jansen, T. Papenbrock, Phys. Rev. C **102**, 054301 (2020). <https://doi.org/10.1103/PhysRevC.102.054301>. arXiv:2006.16774
31. P. Navrátil, B.R. Barrett, Phys. Rev. C **59**, 1906 (1999). <https://doi.org/10.1103/PhysRevC.59.1906>
32. P. Navrátil, G.P. Kamuntavičius, B.R. Barrett, Phys. Rev. C **61**, 044001 (2000). <https://doi.org/10.1103/PhysRevC.61.044001>. arXiv:nucl-th/9907054 [nucl-th]
33. W. Glöckle, The Quantum Mechanical Few-Body Problem (Springer, Berlin (1983)). <https://doi.org/10.1007/978-3-642-82081-6>
34. T. Miyagi, S.R. Stroberg, P. Navrátil, K. Hebeler, J.D. Holt, Phys. Rev. C **105**, 014302 (2022). <https://doi.org/10.1103/PhysRevC.105.014302>
35. E. Epelbaum, J. Golak, K. Hebeler, H. Kamada, H. Krebs, U.-G. Meißner, A. Nogga, P. Reinert, R. Skibiński, K. Topolnicki, Y. Volkotrub, H. Witała, Eur. Phys. J. A **56**, 92 (2020). <https://doi.org/10.1140/epja/s10050-020-00102-2>. arXiv:1907.03608
36. P. Navrátil, Few-Body Syst. **41**, 117 (2007). <https://doi.org/10.1007/s00601-007-0193-3>. arXiv:0707.4680
37. V. Somà, P. Navrátil, F. Raimondi, C. Barbieri, T. Duguet, Phys. Rev. C **101**, 014318 (2020). <https://doi.org/10.1103/PhysRevC.101.014318>. arXiv:1907.09790
38. T. Fukui, L. De Angelis, Y.Z. Ma, L. Coraggio, A. Gargano, N. Itaco, F.R. Xu, Phys. Rev. C **98**, 044305 (2018). <https://doi.org/10.1103/PhysRevC.98.044305>
39. D. Hüber, H. Witała, A. Nogga, W. Glöckle, H. Kamada, Few Body Syst. **22**, 107 (1997). <https://doi.org/10.1007/s006010050057>. arXiv:9611021 [nucl-th]
40. K. Hebeler, H. Krebs, E. Epelbaum, J. Golak, R. Skibiński, Phys. Rev. C **91**, 044001 (2015). <https://doi.org/10.1103/PhysRevC.91.044001>. arXiv:1502.02977
41. R. Roth, A. Calci, J. Langhammer, S. Binder, Phys. Rev. C **90**, 024325 (2014). <https://doi.org/10.1103/PhysRevC.90.024325>. arXiv:1311.3563
42. K. Hebeler, S.K. Bogner, R.J. Furnstahl, A. Nogga, A. Schwenk, Phys. Rev. C **83**, 031301 (2011). <https://doi.org/10.1103/PhysRevC.83.031301>. arXiv:1012.3381
43. S.K. Bogner, R.J. Furnstahl, A. Schwenk, Prog. Part. Nucl. Phys. **65**, 94 (2010). <https://doi.org/10.1016/j.ppnp.2010.03.001>
44. F. Wegner, Annalen der Physik **506**, 77 (1994). <https://doi.org/10.1002/andp.19945060203>
45. T. Miyagi, T. Abe, M. Kohno, P. Navrátil, R. Okamoto, T. Otsuka, N. Shimizu, S.R. Stroberg, Phys. Rev. C **100**, 034310 (2019). <https://doi.org/10.1103/PhysRevC.100.034310>. arXiv:1901.11106
46. R.J. Furnstahl, G. Hagen, T. Papenbrock, Phys. Rev. C **86**, 031301 (2012). <https://doi.org/10.1103/PhysRevC.86.031301>. arXiv:1207.6100
47. S.R. Stroberg, H. Hergert, S.K. Bogner, J.D. Holt, Annu. Rev. Nucl. Part. Sci. **69**, 307 (2019). <https://doi.org/10.1146/annurev-nucl-101917-021120>. arXiv:1902.06154
48. E. Epelbaum, A. Nogga, W. Glöckle, H. Kamada, U.-G. Meißner, H. Witała, Phys. Rev. C **66**, 064001 (2002). <https://doi.org/10.1103/PhysRevC.66.064001>
49. S. Pastore, S.C. Pieper, R. Schiavilla, R.B. Wiringa, Phys. Rev. C **87**, 035503 (2013). <https://doi.org/10.1103/PhysRevC.87.035503>. arXiv:1212.3375
50. P. Gysbers, G. Hagen, J.D. Holt, G.R. Jansen, T.D. Morris, P. Navrátil, T. Papenbrock, S. Quaglioni, A. Schwenk, S.R. Stroberg, K.A. Wendt, Nat. Phys. **15**, 428 (2019). <https://doi.org/10.1038/s41567-019-0450-7>. arXiv:1903.00047