Author(s): Dobaczewski, J.; Bączyk, P.; Becker, P.; Bender, M.; Bennaceur, K.; Bonnard, J.; Gao, Y.; Idini, A.; Konieczka, M.; Kortelainen, M.; Próchniak, L.; Romero, A. M.; Satuła, W.; Shi, Y.; Werner, T. R.; Yu, L. F.

Title: Solution of universal nonrelativistic nuclear DFT equations in the Cartesian deformed harmonic-oscillator basis. (IX) HFODD (v3.06h) : a new version of the program

Solution of universal nonrelativistic nuclear DFT equations in the Cartesian deformed harmonic-oscillator basis. (IX) HFODD (v3.06h): a new version of the program

To cite this article: J Dobaczewski *et al* 2021 *J. Phys. G: Nucl. Part. Phys.* **48** 102001

View the article online for updates and enhancements.

You may also like

- Near Zero Volt Tolerant Lithium Ion Cells: Benefits for Safe Storage, Shipping, and Medical Implants
  Kyle R. Crompton and Brian J. Landi

- Scalable bath cell method for reversible lithium management towards near zero volt tolerance in lithium-ion cells
  Kyle Richard Crompton, Michael Hladky, Jason Staub et al.

- Polyhomogeneity and zero-rest-mass fields with applications to Newman-Penrose constants
  Juan Antonio Valiente Kroon

Guide

Solution of universal nonrelativistic nuclear DFT equations in the Cartesian deformed harmonic-oscillator basis. (IX) HFODD (v3.06h): a new version of the program

J Dobaczewski[1,2,3,4,*] , **P Bączyk[3]** , **P Becker[1]** ,
M Bender[5] , **K Bennaceur[5]** , **J Bonnard[1]** , **Y Gao[2]** ,
A Idini[6] , **M Konieczka[3]** , **M Kortelainen[2,4]** ,
L Próchniak[7] , **A M Romero[1,8]** , **W Satuła[3,4]** , **Y Shi[2,9]** ,
T R Werner[3] and L F Yu[2]

[1] Department of Physics, University of York, Heslington, York YO10 5DD, United Kingdom
[2] Department of Physics, PO Box 35 (YFL), FI-40014 University of Jyväskylä, Finland
[3] Institute of Theoretical Physics, Faculty of Physics, University of Warsaw, ul. Pasteura 5, PL-02093 Warsaw, Poland
[4] Helsinki Institute of Physics, PO Box 64, FI-00014 University of Helsinki, Finland
[5] Univ Lyon, Université Claude Bernard Lyon 1, CNRS, IP2I, UMR 5822, 4 rue E Fermi, F-69622 Villeurbanne Cedex, France
[6] Division of Mathematical Physics, Department of Physics, LTH, Lund University, PO Box 118, S-22100 Lund, Sweden
[7] Heavy Ion Laboratory, University of Warsaw, PL-02093 Warsaw, Poland
[8] Department of Physics and Astronomy, University of North Carolina, Chapel Hill, North Carolina, 27516-3255, United States of America
[9] Department of Physics, Harbin Institute of Technology, Harbin 150001, People's Republic of China

E-mail: jacek.dobaczewski@york.ac.uk

CrossMark

Abstract

We describe the new version (v3.06h) of the code HFODD that solves the universal nonrelativistic nuclear DFT Hartree–Fock or

*Author to whom any correspondence should be addressed.

Hartree–Fock–Bogolyubov problem by using the Cartesian deformed harmonic-oscillator basis. In the new version, we implemented the following new features: (i) zero-range three- and four-body central terms, (ii) zero-range three-body gradient terms, (iii) zero-range tensor terms, (iv) zero-range isospin-breaking terms, (v) finite-range higher-order regularized terms, (vi) finite-range separable terms, (vii) zero-range two-body pairing terms, (viii) multi-quasiparticle blocking, (ix) Pfaffian overlaps, (x) particle-number and parity symmetry restoration, (xi) axialization, (xii) Wigner functions, (xiii) choice of the harmonic-oscillator basis, (xiv) fixed omega partitions, (xv) consistency formula between energy and fields, and we corrected several errors in the previous versions.

Keywords: nuclear DFT, harmonic-oscillator basis, HFODD

S Supplementary material for this article is available online

1. Introduction

The method of solving the Hartree–Fock (HF) equations in the Cartesian harmonic oscillator (HO) basis was presented in [1]. Seven versions of the code HFODD were previously published in seven independent publications: (v1.60r) [2], (v1.75r) [3], (v2.08i) [4], (v2.08k) [5], (v2.40h) [6], (v2.49t) [7] and (v2.73y) [8]. Version (v2.08i) [4] introduced solutions of the Hartree–Fock–Bogolyubov (HFB) equations. Below, we refer to these publications by using roman capitals II–VIII. We also acknowledge earlier applications of the Cartesian deformed HO basis to the solution of the nuclear self-consistent problem [9, 10]. The user's guide for version (v2.40h) is available in [11] and the user's guide for the present version (v3.06h) will be published in [12]. The full distribution of the version (v3.06h) of the code HFODD can be found in the supplemental material (https://stacks.iop.org/JPG/48/102001/mmedia). The code home page is at http://fuw.edu.pl/˜dobaczew/hfodd/hfodd.html, and the code repository is at https://webfiles.york.ac.uk/HFODD/. The repository is meant to serve as the first point of contact for users wishing to run the code. It contains the full downloadable distribution of the version (v3.06h) of the code HFODD along with numerous examples of the input data files accompanied with the corresponding output files. In the future, the repository will be dynamically upgraded; it will contain future distributions of the code along with any bugfixes implemented before the next version of HFODD will have been published in a journal. It will also contain descriptions of new features and examples of new input data files.

The present guide is a long write-up of the new version (v3.06h) of the code HFODD. This extended version supersedes all previous versions of the program. It features a number of new implementations listed in section 2. In serial mode, the new version (v3.06h) of the code HFODD remains fully compatible with all previous versions. One should note, however, that in the new version (v3.06h), features of the parallel mode were not thoroughly tested, and the new developments are not recommended for use in parallel mode. In the same way, options related to temperature or fission were not in the main focus of the present developments and should be considered fragile. Otherwise, information provided in previous publications [1–8] remains valid, unless explicitly mentioned in the present long write-up.

The user must have access to various BLAS, LAPACK and LINPACK subroutines, which should be either pre-installed on a given system or downloaded from the Netlib Repository at the University of Tennessee, Knoxville: http://netlib.org/. Otherwise, generic versions of subroutines are also included in the HFODD distribution, available in the supplemental material and from the code repository at https://webfiles.york.ac.uk/HFODD/, and can be compiled along with the main program and its modules.

Version (v3.06h) of the code HFODD is free software. Anyone can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the license, or any later version. Code HFODD is distributed in the hope that it will be useful, but *without any warranty*; without even the *implied warranty of merchantability or fitness* for a particular purpose. See the GNU General Public License at http://gnu.org/licenses/, for more details. The authors will gladly receive any communication regarding the code. However, no dedicated workforce is available for providing user support of any kind.

In section 2, we review the modifications introduced in version (v3.06h) of the code HFODD. Section 3 lists all additional new input keywords and data values, introduced or modified in version (v3.06h). The rules of building the input data file were defined in section II-3 [2] and in the serial mode of version (v3.06h) they remain exactly the same. These rules specify the generic structure of the input data file, irrespective of which specific keywords are used. In particular, the keywords can be read in any order and, unless explicitly stated in their description, they are independent of one another. In every new version of the code HFODD, the list of keywords grows and covers new implementations, but they always abide by the same rules specified in section II-3 [2]. With the new keywords introduced in the present version (v3.06h) of the code HFODD, the list of available keywords already contains 311 items and their descriptions are scattered over nine different publications. The new user's guide [12] will contain comprehensive coverage of the complete information.

2. Modifications introduced in version (v3.06h)

2.1. Zero-range three- and four-body central terms

In his seminal article [13], Skyrme suggested complementing an effective two-body interaction with a contact three-body term. He also underlined the fact that such a three-body interaction, averaged over one of the particles, gives a contribution to the two-body contact term proportional to the local scalar density ρ_0. This observation motivated the use of a two-body contact density-dependent two-body term by Vautherin and Brink [14]. Their interaction was later extended to include the possibility of having different weights for its spin-direct and spin-exchange parts and, possibly, a nonlinear dependence on the density with a power α. It is usually written as,

$$\hat{v}_3(i, j) = \frac{1}{6} t_3 \left(1 + x_3 \hat{P}_{ij}^\sigma\right) \rho_0^\alpha \left(\mathbf{R}_{ij}\right) \delta \left(\mathbf{r}_{ij}\right), \tag{1}$$

where $\mathbf{R}_{ij}$ and $\mathbf{r}_{ij}$ are the center-of-mass and relative positions of the interacting particles. This general density-dependent two-body interaction cannot be related to an underlying three-body force (even in the case where x_3 and α are set to 1) but provides a phenomenological representation of many-body effects [14]. We recall that this term does not completely obey the Pauli exclusion principle and it generates self-interaction terms.

It was pointed out that the latter features might prevent the unambiguous implementation of the multi-reference (MR) extensions of nuclear energy density functionals (EDFs) [15–20].

In addition, non-integer values of α can lead to multivalued energy kernels in MR calculations [17, 21]. Therefore, it is interesting to go back to Skyrme's original idea and to consider a zero-range gradientless three-body interaction. The form implemented in version (v3.06h) of the code HFODD is the one defined in [22, 23],

$$\hat{v}_3(i, j, k) = 3\, u_0\, \delta(\mathbf{r}_{ij})\, \delta(\mathbf{r}_{ik}). \tag{2}$$

In the literature, $t_3 = 3u_0$ is often used for the coupling constant instead [14, 24]. Skyrme's article [13] also proposed a gradientless four-body contact interaction, which in some recent beyond mean-field calculations [25] was used to complement the three-body pseudopotential of equation (2). The form implemented in version (v3.06h) of the code HFODD is again the one defined in [22],

$$\hat{v}_4(i, j, k, l) = 12\, v_0\, \delta(\mathbf{r}_{ij})\, \delta(\mathbf{r}_{ik})\, \delta(\mathbf{r}_{il}). \tag{3}$$

The factors three and twelve in equations (2) and (3), respectively, count the number of different permutations of the coordinates in the delta functions [22]. The contributions from these two terms in the particle–hole and like-particle particle–particle channels of the energy density as detailed in [22] are fully implemented in version (v3.06h) of the code HFODD.

2.2. Zero-range three-body gradient terms

To allow for a greater flexibility of the three-body contribution to the EDF, one can also consider contact interactions with gradients. Terms of this kind have occasionally been considered since the 1970s, see [23] for an overview, but to date have never been used systematically. The most general isospin-invariant central three-body pseudo-potential with two gradients can be written as [23],

$$
\begin{aligned}
\hat{v}(i, j, k) = u_1 \Bigg\{ & \left(1 + y_1 \hat{P}_{ij}^\sigma\right) \frac{1}{2} \left[\hat{\boldsymbol{k}}_{ij}^{\dagger\, 2}\, \delta(\boldsymbol{r}_{ik})\, \delta(\boldsymbol{r}_{jk}) + \delta(\boldsymbol{r}_{ik})\, \delta(\boldsymbol{r}_{jk})\, \hat{\boldsymbol{k}}_{ij}^{2} \right] \\
& + \left(1 + y_1 \hat{P}_{ik}^\sigma\right) \frac{1}{2} \left[\hat{\boldsymbol{k}}_{ik}^{\dagger\, 2}\, \delta(\boldsymbol{r}_{ij})\, \delta(\boldsymbol{r}_{jk}) + \delta(\boldsymbol{r}_{ij})\, \delta(\boldsymbol{r}_{jk})\, \hat{\boldsymbol{k}}_{ik}^{2} \right] \\
& + \left(1 + y_1 \hat{P}_{jk}^\sigma\right) \frac{1}{2} \left[\hat{\boldsymbol{k}}_{jk}^{\dagger\, 2}\, \delta(\boldsymbol{r}_{ij})\, \delta(\boldsymbol{r}_{ik}) + \delta(\boldsymbol{r}_{ij})\, \delta(\boldsymbol{r}_{ik})\, \hat{\boldsymbol{k}}_{jk}^{2} \right] \Bigg\} \\
+ u_2 \Bigg\{ & \left[1 + y_{21}\hat{P}_{ij}^\sigma + y_{22}\left(\hat{P}_{ik}^\sigma + \hat{P}_{jk}^\sigma\right) \right] \hat{\boldsymbol{k}}_{ij}^{\dagger}\, \delta(\boldsymbol{r}_{ik})\, \delta(\boldsymbol{r}_{jk}) \cdot \hat{\boldsymbol{k}}_{ij} \\
& + \left[1 + y_{21}\hat{P}_{ik}^\sigma + y_{22}\left(\hat{P}_{ij}^\sigma + \hat{P}_{jk}^\sigma\right) \right] \hat{\boldsymbol{k}}_{ik}^{\dagger}\, \delta(\boldsymbol{r}_{ij})\, \delta(\boldsymbol{r}_{jk}) \cdot \hat{\boldsymbol{k}}_{ik} \\
& + \left[1 + y_{21}\hat{P}_{jk}^\sigma + y_{22}\left(\hat{P}_{ij}^\sigma + \hat{P}_{ik}^\sigma\right) \right] \hat{\boldsymbol{k}}_{jk}^{\dagger}\, \delta(\boldsymbol{r}_{ij})\, \delta(\boldsymbol{r}_{ik}) \cdot \hat{\boldsymbol{k}}_{jk} \Bigg\},
\end{aligned}
\tag{4}
$$

with the five parameters u_1, y_1, u_2, y_{21} and y_{22}. The EDF resulting from the three-body contact generators can be expressed as an integral over a local energy density that is built out of the same normal and pairing densities as the standard Skyrme EDF. Contributions from the three-body contact generators to the particle–hole and like-particle $T = 1$ particle–particle terms in the EDF, as elaborated in [23], are fully implemented in version (v3.06h) of the code HFODD.

2.3. Zero-range tensor terms

The Skyrme interaction, in the majority of practical implementations, does not include tensor terms. In version (v3.06h) of the code HFODD, we implemented the conventional zero-range

tensor interaction considered already by Skyrme in his seminal work [13], see also [26–31] and references therein:

$$
\begin{aligned}
\hat{v}^{\mathrm{T}}(i, j) = \frac{1}{2} t_e & \left\{ \left(3(\boldsymbol{\sigma}_i \cdot \boldsymbol{k}')(\boldsymbol{\sigma}_j \cdot \boldsymbol{k}') - (\boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j) \boldsymbol{k}'^2 \right) \delta\left(\boldsymbol{r}_{ij}\right) \right. \\
& \left. + \delta\left(\boldsymbol{r}_{ij}\right) \left(3(\boldsymbol{\sigma}_i \cdot \boldsymbol{k})(\boldsymbol{\sigma}_j \cdot \boldsymbol{k}) - (\boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j) \boldsymbol{k}^2 \right) \right\} \\
& + t_o \left\{ \frac{3}{2}(\boldsymbol{\sigma}_i \cdot \boldsymbol{k}') \delta\left(\boldsymbol{r}_{ij}\right) (\boldsymbol{\sigma}_j \cdot \boldsymbol{k}) + \frac{3}{2}(\boldsymbol{\sigma}_j \cdot \boldsymbol{k}') \delta\left(\boldsymbol{r}_{ij}\right) (\boldsymbol{\sigma}_i \cdot \boldsymbol{k}) \right. \\
& \left. - (\boldsymbol{\sigma}_i \cdot \boldsymbol{\sigma}_j) \boldsymbol{k}' \delta\left(\boldsymbol{r}_{ij}\right) \boldsymbol{k} \right\},
\end{aligned}
\tag{5}
$$

where the first term acts in the relative S- and D-waves, whereas the second component acts in the P-wave. Parameters t_e and t_o are new low-energy coupling constants (LECs), which have to be adjusted to experimental data.

The contact tensor interaction contributes to both the time-even (bilinear in time-even densities) and time-odd (bilinear in time-odd densities) terms of the local EDF. The tensor part of the generalized Skyrme EDF in the time-even (t-even) sector is:

$$
\begin{aligned}
\mathcal{H}_t^{\mathrm{T,t-even}}(\boldsymbol{r}) &= C_t^J \mathsf{J}_t^2 + \Delta \mathcal{H}_t^{\mathrm{T,t-even}}(\boldsymbol{r}) \\
&= (C_t^J + B_t^J) \mathsf{J}_t^2 + B_t^X \frac{1}{2} \left\{ (\mathrm{Tr}\ J_t)^2 + \mathrm{Tr}\ J_t^2 \right\},
\end{aligned}
\tag{6}
$$

where $t = 0, 1$ denotes isoscalar and isovector densities, respectively. Within the conventional pn-separable DFT, these are simply sums and differences of the neutron and proton densities, respectively. The standard local spin-current pseudotensor density $J_{t,\mu\nu}(\boldsymbol{r})$ is defined through the nonlocal spin density $s_{t,\nu}(\boldsymbol{r}, \boldsymbol{r}')$ as:

$$
J_{t,\mu\nu}(\boldsymbol{r}) = \frac{1}{2i} \left\{ (\nabla_\mu - \nabla_\mu') s_{t,\nu}(\boldsymbol{r}, \boldsymbol{r}') \right\}_{\boldsymbol{r} = \boldsymbol{r}'}
\tag{7}
$$

with the sum of the squares of its components conventionally denoted as J_t^2,

$$
\mathsf{J}_t^2 \equiv \sum_{\mu\nu} J_{t,\mu\nu}^2.
\tag{8}
$$

The tensor part of the generalized Skyrme EDF in the time-odd (t-odd) sector is:

$$
\begin{aligned}
\mathcal{H}_t^{\mathrm{T,t-odd}}(\boldsymbol{r}) &= C_t^T \boldsymbol{s}_t \cdot \boldsymbol{T}_t + C_t^{\Delta s} \boldsymbol{s}_t \cdot \Delta \boldsymbol{s}_t + \Delta \mathcal{H}_t^{\mathrm{T,t-odd}}(\boldsymbol{r}) \\
&= (C_t^T + B_t^T) \boldsymbol{s}_t \cdot \boldsymbol{T}_t + (C_t^{\Delta s} + B_t^{\Delta s}) \boldsymbol{s}_t \cdot \Delta \boldsymbol{s}_t \\
&\quad + B_t^F \boldsymbol{s}_t \cdot \boldsymbol{F}_t + B_t^{\nabla s} (\nabla \cdot \boldsymbol{s}_t)^2,
\end{aligned}
\tag{9}
$$

where s and T are the standard spin and spin-kinetic densities, respectively, and the tensor-kinetic density [27] F reads,

$$
F_{t,\nu}(\boldsymbol{r}) = \frac{1}{2} \left[\sum_\mu \left\{ (\nabla_\nu \nabla_\mu' + \nabla_\nu' \nabla_\mu) s_{t,\mu}(\boldsymbol{r}, \boldsymbol{r}') \right\} \right]_{\boldsymbol{r} = \boldsymbol{r}'}.
\tag{10}
$$

In equations (6) and (9), C_t^J, C_t^T and $C_t^{\Delta s}$ denote the functional tensor coupling constant due to the central field, as defined in [1, 32]. Tensor interaction (5) adds to these terms its own

contributions, B_t^J, B_t^T and $B_t^{\Delta s}$, respectively, but these additions do not change the structure of the functional. However, the tensor force also adds new terms of the functional specified by terms $\Delta \mathcal{H}_t^{\mathrm{T,t-even}}$ and $\Delta \mathcal{H}_t^{\mathrm{T,t-odd}}$, and by coupling constants B_t^X, B_t^F and $B_t^{\nabla s}$. The new functional coupling constants B_t^X, B_t^F and $B_t^{\nabla s}$ for $t = 0, 1$, are encoded under the names `CSCT_X`, `CKIT_X` and `CSPT_X` for `X = P, M`, respectively, and printed in the code's output.

The twelve new coupling constants B_t^J, B_t^X, B_t^T, $B_t^{\Delta s}$, B_t^F and $B_t^{\nabla s}$ for $t = 0, 1$ relate to the two parameters of the tensor interaction t_e and $3t_o$ as,

$$B_0^J = \frac{1}{8}(t_e + 3t_o), \qquad B_1^J = -\frac{1}{8}(t_e - t_o), \tag{11}$$

$$B_0^{\Delta s} = \frac{3}{32}(t_e - t_o), \qquad B_1^{\Delta s} = -\frac{1}{32}(3t_e + t_o), \qquad B_t^{\nabla s} = 3B_t^{\Delta s} \tag{12}$$

$$B_t^X = -3B_t^J, \qquad B_t^T = -B_t^J, \qquad B_t^F = 3B_t^J. \tag{13}$$

In order to elucidate the role of the last term on the right-hand side of equation (6) it is convenient to decompose the spin-current pseudotensor density (7) into pseudoscalar, vector and rank-2 pseudotensor components [27], which gives equation (8) in the form:

$$\mathsf{J}_t^2 = \frac{1}{3}\left(J_t^{(0)}\right)^2 + \frac{1}{2}\boldsymbol{J}_t^{\,2} + \sum_{\mu\nu}\left(J_{t,\mu\nu}^{(2)}\right)^2. \tag{14}$$

Moreover, since,

$$\frac{1}{2}\left\{\left(\sum_\mu J_{t,\mu\mu}\right)^2 + \sum_{\mu\nu} J_{t,\mu\nu} J_{t,\nu\mu}\right\} = \frac{2}{3}\left(J_t^{(0)}\right)^2 - \frac{1}{4}\boldsymbol{J}_t^{\,2} + \frac{1}{2}\sum_{\mu\nu}\left(J_{t,\mu\nu}^{(2)}\right)^2, \tag{15}$$

the contribution $\Delta \mathcal{H}_t^{\mathrm{T,t-even}}(\boldsymbol{r})$ can be rewritten into the following form (see [27]):

$$\Delta \mathcal{H}_t^{\mathrm{T,t-even}} = B_t^{J_0}\left(J_t^{(0)}\right)^2 + B_t^{J_1}\boldsymbol{J}_t^{\,2} + B_t^{J_2}\left(J_{t,\mu\nu}^{(2)}\right)^2, \tag{16}$$

where,

$$B_t^{J_0} = -\frac{5}{3}B_t^J, \qquad B_t^{J_1} = \frac{5}{4}B_t^J, \quad \text{and} \quad B_t^{J_2} = -\frac{1}{2}B_t^J. \tag{17}$$

This means that the contact zero-range tensor force does not introduce new terms into the time-even part of the Skyrme functional. It only modifies the conventional Skyrme EDF coupling constants $C_t^{J_0} = \frac{1}{3}C_t^J$, $C_t^{J_1} = \frac{1}{2}C_t^J$ and $C_t^{J_2} = C_t^J$ [27]. Hence, within the single-reference DFT the effect of the tensor interaction can be mimicked by readjusting the Skyrme-force values of $C_t^{J_0}$, $C_t^{J_1}$ and $C_t^{J_2}$, which explains why the tensor interaction is often neglected within the standard Skyrme force. In particular, in version (v3.06h) of the code HFODD, separate use of the coupling constants $C_t^{J_0}$, $C_t^{J_1}$ and $C_t^{J_2}$ has not yet been implemented. Nevertheless, a readjustment or, in general, a direct fit of the functional's coupling constants to a dedicated set of empirical data is well within the spirit of the single-reference DFT, which treats EDF as a primary physical object [28, 30, 31]. One should bear in mind, however, that it breaks bonds between the functional and the underlying interaction, which precludes its application within the MR extensions due to singularities that appear in the energy kernels [17–20].

Values of the tensor LECs, t_e and t_o, can be determined through the large-scale multi-parameter fit to masses, see e.g. [28, 30]. An alternative way was proposed in [29]. This method is based on the simultaneous fit of the spin–orbit strength and tensor's LECs to the single-particle levels and spin–orbit splittings in double-magic nuclei ^{40}Ca, ^{48}Ca and ^{56}Ni.

2.4. Zero-range isospin-breaking terms

Isospin symmetry is not a fundamental symmetry of nature. At the fundamental level of quantum chromodynamics, it is broken due to different masses and charges of constituent quarks. At the energy scales typical for nuclear physics, where quarks and gluons are not resolvable and the proper degrees of freedom are point-like nucleons, the isospin-symmetry breaking (ISB) comes, predominantly, from long-range Coulomb interaction and, albeit to a much lesser degree, from short-range effective ISB nuclear forces.

Effective nuclear force can be divided into four different classes following the scheme introduced by Henley and Miller [33, 34]. Apart of the dominant class-I isoscalar force, there are three different classes of ISB forces including class-II isotensor force, class-III isovector force and class IV force, which mixes isospin already at the two-body level. In finite nuclei the ISB effects manifest themselves very clearly already in the simplest observables, the nuclear masses, through mirror displacement energy (MDE):

$$\text{MDE} = BE(T, T_z = -T) - BE(T, T_z = +T), \tag{18}$$

and triplet displacement energies (TDEs):

$$\text{TDE} = BE(T = 1, T_z = -1) + BE(T = 1, T_z = +1) - 2BE(T = 1, T_z = 0). \tag{19}$$

The MDEs and TDEs are almost exclusively sensitive to the charge-symmetry breaking (CSB or class-III) and charge-independence breaking (CIB or class-II) terms in the nuclear Hamiltonian, respectively. Class-IV force will be neglected as no firm evidence of the effects related to this force has been identified so far in many-body data.

It is well known that none of the displacement energies (18) or (19) can be reproduced using models involving Coulomb interaction as the only source of the ISB, see [35–43] and references cited therein. This deficiency concerns, in particular, the nuclear DFT including its most popular realization based on Skyrme forces, which are isoscalar by construction.

In order to account for the MDEs and TDEs, we extended the conventional Skyrme interaction by adding, first, the leading-order (LO) contact interactions of class-II and class-III [41] and, subsequently, generalizing the ISB Skyrme interaction to the next-to-leading (NLO) order in gradient expansion [43]. The introduced ISB terms read:

$$\hat{V}^{\text{II}}(i,j) = t_0^{\text{II}} \left(1 + x_0^{\text{II}} \hat{P}_{ij}^{\sigma}\right) \delta\left(\boldsymbol{r}_{ij}\right) \hat{T}^{(ij)}$$
$$+ \left[\frac{1}{2} t_1^{\text{II}} \left(1 + x_1^{\text{II}} \hat{P}_{ij}^{\sigma}\right) \left(\delta\left(\boldsymbol{r}_{ij}\right) \boldsymbol{k}^2 + \boldsymbol{k}'^2 \delta\left(\boldsymbol{r}_{ij}\right)\right) \right.$$
$$\left. + t_2^{\text{II}} \left(1 + x_2^{\text{II}} \hat{P}_{ij}^{\sigma}\right) \boldsymbol{k}' \delta\left(\boldsymbol{r}_{ij}\right) \boldsymbol{k} \right] \hat{T}^{(ij)}, \tag{20}$$

$$\hat{V}^{\text{III}}(i,j) = t_0^{\text{III}} \left(1 + x_0^{\text{III}} \hat{P}_{ij}^{\sigma}\right) \delta\left(\boldsymbol{r}_{ij}\right) \hat{T}_z^{(ij)}$$
$$+ \left[\frac{1}{2} t_1^{\text{III}} \left(1 + x_1^{\text{III}} \hat{P}_{ij}^{\sigma}\right) \left(\delta\left(\boldsymbol{r}_{ij}\right) \boldsymbol{k}^2 + \boldsymbol{k}'^2 \delta\left(\boldsymbol{r}_{ij}\right)\right) \right.$$
$$\left. + t_2^{\text{III}} \left(1 + x_2^{\text{III}} \hat{P}_{ij}^{\sigma}\right) \boldsymbol{k}' \delta\left(\boldsymbol{r}_{ij}\right) \boldsymbol{k} \right] \hat{T}_z^{(ij)}, \tag{21}$$

where $\hat{P}_{ij}^{\sigma}$ represents the spin-exchange operator, $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$, $\boldsymbol{k} = \frac{1}{2i}\left(\nabla_i - \nabla_j\right)$, and $\boldsymbol{k}' = -\frac{1}{2i}\left(\nabla_i - \nabla_j\right)$ are the standard relative-momentum operators acting to the right and left, respectively, whereas $\hat{T}^{(ij)} = 3\hat{\tau}_3^{(i)}\hat{\tau}_3^{(j)} - \hat{\vec{\tau}}^{(i)} \circ \hat{\vec{\tau}}^{(j)}$ and $\hat{T}_z^{(ij)} = \hat{\tau}_3^{(i)} + \hat{\tau}_3^{(j)}$ are the isotensor and isovector operators. The contributions to EDF from the isovector and isotensor forces read:

$$
\begin{aligned}
\mathcal{H}_{\mathrm{NLO}}^{\mathrm{III}} = {} & \frac{1}{2}t_0^{\mathrm{III}}\left(1 - x_0^{\mathrm{III}}\right)\left(\rho_n^2 - \rho_p^2 - \boldsymbol{s}_n^2 + \boldsymbol{s}_p^2\right) \\
& + \frac{1}{4}t_1^{\mathrm{III}}\left(1 - x_1^{\mathrm{III}}\right)\left(\tau_n\rho_n - \tau_p\rho_p - \boldsymbol{T}_n \cdot \boldsymbol{s}_n + \boldsymbol{T}_p \cdot \boldsymbol{s}_p\right) \\
& + \frac{1}{4}t_2^{\mathrm{III}}\left(1 + x_2^{\mathrm{III}}\right)\left(3\tau_n\rho_n - 3\tau_p\rho_p + \boldsymbol{T}_n \cdot \boldsymbol{s}_n - \boldsymbol{T}_p \cdot \boldsymbol{s}_p\right) \\
& - \frac{3}{16}t_1^{\mathrm{III}}\left(1 - x_1^{\mathrm{III}}\right)\left(\Delta\rho_n\rho_n - \Delta\rho_p\rho_p - \Delta\boldsymbol{s}_n \cdot \boldsymbol{s}_n + \Delta\boldsymbol{s}_p \cdot \boldsymbol{s}_p\right) \\
& + \frac{1}{16}t_2^{\mathrm{III}}\left(1 + x_2^{\mathrm{III}}\right)\left(3\Delta\rho_n\rho_n - 3\Delta\rho_p\rho_p + \Delta\boldsymbol{s}_n \cdot \boldsymbol{s}_n - \Delta\boldsymbol{s}_p \cdot \boldsymbol{s}_p\right) \\
& - \frac{1}{4}t_1^{\mathrm{III}}\left(1 - x_1^{\mathrm{III}}\right)\left(\boldsymbol{j}_n^2 - \boldsymbol{j}_p^2 - \boldsymbol{J}_n^2 + \boldsymbol{J}_p^2\right) \\
& - \frac{1}{4}t_2^{\mathrm{III}}\left(1 + x_2^{\mathrm{III}}\right)\left(3\boldsymbol{j}_n^2 - 3\boldsymbol{j}_p^2 + \boldsymbol{J}_n^2 - \boldsymbol{J}_p^2\right),
\end{aligned}
\tag{22}
$$

$$
\begin{aligned}
\mathcal{H}_{\mathrm{NLO}}^{\mathrm{II}} = {} & \frac{1}{2}t_0^{\mathrm{II}}\left(1 - x_0^{\mathrm{II}}\right)\big(\rho_n^2 + \rho_p^2 - 2\rho_n\rho_p - 2\rho_{np}\rho_{pn} - \boldsymbol{s}_n^2 - \boldsymbol{s}_p^2 + 2\boldsymbol{s}_n \cdot \boldsymbol{s}_p \\
& + 2\boldsymbol{s}_{np} \cdot \boldsymbol{s}_{pn}\big) + \frac{1}{4}t_1^{\mathrm{II}}\left(1 - x_1^{\mathrm{II}}\right)\big(\tau_n\rho_n + \tau_p\rho_p - \tau_n\rho_p - \tau_p\rho_n \\
& - \tau_{np}\rho_{pn} - \tau_{pn}\rho_{np} - \boldsymbol{T}_n \cdot \boldsymbol{s}_n - \boldsymbol{T}_p \cdot \boldsymbol{s}_p + \boldsymbol{T}_n \cdot \boldsymbol{s}_p + \boldsymbol{T}_p \cdot \boldsymbol{s}_n \\
& + \boldsymbol{T}_{np} \cdot \boldsymbol{s}_{pn} + \boldsymbol{T}_{pn} \cdot \boldsymbol{s}_{np}\big) + \frac{1}{4}t_2^{\mathrm{II}}\left(1 + x_2^{\mathrm{II}}\right)\big(3\tau_n\rho_n + 3\tau_p\rho_p - 3\tau_n\rho_p \\
& - 3\tau_p\rho_n - 3\tau_{np}\rho_{pn} - 3\tau_{pn}\rho_{np} + \boldsymbol{T}_n \cdot \boldsymbol{s}_n + \boldsymbol{T}_p \cdot \boldsymbol{s}_p - \boldsymbol{T}_n \cdot \boldsymbol{s}_p \\
& - \boldsymbol{T}_p \cdot \boldsymbol{s}_n - \boldsymbol{T}_{np} \cdot \boldsymbol{s}_{pn} - \boldsymbol{T}_{pn} \cdot \boldsymbol{s}_{np}\big) - \frac{3}{16}t_1^{\mathrm{II}}\left(1 - x_1^{\mathrm{II}}\right)\big(\Delta\rho_n\rho_n \\
& + \Delta\rho_p\rho_p - \Delta\rho_n\rho_p - \Delta\rho_p\rho_n - \Delta\rho_{np}\rho_{pn} - \Delta\rho_{pn}\rho_{np} \\
& - \Delta\boldsymbol{s}_n \cdot \boldsymbol{s}_n - \Delta\boldsymbol{s}_p \cdot \boldsymbol{s}_p + \Delta\boldsymbol{s}_n \cdot \boldsymbol{s}_p + \Delta\boldsymbol{s}_p \cdot \boldsymbol{s}_n + \Delta\boldsymbol{s}_{np} \cdot \boldsymbol{s}_{pn} \\
& + \Delta\boldsymbol{s}_{pn} \cdot \boldsymbol{s}_{np}\big) + \frac{1}{16}t_2^{\mathrm{II}}\left(1 + x_2^{\mathrm{II}}\right)\big(3\Delta\rho_n\rho_n + 3\Delta\rho_p\rho_p - 3\Delta\rho_n\rho_p \\
& - 3\Delta\rho_p\rho_n - 3\Delta\rho_{np}\rho_{pn} - 3\Delta\rho_{pn}\rho_{np} + \Delta\boldsymbol{s}_n \cdot \boldsymbol{s}_n + \Delta\boldsymbol{s}_p \cdot \boldsymbol{s}_p \\
& - \Delta\boldsymbol{s}_n \cdot \boldsymbol{s}_p - \Delta\boldsymbol{s}_p \cdot \boldsymbol{s}_n - \Delta\boldsymbol{s}_{np} \cdot \boldsymbol{s}_{pn} - \Delta\boldsymbol{s}_{pn} \cdot \boldsymbol{s}_{np}\big) - \frac{1}{4}t_1^{\mathrm{II}}\left(1 - x_1^{\mathrm{II}}\right) \\
& \times \big(\boldsymbol{j}_n^2 + \boldsymbol{j}_p^2 - 2\boldsymbol{j}_n \cdot \boldsymbol{j}_p - 2\boldsymbol{j}_{np} \cdot \boldsymbol{j}_{pn} - \boldsymbol{J}_n^2 - \boldsymbol{J}_p^2 + 2\boldsymbol{J}_n \cdot \boldsymbol{J}_p \\
& + 2\boldsymbol{J}_{np} \cdot \boldsymbol{J}_{pn}\big) - \frac{1}{4}t_2^{\mathrm{II}}\left(1 + x_2^{\mathrm{II}}\right)\big(3\boldsymbol{j}_n^2 + 3\boldsymbol{j}_p^2 - 6\boldsymbol{j}_n \cdot \boldsymbol{j}_p - 6\boldsymbol{j}_{np} \cdot \boldsymbol{j}_{pn} \\
& + \boldsymbol{J}_n^2 + \boldsymbol{J}_p^2 - 2\boldsymbol{J}_n \cdot \boldsymbol{J}_p - 2\boldsymbol{J}_{np} \cdot \boldsymbol{J}_{pn}\big),
\end{aligned}
\tag{23}
$$

Table 1. Values of LECs for LO and NLO ISB forces adjusted to the experimental data for the following parameterizations of the (isoscalar) Skyrme forces: SV_T^{ISB}, $SV_{T,SO}^{ISB}$, SkM^{*ISB} and $SLy4^{ISB}$. The t_0 parameter is given in MeV fm^3, whereas the t_1 and t_2 parameters are in MeV fm^5.

Skyrme force	LO approximation			NLO approximation		
	t	Class-II	Class-III	t	Class-II	Class-III
				t_0	-16 ± 3	11 ± 2
				t_1	22 ± 3	-14 ± 4
SV_T	t_0	3.7 ± 0.4	-7.3 ± 0.3	t_2	1 ± 1	-7.8 ± 0.8
				t_0		5 ± 2
				t_1		-3 ± 3
$SV_{T,SO}$	t_0		-6.7 ± 0.3	t_2		-7.4 ± 0.7
SkM^*	t_0	5.2 ± 0.8	-5.4 ± 0.2			
$SLy4$	t_0	5.1 ± 0.8	-5.5 ± 0.2			

where ρ, τ, s, T, j, and J denote the standard particle, kinetic, spin, spin-kinetic, current and vector spin-current densities, respectively. It is important to emphasize that the contributions of class-III local force (22) depend on the standard *nn* and *pp* densities. Therefore, this force can be taken into account within the conventional *pn*-separable DFT approach. In contrast, contributions due to the class-II force (23) depend explicitly on the mixed *pn*-densities and require the use of *pn*-mixed DFT formulated in [44, 45] and implemented in the previous version (v2.73y) of code HFODD [8] together with the isocranking method [44, 46, 47], which allows control of the isospin degree of freedom.

The inclusion of spin-exchange terms in the ISB contact forces (20) and (21) leads to a simple rescaling of the t_i^{II} and t_i^{III} (for $i = 0$, 1 and 2) parameters. Hence, the spin-exchange terms are effectively redundant and can be omitted from the ISB contact forces. Hence, the LO (NLO) ISB forces introduce two (six) new LECs, which must be adjusted to the existing data. Our studies [41, 43] show that the fitting strategy for the new LECs can be considerably simplified for the following reasons. First, for the physically relevant values of the ISB LECs, contributions of the ISB terms to the total binding energies appear to be relatively small (at least this can be concluded from studies performed in the vicinity of the $N = Z$ line). This implies that the ISB terms can be treated as small perturbations to the Skyrme force with frozen parameters. Second, the MDEs and TDEs are almost exclusively sensitive to the CSB and CIB terms in the nuclear Hamiltonian, respectively. As a consequence, the isovector and isotensor LECs can be adjusted separately through the global fit to the experimental MDEs and TDEs, respectively. This strategy was applied to the SLy4 [48] and SkM* [49] forces as well as to the two variants, SV_T and $SV_{T,SO}$ (see [20, 50]), of the SV density-independent Skyrme interaction [24] that can be used in the beyond-mean-field MR DFT calculations. The resulting parameters are collected in table 1. They seem to be consistent with the values given in [42], where a different fitting strategy was used. The adjusted LECs lead to an excellent global description of the existing data on MDEs and TDEs in the isospin doublets and triplets, see [41, 43] for further details.

2.5. Finite-range higher-order regularized terms

Following the notation introduced in [51], see also [52–54], we define the Cartesian form of the (non-antisymmetrized) central pseudopotential as,

$$\mathcal{V}_C(\boldsymbol{r}'_1\boldsymbol{r}'_2;\boldsymbol{r}_1\boldsymbol{r}_2) = \sum_{nj}\left(W_j^{(n)}\hat{1}_\sigma\hat{1}_\tau + B_j^{(n)}\hat{1}_\tau\hat{P}^\sigma - H_j^{(n)}\hat{1}_\sigma\hat{P}^\tau - M_j^{(n)}\hat{P}^\sigma\hat{P}^\tau \right)$$

$$\times \hat{O}_j^{(n)}(\boldsymbol{k}',\boldsymbol{k})\delta(\boldsymbol{r}'_1 - \boldsymbol{r}_1)\delta(\boldsymbol{r}'_2 - \boldsymbol{r}_2)g_a(\boldsymbol{r}_1 - \boldsymbol{r}_2), \tag{24}$$

which contains the standard identity ($\hat{1}_{\sigma,\tau}$) and exchange ($\hat{P}^{\sigma,\tau}$) operators in the spin and isospin spaces and a Gaussian formfactor,

$$g_a(\boldsymbol{r}) = \frac{e^{-r^2/a^2}}{\left(a\sqrt{\pi}\right)^3}, \tag{25}$$

defined by its width a. In equation (24), index $n = 0, 2, \ldots$ denotes the order of differential operator $\hat{O}_j^{(n)}(\boldsymbol{k}',\boldsymbol{k})$, index $j = 1, 2, \ldots$ numbers different operators of the same order, and the relative-momentum operators are defined as $\boldsymbol{k} = (\nabla_1 - \nabla_2)/2i$ and $\boldsymbol{k}' = (\nabla'_1 - \nabla'_2)/2i$. The standard Wigner, Bartlett, Heisenberg, and Majorana coupling constants, $W_j^{(n)}$, $B_j^{(n)}$, $H_j^{(n)}$ and $M_j^{(n)}$, can also be expressed by the strength parameters $t_j^{(n)}$, $x_j^{(n)}$, $y_j^{(n)}$ and $z_j^{(n)}$ as,

$$W_j^{(n)} = t_j^{(n)}, \qquad B_j^{(n)} = t_j^{(n)}x_j^{(n)}, \qquad H_j^{(n)} = t_j^{(n)}y_j^{(n)}, \qquad M_j^{(n)} = t_j^{(n)}z_j^{(n)}. \tag{26}$$

Up to sixth order ($n = 6$), a full classification of operators $\hat{O}_j^{(n)}(\boldsymbol{k}',\boldsymbol{k})$ was presented in equations (39)–(54) of [51]. The EDF generated from the pseudo-potential functional generators for both the particle and pairing channels can be found in [55]. All fields are implemented self–consistently in the present version (v3.06h) of the code HFODD and the computation of the HO-basis spatial matrix elements of pseudopotentials is coded using the following integrated-by-parts form:

$$\langle \boldsymbol{n}'_1\boldsymbol{n}'_2|\hat{O}_j^{(n)}|\boldsymbol{n}_1\boldsymbol{n}_2\rangle \equiv \int d\boldsymbol{r}'_1\, d\boldsymbol{r}'_2\, d\boldsymbol{r}_1\, d\boldsymbol{r}_2\, \delta(\boldsymbol{r}'_1 - \boldsymbol{r}_1)\delta(\boldsymbol{r}'_2 - \boldsymbol{r}_2)g_a(\boldsymbol{r}_1 - \boldsymbol{r}_2)$$

$$\times \hat{O}_j^{(n)}(\boldsymbol{k}',\boldsymbol{k})\psi_{\boldsymbol{n}'_1}(\boldsymbol{r}'_1)\psi_{\boldsymbol{n}'_2}(\boldsymbol{r}'_2)\psi_{\boldsymbol{n}_1}(\boldsymbol{r}_1)\psi_{\boldsymbol{n}_2}(\boldsymbol{r}_2), \tag{27}$$

where $\psi_{\boldsymbol{n}}(\boldsymbol{r}) \equiv \psi_{n_x n_y n_z}(x, y, z)$ are the 3D deformed HO wave functions, equations (I-71)–(I-73) [1].

Using the explicit expressions [51] for differential operators $\hat{O}_j^{(n)}(\boldsymbol{k}',\boldsymbol{k})$, one can tediously but straightforwardly rewrite them as sums of terms that are products of differential operators acting in the x, y and z directions,

$$\hat{O}_j^{(n)}(\boldsymbol{k}',\boldsymbol{k}) = \sum_{\boldsymbol{m},\boldsymbol{l}} K_{j;\boldsymbol{l}}^{(n;\boldsymbol{m})}\hat{O}_{l_x}^{(m_x)}(k'_x,k_x)\hat{O}_{l_y}^{(m_y)}(k'_y,k_y)\hat{O}_{l_z}^{(m_z)}(k'_z,k_z), \tag{28}$$

where $K_{j;\boldsymbol{l}}^{(n;\boldsymbol{m})} \equiv K_{j;l_x,l_y,l_z}^{(n;m_x,m_y,m_z)}$ for $m_x + m_y + m_z = n$ are integer coefficients and 1D differential operators $\hat{O}_l^{(m)}(k',k)$ of order $m = 0, 2, \ldots$ and index $l = 0, 1, \ldots, m/2$ are given by,

$$\hat{O}_l^{(m)}(k', k) \equiv \left((k'^{\,2} + k^2)/2 \right)^l \left(k'k \right)^{m/2 - l}$$

$$= 2^{-l} \sum_{l'=0}^{l} \binom{l}{l'} k'^{\,m/2 + 2l' - l} k^{m/2 - 2l' + l}. \tag{29}$$

The matrix element given in equation (27) is thus equal to the sum of the products of 1D matrix elements $\langle n_1' n_2' | \hat{O}_l^{(m)} | n_1 n_2 \rangle$,

$$\langle \boldsymbol{n}_1' \boldsymbol{n}_2' | \hat{O}_j^{(n)} | \boldsymbol{n}_1 \boldsymbol{n}_2 \rangle = \sum_{\boldsymbol{m}, \boldsymbol{l}} K_{j;\boldsymbol{l}}^{(n;\boldsymbol{m})} \langle n_{x1}' n_{x2}' | \hat{O}_{l_x}^{(m_x)} | n_{x1} n_{x2} \rangle$$

$$\times \langle n_{y1}' n_{y2}' | \hat{O}_{l_y}^{(m_y)} | n_{y1} n_{y2} \rangle \langle n_{z1}' n_{z2}' | \hat{O}_{l_z}^{(m_z)} | n_{z1} n_{z2} \rangle, \tag{30}$$

where,

$$\langle n_1' n_2' | \hat{O}_l^{(m)} | n_1 n_2 \rangle = \int dr_1' \, dr_2' \, dr_1 \, dr_2 \, \delta(r_1' - r_1) \delta(r_2' - r_2) g_a(r_1 - r_2)$$

$$\times \hat{O}_l^{(m)}(k', k) \psi_{n_1'}(r_1') \psi_{n_2'}(r_2') \psi_{n_1}(r_1) \psi_{n_2}(r_2), \tag{31}$$

and where the 1D Gaussian form factor (25) reads $g_a(r) = e^{-r^2/a^2} / \left(a\sqrt{\pi} \right)$.

The standard way to proceed, which for the local LO term $\hat{O}_1^{(0)} = 1$ was developed in [56], is to replace the products of HO wave functions $\psi_{n_1'}(r_1) \psi_{n_1}(r_1)$ and $\psi_{n_2'}(r_2) \psi_{n_2}(r_2)$ by sums of the HO wave functions. Then, one can use the Moshinsky brackets to introduce the relative coordinate $r_1 - r_2$ on which the Gaussian form factor depends. At higher orders, this approach requires the explicit treatment of terms that stem from expanding powers of relative-momentum operators that appear in equation (29). Although this tedious procedure was implemented up to fourth order in version (v3.06h) of the code HFODD, an alternative and more compact procedure is to reverse the order of steps and begin by performing two Moshinsky transformations,

$$\psi_{n_1'}(r_1') \psi_{n_2'}(r_2') = \sum_{N'=0}^{n_1' + n_2'} M_{N'}^{n_1' n_2'} \psi_{N'}(R') \psi_{n_1' + n_2' - N'}(r'), \tag{32}$$

$$\psi_{n_1}(r_1) \psi_{n_2}(r_2) = \sum_{N=0}^{n_1 + n_2} M_{N}^{n_1 n_2} \psi_{N}(R) \psi_{n_1 + n_2 - N}(r), \tag{33}$$

for $R' = \frac{r_1' + r_2'}{\sqrt{2}}$, $r' = \frac{r_1' - r_2'}{\sqrt{2}}$, $R = \frac{r_1 + r_2}{\sqrt{2}}$ and $r = \frac{r_1 - r_2}{\sqrt{2}}$. Since in equation (29), the relative-momentum operators $k = \frac{-i}{\sqrt{2}} \frac{\partial}{\partial r}$ and $k' = \frac{-i}{\sqrt{2}} \frac{\partial}{\partial r'}$ act only on wave functions $\psi_{n_1 + n_2 - N}(r)$ and $\psi_{n_1' + n_2' - N'}(r')$, respectively, the integrals over R' and r' can be performed, which gives,

$$\langle n_1' n_2' | \hat{O}_l^{(m)} | n_1 n_2 \rangle = \frac{(-1)^{m/2}}{2^{l+m/2}} \sum_{N'=0}^{n_1' + n_2'} M_{N'}^{n_1' n_2'} \sum_{N=0}^{n_1 + n_2} M_{N}^{n_1 n_2} \left[\int dR \, \psi_{N'}(R) \psi_{N}(R) \right]$$

$$\times \left[\sum_{l'=0}^{l} \binom{l}{l'} \int dr \, \psi_{n_1' + n_2' - N'}^{(m/2 + 2l' - l)}(r) \, \psi_{n_1 + n_2 - N}^{(m/2 - 2l' + l)}(r) g_a \left(\sqrt{2} r \right) \right], \tag{34}$$

where superscripts denote derivatives of wave functions: $\psi_n^{(i)}(r) = \frac{d^i}{dr^i}\psi_n(r)$. The orthogonality of wave functions $\psi_N(R)$ allows for the presentation of the final result as,

$$\langle n_1' n_2' | \hat{O}_l^{(m)} | n_1 n_2 \rangle = \frac{(-1)^{m/2}}{2^{l+m/2}} \sum_{N=0}^{\min(n_1'+n_2', n_1+n_2)} M_N^{n_1' n_2'} M_N^{n_1 n_2}$$

$$\times \sum_{l'=0}^{l} \binom{l}{l'} C_{n_1'+n_2'-N, n_1+n_2-N}^{(m/2+2l'-l, m/2-2l'+l)}(a), \tag{35}$$

where,

$$C_{n',n}^{(i',i)}(a) = \int dr\, \psi_{n'}^{(i')}(r) \psi_n^{(i)}(r) g_a\left(\sqrt{2}r\right). \tag{36}$$

Similarly, as was shown in [56], relatively simple analytic expressions can be derived for coefficients $C_{n',n}^{(i',i)}(a)$. However, these expressions involve alternating-sign sums of ratios of large factorials and are thus prone to generating significant numerical instabilities [57, 58]. Here, we argue that using these analytical expressions in practical implementations is not necessary. Indeed, a very simple and extremely stable numerical derivation based on Gauss–Hermite quadratures is possible; it was already implemented in the previous version (v2.73y) of HFODD [8] to treat the Gogny force.

In the context of the higher-order finite-range functional generators (24) discussed here, the numerical implementation works as follows. First, we represent derivatives of the HO wave functions (I-72) as,

$$\psi_n^{(i)}(r) = \frac{d^i}{dr^i}\psi_n(r) \equiv b^{i+1/2} H_n^{(i)}(\xi) e^{-\xi^2/2}, \tag{37}$$

where $\xi = br$ is the position r scaled by the oscillator constant $b = \sqrt{m\omega/\hbar}$, and $H_n^{(i)}(\xi)$ are polynomials of order $n + i$, which can be easily derived from the standard Hermite polynomials $H_n^{(0)}(\xi)$ and their first derivatives $H_n^{(0)'}(\xi)$, e.g.:

$$H_n^{(1)}(\xi) = H_n^{(0)'}(\xi) - \xi H_n^{(0)}(\xi), \tag{38}$$

$$H_n^{(2)}(\xi) = \left(\xi^2 - 2n - 1\right) H_n^{(0)}(\xi), \tag{39}$$

$$H_n^{(3)}(\xi) = \left(\xi^2 - 2n - 1\right) H_n^{(0)'}(\xi) - \left(\xi^3 - (2n+3)\xi\right) H_n^{(0)}(\xi), \tag{40}$$

$$H_n^{(4)}(\xi) = 4\xi H_n^{(0)'}(\xi) + \left(\xi^4 - (4n+6)\xi^2 + 4n^2 + 4n + 3\right) H_n^{(0)}(\xi). \tag{41}$$

This allows one to represent equation (36) as,

$$C_{n',n}^{(i',i)}(a) = \frac{b^{i'+i}}{a\sqrt{\pi}} \int d\xi\, H_{n'}^{(i')}(\xi) H_n^{(i)}(\xi) e^{-\frac{2+b^2a^2}{b^2a^2}\xi^2}$$

$$= \frac{b^{i'+i+1}}{\sqrt{\pi}\sqrt{2+b^2a^2}} \int d\eta\, H_{n'}^{(i')}(\epsilon\eta) H_n^{(i)}(\epsilon\eta) e^{-\eta^2}$$

$$= \frac{b^{i'+i+1}}{\sqrt{\pi}\sqrt{2+b^2a^2}} \sum_{k=1}^{K} W_k H_{n'}^{(i')}(\epsilon\eta_k) H_n^{(i)}(\epsilon\eta_k), \tag{42}$$

where,

$$\epsilon = \frac{ba}{\sqrt{2 + b^2 a^2}}, \tag{43}$$

and W_k and η_k are, respectively, weights and nodes of the Gauss–Hermite quadrature of order $K = n' + n + i' + j + 1$. For calculations employing the HO basis of up to N_0 quanta in the given Cartesian direction x, y, or z, and for derivatives up to fourth order, the quadrature of order $K = 2N_0 + 5$ thus gives the exact result, and no accumulation of numerical errors is expected.

Exactly the same method can be used to evaluate the Moshinsky coefficients, which in their exact analytical form (VI-63) [6] also involve numerically unstable alternating-sign sums of ratios of large factorials. Indeed, by setting in equation (33) $r_1 = r_2 \equiv r$ and inserting equation (37) for $i = 0$, we obtain:

$$H_{n_1}^{(0)}(\xi) H_{n_2}^{(0)}(\xi) = \sum_{N=0}^{n_1+n_2} M_N^{n_1 n_2} H_N^{(0)}(\sqrt{2}\xi) H_{n_1+n_2-N}^{(0)}(0). \tag{44}$$

We now can multiply both sides by $\sqrt{2} H_{N'}^{(0)}(\sqrt{2}\xi) e^{-2\xi^2}$, integrate over ξ, and use the orthogonality condition of the Hermite polynomials on the right-hand side. This finally gives,

$$\int d\eta \, H_{n_1}^{(0)}\left(\frac{\eta}{\sqrt{2}}\right) H_{n_2}^{(0)}\left(\frac{\eta}{\sqrt{2}}\right) H_N^{(0)}(\eta) e^{-\eta^2} = M_N^{n_1 n_2} H_{n_1+n_2-N}^{(0)}(0), \tag{45}$$

where $\eta = \sqrt{2}\xi$. This allows one to determine the exact Moshinsky coefficients through a numerically stable Gauss–Hermite quadrature of order $K = n_1 + n_2 + N + 1$,

$$M_N^{n_1 n_2} = \left(H_{n_1+n_2-N}^{(0)}(0)\right)^{-1} \sum_{k=1}^{K} W_K H_{n_1}^{(0)}\left(\frac{\eta_k}{\sqrt{2}}\right) H_{n_2}^{(0)}\left(\frac{\eta_k}{\sqrt{2}}\right) H_N^{(0)}(\eta_k). \tag{46}$$

Therefore, equations (42) and (46) give an exact and numerically stable representation of the 1D matrix elements (35) of higher-order generators. Furthermore, coefficients $M_N^{n_1 n_2}$ and $C_{n',n}^{(i',i)}(a)$ have to be calculated only once and if needed, stored.

In the special case of local generators discussed in [52], the central pseudopotential (24) reduces for $W^{(n)} \equiv W_1^{(n)} - W_2^{(n)}, B^{(n)} \equiv B_1^{(n)} - B_2^{(n)}, H^{(n)} \equiv H_1^{(n)} - H_2^{(n)}$ and $M^{(n)} \equiv M_1^{(n)} - M_2^{(n)}$ to,

$$\mathcal{V}_C^{\text{loc}}\left(\boldsymbol{r}_1'\boldsymbol{r}_2'; \boldsymbol{r}_1\boldsymbol{r}_2\right) = \sum_n \left(W^{(n)}\hat{1}_\sigma \hat{1}_\tau + B^{(n)}\hat{1}_\tau \hat{P}^\sigma - H^{(n)}\hat{1}_\sigma \hat{P}^\tau \right.$$

$$\left. - M^{(n)}\hat{P}^\sigma \hat{P}^\tau \right) \delta(\boldsymbol{r}_1' - \boldsymbol{r}_1)\delta(\boldsymbol{r}_2' - \boldsymbol{r}_2)V^{(n)}(\boldsymbol{r}_1 - \boldsymbol{r}_2), \tag{47}$$

where,

$$V^{(n)}(\boldsymbol{r}) \equiv 2^{-n}\Delta^{n/2} g_a(\boldsymbol{r}), \tag{48}$$

and where Δ is the standard differential Laplace operator. Explicitly, this gives:

$$V_n^{(0)}(\boldsymbol{r}) = g_a(\boldsymbol{r}), \tag{49}$$

$$V_n^{(1)}(\boldsymbol{r}) = \frac{1}{a^2}\left(2\left(\frac{r^2}{a^2}\right) - 3\right) g_a(\boldsymbol{r}), \tag{50}$$

$$V_n^{(2)}(\boldsymbol{r}) = \frac{1}{a^4}\left(4\left(\frac{\boldsymbol{r}^4}{a^4}\right) - 20\left(\frac{\boldsymbol{r}^2}{a^2}\right) + 15\right)g_a(\boldsymbol{r}), \tag{51}$$

$$V_n^{(3)}(\boldsymbol{r}) = \frac{1}{a^6}\left(8\left(\frac{\boldsymbol{r}^6}{a^6}\right) - 84\left(\frac{\boldsymbol{r}^4}{a^4}\right) + 210\left(\frac{\boldsymbol{r}^2}{a^2}\right) - 105\right)g_a(\boldsymbol{r}). \tag{52}$$

Similarly, as for the nonlocal operators above in equation (28), we can now rewrite the potentials (48) as sums of terms that are products of powers of positions x, y, and z. That is,

$$V^{(n)}(\boldsymbol{r}) = \sum_{\boldsymbol{m}} K^{(n;\boldsymbol{m})} x^{m_x} y^{m_y} z^{m_z}, \tag{53}$$

where $K^{(n;\boldsymbol{m})} \equiv K^{(n;m_x,m_y,m_z)}$ for $0 \leqslant m_x + m_y + m_z \leqslant n$ are integer coefficients. The matrix element of two-body potential $V^{(n)}(\boldsymbol{r}_1 - \boldsymbol{r}_2)$ is thus equal to the sum of the products of 1D matrix elements,

$$\langle \boldsymbol{n}_1' \boldsymbol{n}_2' | V^{(n)} | \boldsymbol{n}_1 \boldsymbol{n}_2 \rangle = \frac{1}{\left(a\sqrt{\pi}\right)^3} \sum_{\boldsymbol{m}} K^{(n;\boldsymbol{m})} \langle n_{x_1}' n_{x_2}' | (x_1 - x_2)^{m_x} e^{-(x_1-x_2)^2/a^2}$$

$$\times |n_{x_1} n_{x_2}\rangle \langle n_{y_1}' n_{y_2}' | (y_1 - y_2)^{m_y} e^{-(y_1-y_2)^2/a^2} | n_{y_1} n_{y_2}\rangle$$

$$\times \langle n_{z_1}' n_{z_2}' | (z_1 - z_2)^{m_z} e^{-(z_1-z_2)^2/a^2} | n_{z_1} n_{z_2}\rangle, \tag{54}$$

where,

$$\langle n_1' n_2' | (r_1 - r_2)^m e^{-(r_1-r_2)^2/a^2} | n_1 n_2 \rangle = \int dr_1 \, dr_2 \, (r_1 - r_2)^m e^{-(r_1-r_2)^2/a^2}$$

$$\times \psi_{n_1'}(r_1)\psi_{n_2'}(r_2)\psi_{n_1}(r_1)\psi_{n_2}(r_2)$$

$$= \sum_{N=0}^{\bar{N}} M_N^{n_1' n_2'} M_N^{n_1 n_2} C_{n_1'+n_2'-N, n_1+n_2-N}^{(m)}(a), \tag{55}$$

and where $\bar{N} = \min(n_1' + n_2', n_1 + n_2)$ and,

$$C_{n',n}^{(m)}(a) = \frac{b^{1-m}}{\sqrt{\pi}\sqrt{2 + b^2 a^2}} \sum_{k=1}^{K} W_k H_{n'}^{(0)}(\epsilon \eta_k) H_n^{(0)}(\epsilon \eta_k) \times (\epsilon \eta_k)^m, \tag{56}$$

for ϵ given in equation (43) and W_k and η_k being, respectively, weights and nodes of the Gauss–Hermite quadrature of order $K = n' + n + m + 1$.

2.6. Finite-range separable terms

The separable pairing force in the isovector $^1S_0$ channel, introduced by Tian *et al* in the spherical case [59] and by Nikšić *et al* in the 3D deformed case [60, 61], is implemented in version (v3.06h) of the code HFODD. The general expression of this interaction in the 3D Cartesian coordinates is,

$$\mathcal{V}_S(\boldsymbol{r}_1'\boldsymbol{r}_2';\boldsymbol{r}_1\boldsymbol{r}_2) = \left(\tilde{W}\hat{1}_\sigma\hat{1}_\tau + \tilde{B}\hat{1}_\tau\hat{P}^\sigma - \tilde{H}\hat{1}_\sigma\hat{P}^\tau - \tilde{M}\hat{P}^\sigma\hat{P}^\tau \right)$$
$$\times\ \delta(\boldsymbol{R}' - \boldsymbol{R})P(\boldsymbol{r}')P(\boldsymbol{r}), \tag{57}$$

where formfactor $P(\boldsymbol{r})$ is equal to a sum of Gaussians (25),

$$P(\boldsymbol{r}) = \sum_k^K A_k g_{a_k}(\boldsymbol{r}), \tag{58}$$

and $\boldsymbol{r}' = \boldsymbol{r}_1' - \boldsymbol{r}_2'$, $\boldsymbol{r} = \boldsymbol{r}_1 - \boldsymbol{r}_2$, $\boldsymbol{R}' = \frac{1}{2}(\boldsymbol{r}_1' + \boldsymbol{r}_2')$ and $\boldsymbol{R} = \frac{1}{2}(\boldsymbol{r}_1 + \boldsymbol{r}_2)$ are the relative and center-of-mass coordinates. To avoid redundancy with coupling constants $\tilde{W}$, $\tilde{B}$, $\tilde{H}$ and $\tilde{M}$, one should use normalization $\sum_k A_k = 1$.

A detailed derivation of the matrix elements of the separable generators in zero order can be found in [62] and references therein. This implementation was compared with an updated version of the code HOSPHE [63], where the separable interaction was implemented in spherical symmetry. The figure of [64] was reproduced up to a precision of 1 eV, therefore confirming the accuracy of our implementation.

2.7. Zero-range two-body pairing terms

In version (v3.06h) of the code HFODD, all terms of the pairing functional [27] that correspond to the Skyrme functional were implemented.

2.8. Multi-quasiparticle blocking

Quasiparticle blocking was initially introduced in version (v2.40h) of the code HFODD (see VI) to allow for the description of odd-*A* or odd–odd paired nuclei. It consists of looking for a solution of the HFB equations with as ansatz a vacuum $|\Phi\rangle$ onto which a single-quasiparticle excitation $\beta_k^\dagger$ is applied, $|\Phi_k\rangle = \beta_k^\dagger|\Phi\rangle$. Specifically, at each iteration, the code selects the quasiparticle state k in the matrix φ and exchanges its upper (B^*) and lower (A^*) components with those (A, B) of its partner of opposite quasiparticle energy, see equations (VI-83)–(VI-86) [6]. According to the option requested by the user, see keywords `BLOCKFIX_N` or `BLOCKFIX_P` in section VI-3.3 [6], blocked quasiparticle k may be kept the same throughout the calculation or selected at each iteration as the one having the maximum overlap with a single-particle state (or its time-reversed image) chosen beforehand.

In version (v3.06h) of the code HFODD, this method was extended to HFB states $|\Phi_{\boldsymbol{k}}\rangle$ with an arbitrary number r of quasiparticle excitations $\boldsymbol{k} \equiv (k_1,\ldots,k_r)$:

$$|\Phi_{\boldsymbol{k}}\rangle = \prod_{\nu=1}^{r} \beta_{k_\nu}^\dagger|\Phi\rangle, \tag{59}$$

where $|\Phi\rangle$ is the HFB vacuum for the quasiparticle operators β_k, $k = 1,\ldots,M$. The wave function in equation (59) is represented by the $2M \times M$ matrices $(\varphi_{\boldsymbol{k}}, \chi_{\boldsymbol{k}})$ obtained by swapping the components of the blocked quasiparticles in the solutions of the HFB solution (φ, χ). For instance, the wave functions associated with negative energies may schematically be written as,

$$\varphi_{\boldsymbol{k}} = \begin{pmatrix} B_1^* & \cdots & A_{k_1} & B_{k_1+1}^* & \cdots & A_{k_r} & \cdots & B_M^* \\ A_1^* & \cdots & B_{k_1} & A_{k_1+1}^* & \cdots & B_{k_r} & \cdots & A_M^* \end{pmatrix}. \tag{60}$$

Numerically, the quasiparticles k_ν, $\nu = 1 \rightarrow r$, defined via successive applications of the procedure described above, whilst ensuring a given label, can be selected only once.

2.9. Pfaffian overlaps

Computation of the overlap kernels between HFB wave functions is of crucial importance in MR calculations and for symmetry restoration. To date, these scalar products were most often evaluated using the Onishi formula [65], which suffers from a sign ambiguity due to a square-root appearing there. This limitation was then overcome by Robledo via a new expression involving a Pfaffian [66]. Version (v3.06h) of the code HFODD includes a new module based on an equivalent Pfaffian formulation derived in [67], which allows for determining the overlap between two arbitrary, potentially blocked HFB states of the general form (59) as,

$$
\langle \Phi_{\boldsymbol{k}} | \Phi'_{\boldsymbol{k}'} \rangle = (-1)^{M(M-1)/2}(-1)^{r(r-1)/2}\mathrm{pf}
\begin{pmatrix}
B^T A & B^T p^\dagger & B^T q'^T & B^T B'^* \\
-p^* B & q^* p^\dagger & q^* q'^T & q^* B'^* \\
-q' B & -q' q^\dagger & p' q'^T & p' B'^* \\
-B'^\dagger B & -B'^\dagger q^\dagger & -B'^\dagger p'^T & A'^\dagger B'^*
\end{pmatrix}.
\tag{61}
$$

This relation holds for non-normalized wave functions. Square matrices A and B (A' and B') denote the usual blocks of the Bogolyubov transformation, which correspond to the non-blocked HFB state $|\Phi\rangle$ ($|\Phi'\rangle$), whereas rectangular matrices p and q (p' and q') contain components of the r (r') blocked quasiparticle states, see [67]. In the case of single-quasiparticle blocking, the latter matrices reduce to row vectors, whereas for non-blocked HFB wave functions, the corresponding rows and columns do not appear in matrix (61).

Equation (61) is valid only for a complete quasiparticle space. Consequently, the Pfaffian formula cannot be used when a cut-off in the space of quasiparticle states is implemented, see section IV-3.1 [4]. Therefore, to use the Pfaffian formula, the pairing cut-off must be handled within the two-basis method, see section VII-2.2.1 [7]. Moreover, since the Pfaffian formula is based on associating the non-blocked (even) HFB state with the product of all quasiparticle annihilation operators acting on the true vacuum, $|\Phi\rangle \propto \prod_i \beta_i |0\rangle$, the number of quasiparticles must be even. This implies that the dimension of the single-particle space generated by the two-basis method must be even.

2.10. Particle-number and parity symmetry restoration

The HFB method accounts for pairing correlations through the breaking of the U(1) symmetry associated with particle-number conservation. The above computation of overlaps allows us to implement the restoration of correct proton and neutron numbers by projection after variation of a symmetry-unrestricted HFB state $|\Phi\rangle$ [68]. In version (v3.06h) of the code HFODD, projections of the total particle number A and isospin projection (IP) $T_z = (N - Z)/2$ were implemented by introducing two new independent keywords. Activating only one of those projections thus allows for a full particle-number-symmetry restoration for nuclei where one of the species, protons or neutrons, is unpaired in $|\Phi\rangle$. In future releases, this implementation will be optimized by considering a 1D gauge-angle integration that allows for the simultaneous restoration of both proton- and neutron-number symmetries, according to the methodology presented in [15] and routinely used in other implementations, see, e.g. [69].

Version (v3.06h) of the code HFODD also incorporates parity restoration by means of the projector $\hat{P}_\pi = (1 + \pi \hat{\Pi})/2$ where $\pi = \pm$ and $\hat{\Pi}$ is the inversion transformation. Finally, a state with good quantum numbers A, T_z, I^π, M, and K is obtained as,

$$
\begin{aligned}
|AT_z; I^\pi MK\rangle &= \hat{P}_A \hat{P}_{T_z} \hat{P}^I_{MK} \hat{P}_\pi |\Phi\rangle \\
&= \frac{2I+1}{16\pi^4} \int_0^\pi \mathrm{d}\phi \, \mathrm{e}^{-\mathrm{i}\phi A} \int_0^\pi \mathrm{d}\phi_T \mathrm{e}^{-\mathrm{i}\phi_T \, T_z} \int \mathrm{d}\Omega \, D^{I^*}_{MK}(\Omega) \mathrm{e}^{\mathrm{i}\phi \hat{A}} \, \mathrm{e}^{\mathrm{i}\phi_T \hat{T}_z} \hat{R}(\Omega)(1 + \pi \hat{\Pi})|\Phi\rangle.
\end{aligned}
$$

(62)

The operations detailed in section VI-2.1 [6] for the angular-momentum projection (AMP) of Slater determinants were generalized to the HFB states. Quasiparticles $\tilde{\varphi}^T$, transformed by generic symmetry operators $\hat{T}$ that appear in equation (62), read:

$$
\tilde{\varphi}^T = \begin{pmatrix} T & 0 \\ 0 & T^* \end{pmatrix} \begin{pmatrix} B^* \\ A^* \end{pmatrix} = \begin{pmatrix} TB^* \\ T^*A^* \end{pmatrix}, \quad \text{for } \tilde{\varphi} = \begin{pmatrix} B^* \\ A^* \end{pmatrix},
$$

(63)

characterizing state $|\Phi\rangle$ and T denoting the representation of $\hat{T}$ in the single-particle basis. Then, kernels of observables are computed according to the generalized Wick's theorem [70] in terms of the transition normal and pairing densities, and the overlap kernels are evaluated according to equation (61).

In version (v3.06h) of the code HFODD, particle-number projection is realized by using Gauss–Tchebyschev quadratures, whereas the discrete parity projector is applied explicitly. The numerical treatment of the integration over the Euler angles Ω was described in section VI-2.1 [6].

2.11. Axialization

In version (v3.06h) of the code HFODD, axial self-consistent solutions can be obtained by projecting wave functions on the axial shape, with the symmetry axis oriented along the z axis. This is achieved by projecting the particle-hole or pairing mean-field and/or particle-hole or pairing density matrix (pairing tensor) on those corresponding to the axial symmetry. Specifically, this is achieved by expanding the Cartesian harmonic-oscillator basis used by the code on states having good quantum numbers Ω_k, which are the eigenvalues of the z component of the single-particle angular momentum. Then, at each iteration, only the matrix elements of the particle-hole matrices that are diagonal in Ω_k are kept and/or only the off-diagonal $(\Omega_k, -\Omega_k)$ matrix elements of the pairing matrices are kept. At convergence, an axial state is obtained with all single-particle or quasiparticle states having good quantum numbers Ω_k. The axialization helps to stabilize the convergence of states, which at self-consistency are axial, but during the convergence can wander towards non-axial deformations and thus converge slowly or sometimes not at all.

2.12. Wigner functions

To perform the angular-momentum and isospin-projection calculations, previously the code HFODD used the Wigner formula to compute the Wigner d functions, $d^j_{m,n}(\theta)$. For $j \geqslant 50$, the Wigner formula is known to suffer from a loss of precision [71], which is due to the fact that with $j \gg 1$ and $\theta \neq 0, \pi$, it relies on the cancellation of very large terms with alternating signs.

In [71], a robust procedure of computing the d functions was proposed. In this method, the d functions were expended using the Fourier series. In [72], another method was proposed, which was based on the diagonalization of the angular-momentum operator J_y in the basis of eigenstates of J_z. Version (v3.06h) of the code HFODD, after implementing a few corrections, uses the code published in [72].

2.13. Choice of the harmonic-oscillator basis

To fix the HO basis used in the program (see section II-4 [2]), one needs to choose suitable oscillator constants b_k or, equivalently, oscillator frequencies ω_k or oscillator lengths, L_k, in three Cartesian directions,

$$b_k = \sqrt{\frac{m\omega_k}{\hbar}}, \quad L_k = \sqrt{\frac{\hbar}{m\omega_k}}, \quad \text{for } k = x, y, z. \tag{64}$$

There are many possible ways to determine b_k or ω_k. The methods implemented in version (v3.06h) of the code HFODD are described below. They correspond to different values of the variable INPOME set by using new functionalities of keyword FREQBASIS, see section 3.9, which are described below.

(a) For the default value INPOME $= 0$, the code uses values of the basis-deformation input parameters $\alpha_{2\mu}$ to define surface Σ:

$$\Sigma : R(\theta, \phi) = c(\alpha) \left(1 + \sum_{\lambda\mu} \alpha_{\lambda\mu} Y^*_{\lambda\mu}(\theta, \phi) \right), \tag{65}$$

and then it determines the mean squared values of positions r_k^2 over the interior of Σ:

$$R_k^2 = \int_{r < R(\theta,\phi)} r_k^2 \, dV, \quad k = x, y, z. \tag{66}$$

Conditions,

$$\omega_x R_x = \omega_y R_y = \omega_z R_z, \tag{67}$$

and,

$$\omega_x \omega_y \omega_z = \omega_0^3, \tag{68}$$

are then used to determine ω_k, with ω_0 calculated according to equation (I-3) [1]. Parameter $c(\alpha)$ is fixed by the condition that the volume V_0 inside Σ is equal to,

$$V_0 = \frac{4}{3}\pi R_0^3, \tag{69}$$

where $R_0 = r_0 A^{1/3}$, and r_0 is given by variable R0PARM read under keyword SURFACE_PAR, see section II-3.6 [2]. This prescription works well assuming that $\alpha_{\lambda\mu}$ are real and $\alpha_{21} = 0$, which means, among others, that surface (65) is in the principal-axes frame of the quadrupole deformation.

(b) For INPOME $= 1$, the oscillator frequencies ω_k are given explicitly as input parameters of the program; $\omega_x =$ BASINX, $\omega_y =$ BASINY, $\omega_z =$ BASINZ.

(c) For INPOME $= 2$, the oscillator lengths L_k are given explicitly as input parameters of the program; $L_x =$ BASINX, $L_y =$ BASINY, $L_z =$ BASINZ.

(d) For INPOME $= 3$, the oscillator constants b_k are given explicitly as input parameters of the program; $b_x =$ BASINX, $b_y =$ BASINY, $b_z =$ BASINZ.

(e) For INPOME = 4, to calculate the oscillator lengths L_k the code uses the value of the basis-deformation input parameter α_{20},

$$L_x = L_0 \, \exp(-\sqrt{\frac{5}{16\pi}}\alpha_{20}), \tag{70}$$

$$L_y = L_0 \, \exp(-\sqrt{\frac{5}{16\pi}}\alpha_{20}), \tag{71}$$

$$L_z = L_0 \, \exp(\sqrt{\frac{5}{4\pi}}\alpha_{20}), \tag{72}$$

in analogy to equation (1.88) in [70]. This prescription generates an axial basis. Here, the code uses $L_0 = \sqrt{2^*20.735\,53/\hbar\omega_0}$ for $\hbar\omega_0 = 1.2 * 41 * A^{-1/3}$.

(f) For INPOME = 5, the code uses values of mass quadrupole constraints, $\bar{Q}_{20}$ and $\bar{Q}_{22}$ (in barn), see keyword MULTCONSTR in section II-3.7 [2], to calculate β and γ deformation parameters as,

$$\beta = C\sqrt{\bar{Q}_{20}^2 + \bar{Q}_{22}^2}, \qquad \gamma = \text{atan}\,(\bar{Q}_{22}/\bar{Q}_{20}), \tag{73}$$

where,

$$C = 10^2 \frac{\sqrt{5\pi}}{3AR_0^2}, \qquad R_0 = r_0 A^{1/3}, \tag{74}$$

and r_0 (in fm) is given by variable R0PARM read under keyword SURFAC_PAR, see section II-3.6 [2]. Then, to fix frequencies ω_k, the code employs conditions (67) and (68) with,

$$R_k = R_0 \left(1 + \sqrt{\frac{5}{4\pi}}\beta \, \cos(\gamma - 2k\pi/3) \right). \tag{75}$$

(g) For INPOME = 6, the code uses values of the basis-deformation input parameters α_{20} and α_{22} to calculate oscillator frequencies ω_k as,

$$\omega_k = \omega_0 \exp\left(-\sqrt{\frac{5}{4\pi}}\beta \, \cos(\gamma - 2k\pi/3) \right), \tag{76}$$

where,

$$\beta = \sqrt{\alpha_{20}^2 + 2\alpha_{22}^2}, \qquad \gamma = \text{atan}\,(\sqrt{2}\alpha_{22}/\alpha_{20}), \tag{77}$$

and ω_0 is calculated according to equation (I-3) [1].

(h) For INPOME = 7, the code uses values of mass multipole constraints, $\bar{Q}_{\lambda\mu}$ (in barn $\lambda/2$), see keyword MULTCONSTR in section II-3.7 [2], to calculate Bohr deformations $\alpha_{\lambda\mu}$ according to the method presented in section VI-2.5 [6], see also keyword BOHR_BETAS in section VI-3.5 [6]. Values of $\alpha_{\lambda\mu}$ are then used in the algorithm developed for INPOME = 0, see point (a) above.

Options INPOME = 5 and 7 were developed to automatically adjust the HO basis to the quadrupole constraints requested in, e.g. fission-barrier calculations. However, a new functionality of keyword MULTCONSTR, see section 3.9, allows for reading values of $\bar{Q}_{\lambda\mu}$ irrespective of whether they are used as constraints.

Options `INPOME = 0` and 4–7 ignore values of input data `BASINX`, `BASINY` and `BASINZ`.

Options `INPOME = 0`, 4 and 6 use values of the basis-deformation input parameters $\alpha_{2\mu}$ read under keyword `SURFAC_DEF`, see section II-3.6 [2]. However, for `IBCONT = 1`, see keyword `CONT_BASIS`, values read from the basis file override those read under keyword `SURFAC_DEF`.

Note also that in the parallel mode of the code HFODD, the basis deformation can be automatically adjusted by setting `IBASIS = 1` under keyword `BASISAUTOM`, see section VIII-3.1.4 [8].

2.14. Fixed Ω partitions

In version (v3.06h) of the code HFODD, without pairing and for broken simplex symmetry, arbitrary partitions of particles among different Ω blocks were implemented, where Ω denotes the eigenvalue of a given Cartesian component of the single-particle angular momentum on the axial-symmetry axis. To this end, every single-particle state with the calculated projection of the angular momentum equal to Ω_i is attributed to a given Ω block if $\Omega - \frac{1}{2} \leqslant \Omega_i < \Omega + \frac{1}{2}$. Although this attribution can always be performed, it can serve its purpose only if the single-particle states are eigenstates of the given Cartesian component of the angular momentum, that is, their alignments are properly quantized. This requires that (i) the nucleus has an axial shape and (ii) the Kramers degeneracy is lifted by aligning individual angular momenta along the symmetry axis. The first requirement can be fulfilled by constraining the non-axial quadrupole deformation to zero, see section II-3.7 [2], or better, by using the axialization option described in sections 2.11 and 3.4. The second requirement can be fulfilled by using a small value ($\approx$1 keV) of the cranking frequency along the symmetry axis. Indeed, when an unpaired nucleus has axial symmetry with the symmetry axis aligned with the given Cartesian direction, cranking along that axis does not change the single-particle wave functions, but only splits the corresponding single-particle energies as required. A soft attribution condition specified above allows for a correct convergence to an axial state even if during the convergence one or both requirements (i) and (ii) are only approximately fulfilled.

2.15. Consistency formula between energy and fields

Many authors of Hartree–Fock solvers have implemented a consistency formula, which allows one to check, by summing over the energies of the occupied single-particle states, whether the total energy and mean field are consistent in their implementation (see for example [73, 74]). Such a formula was also used to define the stability energy employed in the code HFODD as a criterion to terminate iterations, see equation (I-37) [1]. Here, we show that the consistency formula can be extended to the case of the HFB calculations and to the energy density, which contains linear, bilinear, trilinear, quadrilinear or possibly higher couplings of densities. Note that this energy density is not necessarily derived from an interaction; it is sufficient that it contains products of densities that are contractions of n creation and n annihilation operators evaluated in an HFB state $|\Phi\rangle$.

Up to $n = 4$, the total energy of a nucleus can be split as,

$$E = E_1 + E_2 + E_3 + E_4, \tag{78}$$

with,

$$E_1 = \sum_{ij} v_{ij}^{(1)} \rho_{ji}, \tag{79}$$

$$E_2 = \sum_{ijkl} \left(v^{(2)}_{ij,kl} \, \rho_{ki}\rho_{lj} + \tilde{v}^{(2)}_{ij,kl} \, \kappa^*_{ij}\kappa_{kl} \right), \tag{80}$$

$$E_3 = \sum_{ijklmn} \left(v^{(3)}_{ijk,lmn} \, \rho_{li}\rho_{mj}\rho_{nk} + \tilde{v}^{(3)}_{ijk,lmn} \, \kappa^*_{ij}\kappa_{lm}\rho_{nk} \right), \tag{81}$$

$$E_4 = \sum_{ijklmnop} \left(v^{(4)}_{ijkl,mnop} \, \rho_{mi}\rho_{nj}\rho_{ok}\rho_{pl} + \tilde{v}^{(4)}_{ijkl,mnop} \, \kappa^*_{ij}\kappa_{mn}\rho_{ok}\rho_{pl} \right.$$
$$\left. + \tilde{\tilde{v}}^{(4)}_{ijkl,mnop} \, \kappa^*_{ij}\kappa^*_{kl}\kappa_{mn}\kappa_{op} \right), \tag{82}$$

where $v^{(1)}$ is the one-body kinetic operator and $v^{(2)}$, $\tilde{v}^{(2)}$, $v^{(3)}$, $\tilde{v}^{(3)}$, $v^{(4)}$, $\tilde{v}^{(4)}$ and $\tilde{\tilde{v}}^{(4)}$ are two-, three- and four-body scalar Hermitian matrix elements, which fulfill the same usual properties under the exchange of indices as the matrix elements of interactions, and the standard density matrix and pairing tensor of state $|\Phi\rangle$ are given by $\rho_{ij} = \langle\Phi|a^\dagger_j a_i|\Phi\rangle$ and $\kappa_{ij} = \langle\Phi|a_j a_i|\Phi\rangle$.

For simplicity, the consistency formula is here derived assuming one species of nucleons only, that is, only one chemical potential λ. We also do not consider the possibility that the n-body matrix elements depend on the one-body density (as is the case when they are derived from a density-dependent interaction or when the Slater approximation is used for the Coulomb-exchange term of the energy). The generalization for these cases is straightforward.

From the energy (78), one obtains the normal field:

$$h_{ij} = v^{(1)}_{ij} + \sum_{kl} 2\, v^{(2)}_{ik,jl} \, \rho_{lk} + \sum_{klmn} \left(3\, v^{(3)}_{ikl,jmn} \, \rho_{mk}\rho_{nl} + \tilde{v}^{(3)}_{lki,mnj} \, \kappa^*_{kl}\kappa_{mn} \right)$$
$$+ \sum_{klmnop} \left(4\, v^{(4)}_{iklm,jnop} \, \rho_{nk}\rho_{ol}\rho_{pm} + 2\, \tilde{v}^{(4)}_{lkmi,nopj} \, \kappa^*_{lk}\kappa_{no}\rho_{pm} \right), \tag{83}$$

and the pairing field:

$$\tilde{h}_{ij} = \sum_{kl} 2\, \tilde{v}^{(2)}_{ij,kl} \, \kappa^*_{lk} + \sum_{klmn} 2\, \tilde{v}^{(3)}_{kln,jim} \, \kappa^*_{kl}\rho_{mn}$$
$$+ \sum_{klmnop} \left(2\, \tilde{v}^{(4)}_{klmn,jiop} \, \kappa^*_{kl}\rho_{om}\rho_{pn} + 4\, \tilde{\tilde{v}}^{(4)}_{klmn,opji} \, \kappa^*_{kl}\kappa^*_{mn}\kappa_{op} \right). \tag{84}$$

Assuming the HFB equations have been solved, the quasiparticle wave-function spinors,

$$\Psi_j = \begin{pmatrix} U_j \\ V_j \end{pmatrix}, \tag{85}$$

fulfill the equations:

$$\sum_j h_{ij}U_j + \tilde{h}_{ij}V_j = (E_i + \lambda)\, U_i, \tag{86}$$

$$\sum_j \tilde{h}^*_{ij}U_j - h^*_{ij}V_j = (E_i - \lambda)\, V_i, \tag{87}$$

where E_i are the (positive) quasiparticle energies. Multiplying the second equation by V_i^* and summing over i, one obtains the consistency formula:

$$E_1 + 2E_2 + 3E_3 + 4E_4 = \sum_i V_i^2 (\lambda - E_i). \tag{88}$$

This allows one to define the HFB stability energy,

$$\delta\mathcal{E}_{\text{HFB}} = \sum_i V_i^2 (\lambda - E_i) - (E_1 + 2E_2 + 3E_3 + 4E_4), \tag{89}$$

which can be used as a measure of deviation of state $|\Phi\rangle$ from the self-consistent solution.

2.16. Corrected errors

In version (v3.06h) of the code HFODD, we corrected a few little significant errors and two significant errors, sections 2.16.1 and 2.16.2, found in the previous versions of HFODD.

2.16.1. Incorrect signs of the Yukawa energies. In the published versions (v2.08i) [4], (v2.08k) [5], (v2.40h) [6], (v2.49t) [7] and (v2.73y) [8] of the code HFODD, the signs of the Yukawa energies were inverted. This error was corrected in the results published in [75].

2.16.2. Incorrect off-diagonal generator coordinate method (GCM) kernels. Between versions (v2.10a) and (v2.99u), calculations of the off-diagonal GCM kernels were incorrect. The error manifested itself only when the single-particle wave functions were not real, and was present in the published versions (v2.40h) [6], (v2.49t) [7] and (v2.73y) [8].

2.16.3. Definition of the Schiff moment. Between versions (v2.19n) and (v2.80m), the factor of 1/10 usually included in the definition of the standard Schiff moment, see equation (2) in [75], was missing from the values printed on the output file. This inconsistent definition was implemented in the published versions (v2.40h) [6], (v2.49t) [7] and (v2.73y) [8].

2.16.4. Time-odd symmetries in AMP. Before version (v2.66b), for conserved time-odd symmetries (ISIMTX = 1, or ISIMTY = 1 or ISIMTZ = 1, see section IV-3.2 [4]), the AMP was allowed and might give inconsistent results. This error was thus present in the published version (v2.49t) [7] of HFODD and corrected in the published version (v2.73y) [8]. However, in the latter publication it was not described.

2.16.5. Very large harmonic-oscillator bases. As it turns out, for very large harmonic-oscillator bases of NOSCIL > 36, see section II-3.6 [2], the code may behave erratically. Therefore, beginning with version (v2.81b), calculations with NOSCIL > 36 are not allowed. This issue awaits future debugging.

2.16.6. Inconsistent input data in angular-momentum and isospin projection. When keywords PROJECTGCM (section VI-3.2 [6]) and PROJECTISO (section VII-3.1 [7]) were simultaneously used in the input data file, the type of calculation performed could depend on the order in which they were used. This contradicted the rules of building the input data file defined in section II-3 [2]. Moreover, for IPRROT = 0 (see section 3.9), the remaining input data read under keyword PROJECTGCM were not ignored, which could trigger the AMP against the user's intentions.

In version (v3.06h) of the code HFODD, variables

- IPRROT (keyword PROJECTGCM)

- `IPRISO` (keyword `PROJECTISO`)
- `IPRNUM` (keyword `PROJPARNUM`)
- `IPRVEC` (keyword `PROJVECNUM`)
- `IPRPTY` (keyword `PROJPARITY`)

must be synchronized, that is, their non-zero values must all be equal to one another. Internally, they are replaced by the single variable `IPRGCM`. For any of these variables equal to 0 (not equal to 0), the remaining input data read under the corresponding keyword are ignored (used for defining the corresponding projection).

2.16.7. Inconsistent input data in tilted angular momentum. When keywords `OMEGA_XYZ` and `OMEGA_RTP` (section IV-3.5 [4]) were simultaneously used in the input data file, the type of calculation performed could depend on the order in which they were used. This contradicted the rules of building the input data file defined in section II-3 [2]. In version (v3.06h) of the code HFODD, the simultaneous use of these two keywords is no longer allowed.

2.16.8. Incorrect information stored on the kernel file. For runs without isospin-symmetry restoration, not all kernels were stored on the kernel file, see section VI-3.2 [6]; nevertheless those not stored were later used in the printouts. This was causing differences between results printed in the runs where the kernels were calculated and those where the kernels were read from the kernel file.

2.16.9. Incorrect information stored on the RECORD file. After version (v1.78), Fermi energies, pairing gaps and Lipkin–Nogami parameters were incorrectly stored on the RECORD file. Consequently, the smooth continuation of runs with pairing could have been impeded. The error had no effect on the final converged results. It was present in all published versions of HFODD after the pairing was introduced in version (v2.08i) [4].

2.16.10. Incorrect description of keyword `FILSIG_NEU`. In version (v2.40h) [6], the description of keyword `FILSIG_NEU` was incorrect. It should have referred to *twice* the number of particles, that is, it should have read: 'matrices `KOFILG` contain twice the number of particles put into the states between `KHFILG` and `KPFILG`, by using partial occupation factors of `KOFILG/(KPFILG-KHFILG + 1)/2`' for them.

3. Input data file

The rules of building the input data file were defined in section II-3 [2]; in version (v3.06h) of the code HFODD these rules remain exactly the same. All previous items (keywords) of the input data file remain valid, and several new ones were added, as described in sections 3.1–3.7. For some previous items, new features or new values of variables were added (section 3.9).

For every keyword listed below, we give the default values and names of the variables read. Apart from character variables, which must start at the 13th column of the input line, all other variables are read in FORTRAN free format. Nevertheless, it is good practice to include in the input file the integer or real constants when reading the INTEGER TYPE [IMPLICIT INTEGER (I-N)] or REAL TYPE [IMPLICIT REAL (A-H, O-Z)] variables, respectively.

3.1. Interaction

3.1.1. Zero-range central terms.
   Keyword: `2BODYDELTA`
   $0., 0 = $ `TWOINP, ITWOIN`

For $\texttt{ITWOIN} = 1$, the value of a two-body zero-range parameter $\texttt{TWOINP}$ is added to the Skyrme parameter t_0. This option is introduced only for convenience of handling the input data in cases when a two-body zero-range interaction is handled independently of the Skyrme force. For $\texttt{ITWOIN} = 0$, the value of $\texttt{TWOINP}$ is ignored.

Keyword: $\texttt{3BODYDELTA}$

$0., 0 = \texttt{THRINP, ITHRIN}$

For $\texttt{ITHRIN} = 1$, the value of $\texttt{THRINP}$ defines the three-body zero-range parameter u_0, equation (2). For $\texttt{ITHRIN} = 0$, the value of $\texttt{THRINP}$ is ignored and the three-body zero-range force is not taken into account.

Keyword: $\texttt{4BODYDELTA}$

$0., 0 = \texttt{FOUINP, IFOUIN}$

For $\texttt{IFOUIN} = 1$, the value of $\texttt{FOUINP}$ defines the four-body zero-range parameter v_0, equation (3). For $\texttt{IFOUIN} = 0$, the value of $\texttt{FOUINP}$ is ignored and the four-body zero-range force is not taken into account.

Keyword: $\texttt{SKYRMEINPU}$

$0., 0., 0., 0., 0., 0., 0., 0., 0., 1. =$	$\texttt{T0\_DAT, X0\_DAT,}$
	$\texttt{T1\_DAT, X1\_DAT,}$
	$\texttt{T2\_DAT, X2\_DAT,}$
	$\texttt{T3\_DAT, X3\_DAT,}$
	$\texttt{WW\_DAT, PO\_DAT}$

Keyword: $\texttt{SKYRME\_ERR}$

$0., 0., 0., 0., 0., 0., 0., 0., 0., 1. =$	$\texttt{T0\_ERR, X0\_ERR,}$
	$\texttt{T1\_ERR, X1\_ERR,}$
	$\texttt{T2\_ERR, X2\_ERR,}$
	$\texttt{T3\_ERR, X3\_ERR,}$
	$\texttt{WW\_ERR, PO\_ERR}$

Keyword: $\texttt{SKYRME\_FAC}$

$0., 0., 0., 0., 0., 0., 0., 0., 0., 1. =$	$\texttt{T0\_FAC, X0\_FAC,}$
	$\texttt{T1\_FAC, X1\_FAC,}$
	$\texttt{T2\_FAC, X2\_FAC,}$
	$\texttt{T3\_FAC, X3\_FAC,}$
	$\texttt{WW\_FAC, PO\_FAC}$

For the Skyrme-force acronym, section IV-3.1 [4]), $\texttt{SKYRME} = \texttt{INPU}$, values of the ten input parameters above correspond to the standard Skyrme parameters, t_0, x_0, t_1, x_1, t_2, x_2, t_3, x_3, W_0 and α, where α is the power of density in the density-dependent term. Each parameter is determined as, e.g. $t_0 = \texttt{T0\_DAT} + \texttt{T0\_ERR} * \texttt{T0\_FAC}$. The formula allows for a systematic modification of the central value $\texttt{T0\_DAT}$, shifted by a step $\texttt{T0\_ERR}$ multiplied by a factor $\texttt{T0\_FAC}$. This is useful when building the Jacobian matrix [76] of derivatives of observables over the Skyrme parameters.

Keyword: $\texttt{SKYRMEPAIR}$

$0 = \texttt{KETAPA}$

For KETAPA $= 1$ or 2, the pairing terms of the Skyrme functional [27, 77] are taken into account. However, for KETAPA $= 2$, the pairing terms generated by the spin–orbit force W_0 are neglected. For KETAPA $= 0$, all pairing terms of the Skyrme functional are neglected. KETAPA $= 1$ allows for a fully self-consistent pairing calculation performed for the SKYRME $=$ SKP Skyrme parameters [77], but of course it can also be used for any other variant of the Skyrme force. KETAPA > 0 requires NOZEPA $= 0$. In version (v3.06h) of the code HFODD, KETAPA > 0 still requires ISIMPY $= 0$, ISIQTY $= 0$, IPNMIX $= 0$, IFTEMP $= 0$ and KETA_T $= 0$.

3.1.2. Zero-range three-body gradient terms.
Keyword: 3BODYGRAD

0., 0., 0., 0., 0., 0 = TGRA10, TGRA11, TGRA20, TGRA21, TGRA22, IGRAIN

For IGRAIN $= 1$, the parameters of the three-body gradient force (4) are defined as $u_1 =$ TGRA10, $y_1 =$ TGRA11, $u_2 =$ TGRA20, $y_{21} =$ TGRA21 and $y_{22} =$ TGRA22. For IGRAIN $= 0$, the values of TGRA10, TGRA11, TGRA20, TGRA21 and TGRA22 are ignored and the three-body gradient force is not taken into account. In a given run of the code HFODD, keyword 3BODYGRAD must not be simultaneously used with keyword 3BODYGRUY.

Keyword: 3BODYGRUY

0., 0., 0., 0., 0., 0 = TGRA10, TU1_Y1, TGRA20, TU2Y21, TU2Y22, IGRAIN

For IGRAIN $= 1$, parameters of the three-body gradient force (4) are defined as $u_1 =$ TGRA10, $u_1 y_1 =$ TU1_Y1, $u_2 =$ TGRA20, $u_2 y_{21} =$ TU2Y21, and $u_2 y_{22} =$ TU2Y22. For IGRAIN $= 0$, the values of TGRA10, TU1_Y1, TGRA20, TU2Y21, and TU2Y22 are ignored and the three-body gradient force is not taken into account. For keyword 3BODYGRUY, values of TGRA10 $= 0$ or TGRA20 $= 0$ are not allowed. In a given run of the code HFODD, keyword 3BODYGRUY must not be simultaneously used with keyword 3BODYGRAD.

3.1.3. Zero-range tensor terms.
Keyword: SKYRMETENS

0., 0., 0 = TEINPU, TOINPU, KETA_T

For KETA_T $= 2$, parameters of the zero-range tensor force, equation (5), are defined as $t_e =$ TEINPU, $t_o =$ TOINPU. For KETA_T $= 1$, the values of t_e and t_o correspond to those pre-defined for a given Skyrme force selected by its acronym, see section IV-3.1 [4]. For KETA_T $= 0$, the values of TEINPU and TOINPU, are ignored and the tensor force is not taken into account. In version (v3.06h) of the code HFODD, KETA_T> 0 still requires IPNMIX $= 0$ and KETAPA $= 0$.

Keyword: TEN_ADD_PM

0., 0., 0., 0., 0., 0., =	ASCT_P, ASCT_M,
	AKIT_P, AKIT_M,
	ASPT_P, ASPT_M

By using keyword TEN_ADD_PM, tensor coupling B_t^X, B_t^F and $B_t^{\nabla s}$ for $t = 0, 1$, see section 2.3, can be shifted by adding the values of ASCT_X, AKIT_X, and ASPT_X for X $=$ P, M, respectively.

Keyword: TEN_ADD_TS

The same as for keyword TEN_ADD_PM but for the tensor coupling constants in the total-sum representation, see equations (I-14)–(I-15) [1] and section II-3.2 [2].

Keyword: TEN_SCA_PM

0., 0., 0., 0., 0., 0., =	ASCT_T, ASCT_S,
	AKIT_T, AKIT_S,
	ASPT_T, ASPT_S

0., 0., 0., 0., 0., 0., =	SSCT_P, SSCT_M,
	SKIT_P, SKIT_M,
	SSPT_P, SSPT_M

By using keyword TEN_SCA_PM, tensor coupling B_t^X, B_t^F and $B_t^{\nabla s}$ for $t = 0, 1$, see section 2.3, can be scaled by multiplying them with the values of SSCT_X, SKIT_X and SSPT_X for X = P, M, respectively.

Keyword: TEN_SCA_TS

0., 0., 0., 0., 0., 0., =	SSCT_T, SSCT_S,
	SKIT_T, SKIT_S,
	SSPT_T, SSPT_S

The same as for keyword TEN_SCA_PM but for the tensor coupling constants in the total-sum representation, see equations (I-14)–(I-15) [1] and section II-3.2 [2].

3.1.4. Zero-range isospin-breaking terms.   **Keyword:** CBR_CC_CL2

0, 0., 0., 0., 0., 0., 0. = I_2CBR,	T02CBR, X02CBR,
	T12CBR, X12CBR,
	T22CBR, X22CBR

For I_2CBR = 1, class-II ISB terms are included in the calculation with parameters: t_0^{II} = T02CBR, x_0^{II} = X02CBR, t_1^{II} = T12CBR, x_1^{II} = X12CBR, t_2^{II} = T22CBR, x_2^{II} = X22CBR, see equation (20). Note, that the interaction of class II requires p–n mixing (IPNMIX = 1). In version (v3.06h) of the code HFODD, I_2CBR = 1 still requires IPRGCM = 0.

Keyword: CBR_CC_CL3

0, 0., 0., 0., 0., 0., 0. = I_3CBR,	T03CBR, X03CBR,
	T13CBR, X13CBR,
	T23CBR, X23CBR

For I_3CBR = 1, class-III ISB terms are included in the calculation with parameters: t_0^{III} = T03CBR, x_0^{III} = X03CBR, t_1^{III} = T13CBR, x_1^{III} = X13CBR, t_2^{III} = T23CBR, x_2^{III} = X23CBR, see equation (21). In version (v3.06h) of the code HFODD, I_3CBR = 1 still requires IPRGCM = 0.

3.1.5. Higher-order regularized terms.

Keyword: `REGULFORCE`

$0 = $ `I_REGA`

For `I_REGA` > 0, the average mean-field energies of the finite-range regularized central pseudopotentials (24) or (47) are calculated. For `I_REGA` $= 2$ or 3, the corresponding direct mean fields are included in the self-consistent mean field. For `I_REGA` $= 2$ or 4, the corresponding exchange mean fields are included in the self-consistent mean field. Altogether, `I_REGA` $= 1$ demands calculations of contributions to energy only, whereas `I_REGA` $= 2$ demands full self-consistent calculations with both direct and exchange mean fields included. For `I_REGA` $= 0$, the finite-range regularized central pseudopotential is ignored.

For `I_REGA` > 0 and `IPAHFB` > 1, the code issues a warning to the effect that, unless the zero-range pairing strengths are explicitly set to zero, see section IV-3.1 [4], the corresponding pairing will still be active. For `NOZEPA` $= 1$, see section 3.8, the zero-range pairing is neglected, and the warning is not printed. In version (v3.06h) of the code HFODD, `I_REGA` > 0 still requires `IPNMIX` $= 0$, `IRENMA` $= 0$ and `IBROYD` $= 0$.

Keyword: `REGUL_PAIR`

$0 = $ `IREGPA`

For `IREGPA` > 0, the average pairing energies of the finite-range regularized central pseudopotential (24) or (47) are calculated. For `IREGPA` $= 2$, the corresponding pairing fields are included in the self-consistent pairing field. Altogether, `IREGPA` $= 1$ demands calculations of contributions to the pairing energy, whereas `I_REGA` $= 2$ demands full self-consistent calculations with pairing fields included. For `I_REGA` $= 0$, the pairing contribution of the finite-range regularized central pseudopotential is ignored. `IREGPA` > 0 requires `I_REGA` > 0 and `IPAHFB` > 0. In version (v3.06h) of the code HFODD, `IREGPA` > 0 still requires `IPNMIX` $= 0$, `IRENMA` $= 0$ and `IBROYD` $= 0$.

Keyword: `REGUCOUPLI`

100, 0, 1.	$=$	`IREREJ(1), NREREJ(1), REGWID`
0., 0., 0., 0.	$=$	`REJVCC(1, 1), REJVCC(1, 2), REJVCC(1, 3), REJVCC(1, 4)`
0., 0., 0., 0.	$=$	`REJVCC(2, 1), REJVCC(2, 2), REJVCC(2, 3), REJVCC(2, 4)`
	....	
0., 0., 0., 0.	$=$	`REJVCC(`i`, 1), REJVCC(`i`, 2), REJVCC(`i`, 3), REJVCC(`i`, 4)`

After reading the first line, the code reads $i = $ `IREREJ(1)` lines with

- Four coupling constants $W_j^{(n)}$, $B_j^{(n)}$, $H_j^{(n)}$ and $M_j^{(n)}$ per line (for `NREREJ(1)` < 0) and uses pseudopotential (24).
- Four coupling constants $W^{(n)}$, $B^{(n)}$, $H^{(n)}$ and $M^{(n)}$ per line (for `NREREJ(1)` > 0), and uses local pseudopotential (47).

$N = 2 *$ |`NREREJ(1)`| denotes the order of expansion (the maximum value of n) and $a = $ `REGWID` denotes the width of the Gaussian formfactor $g_a(\boldsymbol{r})$, see section 2.5.

For `NREREJ(1)` < 0, the code reads the coupling constants corresponding to the terms defined in equations (42)–(54) of [51], with the exception of coupling constants corresponding to the terms that depend on the $\hat{T}_3$ operator, see equation (41) of [51]. The latter terms were not yet implemented. Consecutive lines of input are numbered by index $k = 1, \ldots, i$ and

correspond to the coupling constants defined by indices n and j as:

$$
\begin{array}{ccc|ccc|ccc|ccc}
k & n & j & k & n & j & k & n & j & k & n & j \\
\hline
1 & 0 & 1 & 2 & 2 & 1 & 4 & 4 & 1 & 7 & 6 & 1 \\
  &   &   & 3 & 2 & 2 & 5 & 4 & 2 & 8 & 6 & 2 \\
  &   &   &   &   &   & 6 & 4 & 3 & 9 & 6 & 3 \\
  &   &   &   &   &   &   &   &   & 10 & 6 & 4
\end{array}
\qquad (90)
$$

The number of lines read i must be consistent with the order of expansion, that is, for NREREJ(1) < 0 or NREREJ(1) $\geqslant 0$, there must be exactly $i = (N+2)(N+4)/8$ or $i = (N+2)/2$ lines read, respectively.

For IREREJ(1) $= 0$, no lines with parameters are read, and the code uses the local higher-order pseudopotential (47) with coupling constants $W^{(n)}$, $B^{(n)}$, $H^{(n)}$ and $M^{(n)}$ derived from the Gogny interaction up to order N. The methodology and equations developed in [52] are then used. For IREREJ(1) $= 0$, NREREJ(1) < 0 is not allowed.

In a given run of the code HFODD,

- Keyword REGUCOUPLI must not be simultaneously used with keyword REGUL_TXYZ.
- All read values of array IREREJ must be the same and are internally used as variable IREREG.
- All read values of array NREREJ must be the same and are internally used as variable N3LORD.

Internally, the code uses array REGVCC $=$ REJVCC. To inform the code on whether the coupling constants had been read from the input data file, all elements of array IREREJ are predefined to 100.

In version (v3.06h) of the code HFODD, calculations with nonlocal pseudopotential (24) or local pseudopotential (47) are implemented up to order $N = 4$ (N2LO) or $N = 6$ (N3LO), respectively. Consequently, the only allowed values are $-2 \leqslant$ N3LORD $\leqslant 3$.

Keyword: REGCOUPERR

```
100, 0,            =    IREREJ(2), NREREJ(2)
0., 0., 0., 0.     =    REJERR(1, 1), REJERR(1, 2), REJERR(1, 3), REJERR(1, 4)
0., 0., 0., 0.     =    REJERR(2, 1), REJERR(2, 2), REJERR(2, 3), REJERR(2, 4)

           ................................................................
0., 0., 0., 0.     =    REJERR(i, 1), REJERR(i, 2), REJERR(i, 3), REJERR(i, 4)
```

Keyword: REGCOUPFAC

```
100, 0,            =    IREREJ(3), NREREJ(3)
0., 0., 0., 0.     =    REJFAC(1, 1), REJFAC(1, 2), REJFAC(1, 3), REJFAC(1, 4)
0., 0., 0., 0.     =    REJFAC(2, 1), REJFAC(2, 2), REJFAC(2, 3), REJFAC(2, 4)

           ................................................................
0., 0., 0., 0.     =    REJFAC(i, 1), REJFAC(i, 2), REJFAC(i, 3), REJFAC(i, 4)
```

Keyword: REGUL_TXYZ

100, 0, 1.	=	IREREJ(4), NREREJ(4), REGWID
0., 0., 0., 0.	=	REJTCC(1, 1), REJTCC(1, 2), REJTCC(1, 3), REJTCC(1, 4)
0., 0., 0., 0.	=	REJTCC(2, 1), REJTCC(2, 2), REJTCC(2, 3), REJTCC(2, 4)
		....
0., 0., 0., 0.	=	REJTCC(i, 1), REJTCC(i, 2), REJTCC(i, 3), REJTCC(i, 4)

The three keywords above allow for reading the coupling constants of pseudopotentials (24) or (47) in the format analogous to that used for keyword REGUCOUPLI. Array REJTCC contains the strength parameters of equation (26), that is,

$$
\begin{aligned}
\text{REGVCC}(:, 1) &= \text{REJTCC}(:, 1) \\
\text{REGVCC}(:, 2) &= \text{REJTCC}(:, 1)^* \text{REJTCC}(:, 2) \\
\text{REGVCC}(:, 3) &= \text{REJTCC}(:, 1)^* \text{REJTCC}(:, 3) \\
\text{REGVCC}(:, 4) &= \text{REJTCC}(:, 1)^* \text{REJTCC}(:, 4).
\end{aligned} \tag{91}
$$

Arrays REJERR and REJFAC modify array REGVCC as

$$
\text{REGVCC}(:, :) = \text{REGVCC}(:, :) + \text{REJERR}(:, :)^* \text{REJFAC}(:, :). \tag{92}
$$

This modification of array REGVCC is performed regardless of whether keyword REGUL_TXYZ or REGUCOUPLI was used to define it. In a given run of the code HFODD,

- Keyword REGUCOUPLI must not be simultaneously used with keyword REGUL_TXYZ.
- All read values of array IREREJ must be the same and are internally used as variable IREREG.
- All read values of array NREREJ must be the same and are internally used as variable N3LORD.

To inform the code on whether the coupling constants had been read from the input data file, all elements of array IREREJ are predefined to 100.

3.1.6. Separable terms.

Keyword: SEPARGAUSS

$0 = $ I_SEPA

For I_SEPA > 0, the average mean-field energies of the separable pseudopotential (57) are calculated. For I_SEPA $= 2$ or 3, the corresponding direct mean fields are included in the self-consistent mean field. For I_SEPA $= 2$ or 4, the corresponding exchange mean fields are included in the self-consistent mean field. Altogether, I_SEPA $= 1$ demands calculations of contributions to energy only, whereas I_SEPA $= 2$ demands full self-consistent calculations with both direct and exchange mean fields included. For I_SEPA $= 0$, the mean field corresponding to the separable pseudopotential is ignored.

For I_SEPA > 0 and IPAHFB > 1, the code issues a warning to the effect that, unless the zero-range pairing strengths are explicitly set to zero, see section IV-3.1 [4], the corresponding pairing will still be active. For NOZEPA $= 1$, see section 3.8, the zero-range pairing is neglected, and the warning is not printed. In version (v3.06h) of the code HFODD, I_SEPA > 0 still requires IPNMIX $= 0$, IRENMA $= 0$ and IBROYD $= 0$.

Keyword: SEPAR_PAIR

$0 = $ ISEPPA

For ISEPPA > 0, the average pairing energies of the separable pseudopotential (57) are calculated. For ISEPPA $= 2$, the corresponding pairing fields are included in the self-consistent pairing field. Altogether, ISEPPA $= 1$ demands calculations of contributions to the

pairing energy, whereas I_SEPA = 2 demands full self-consistent calculations with pairing fields included. For I_SEPA = 0, the pairing contribution of the separable pseudopotential is ignored. ISEPPA > 0 requires IPAHFB > 0. In version (v3.06h) of the code HFODD, ISEPPA > 0 still requires IPNMIX = 0, IRENMA = 0 and IBROYD = 0.

Keyword: SEPCOUPLI

0, 0	=	IVISEP, N3SERD
0., 0., 0., 0.	=	SEPVIC(1, 1), SEPVIC(1, 2), SEPVIC(1, 3), SEPVIC(1, 4)

After reading the first line, the code reads one line with the four coupling constants $\tilde{W}$, $\tilde{B}$, $\tilde{H}$ and $\tilde{M}$ and uses separable pseudopotential (57). In version (v3.06h) of the code HFODD, only values of i = IVISEP = 0 or 1 and n = N3SERD = 0 are allowed; other values may become available after higher-order derivative terms are implemented, in analogy to those of the regularized pseudopotential (24). Unless keyword SEPAR_FORM with NUSEGA > 0 is used, the use of keyword SEPCOUPLI implies that the formfactor (58) is composed of one Gaussian only, with the default values of K = NUSEGA = 1, A_1 = SEPGAU(1) = 1 and a_1 = SEPWID(1) = 1. Internally, the code uses variable IRESEP = IVISEP and array SEPVCC = SEPVIC. In a given run of the code HFODD, keyword SEPCOUPLI must not be simultaneously used with keyword SEP_TXYZ.

Keyword: SEP_TXYZ

0, 0	=	IVISEP, N3SERD
0., 0., 0., 0.	=	SEPTIC(1, 1), SEPTIC(1, 2), SEPTIC(1, 3), SEPTIC(1, 4)

Keyword: SEPCOUPERR

0	=	IERSEP
0., 0., 0., 0.	=	SEPERR(1, 1), SEPERR(1, 2), SEPERR(1, 3), SEPERR(1, 4)

Keyword: SEPCOUPFAC

0	=	IFASEP
0., 0., 0., 0.	=	SEPFAC(1, 1), SEPFAC(1, 2), SEPFAC(1, 3), SEPFAC(1, 4)

The three keywords above allow for reading the coupling constants of the separable pseudopotential (57) in the format analogous to that used for keyword SEPCOUPLI. Array SEPTIC contains strength parameters $\tilde{t}$, $\tilde{x}$, $\tilde{y}$ and $\tilde{z}$ defining the coupling constants as $\tilde{W} = \tilde{t}$, $\tilde{B} = \tilde{t}\tilde{x}$, $\tilde{H} = \tilde{t}\tilde{y}$ and $\tilde{M} = \tilde{t}\tilde{z}$, that is,

$$
\begin{aligned}
\text{SEPVIC}(:, 1) &= \text{SEPTIC}(:, 1) \\
\text{SEPVIC}(:, 2) &= \text{SEPTIC}(:, 1) * \text{SEPTIC}(:, 2) \\
\text{SEPVIC}(:, 3) &= \text{SEPTIC}(:, 1) * \text{SEPTIC}(:, 3) \\
\text{SEPVIC}(:, 4) &= \text{SEPTIC}(:, 1) * \text{SEPTIC}(:, 4).
\end{aligned}
\tag{93}
$$

Internally, the code uses variable $\mathtt{IRESEP} = \mathtt{IVISEP}$ and array $\mathtt{SEPVCC} = \mathtt{SEPVIC}$. Arrays $\mathtt{SEPERR}$ and $\mathtt{SEPFAC}$ modify array $\mathtt{SEPVCC}$ as,

$$\mathtt{SEPVCC(:,:)} = \mathtt{SEPVCC(:,:)} + \mathtt{SEPERR(:,:)} * \mathtt{SEPFAC(:,:)}. \tag{94}$$

This modification of array $\mathtt{SEPVCC}$ is performed regardless of whether keyword $\mathtt{SEP\_TXYZ}$ or $\mathtt{SEPCOUPLI}$ was used to define it. In version (v3.06h) of the code HFODD, only values of $i = \mathtt{IERSEP} = \mathtt{IFASEP} = \mathtt{IVISEP} = 0$ or 1 and $n = \mathtt{N3SERD} = 0$ are allowed. In a given run of the code HFODD, keyword $\mathtt{SEP\_TXYZ}$ must not be simultaneously used with keyword $\mathtt{SEPCOUPLI}$.

Keyword: $\mathtt{SEPAR\_FORM}$

1	=	NUSEGA
1., 1., …, 1.	=	SEPWID(1), SEPWID(2), …, SEPWID(NUSEGA)
1., 0., …, 0.	=	SEPGAU(1), SEPGAU(2), …, SEPGAU(NUSEGA)

After reading the first line, for $K = \mathtt{NUSEGA} > 0$ the code reads two lines with widths $a_k = \mathtt{SEPWID}(k)$ and amplitudes $A_k = \mathtt{SEPGAU}(k)$ of Gaussians that define formfactor (58) of the separable pseudopotential (57).

3.2. Symmetries

Keyword: $\mathtt{HFB2HF}$
$0, 0, = \mathtt{IPA2HF}(0), \mathtt{IPA2HF}(1)$
Keyword: $\mathtt{GAP2HF}$
$0., 0., = \mathtt{DEL2HF}(0), \mathtt{DEL2HF}(1)$

The two keywords above allow for the use of a hybrid method of calculations, where the HF method is used for neutrons (protons), $\mathtt{IPA2HF}(0) = 1(0)$, and the HFB method is used for protons (neutrons), $\mathtt{IPA2HF}(0) = 0(1)$. For $\mathtt{IPA2HF}(0) = \mathtt{IPA2HF}(1) = 0$, the hybrid method is inactive and the code proceeds as dictated by other keywords handling the HF/HFB method, whereas for $\mathtt{IPA2HF}(0) = \mathtt{IPA2HF}(1) = 1$, the HF method is enforced for both neutrons and protons, irrespective of what is dictated by other keywords handling the HF/HFB method.

For $\mathtt{IPA2HF}(0) = 2$ or $\mathtt{IPA2HF}(1) = 2$, the HFB calculations requested for neutrons or protons will *during the iterations* automatically switch over to HF as soon as the neutron or proton pairing gap goes below $\mathtt{DEL2HF}(0)$ or $\mathtt{DEL2HF}(1)$, respectively. The user is responsible for properly setting the keywords handling the HF method before the HFB run is started, because the correctness and consistency of these keywords would not have been pre-tested. $\mathtt{IPA2HF}(0) > 0$ or $\mathtt{IPA2HF}(1) > 0$ requires $\mathtt{IPAHFB}(0) > 0$.

3.3. Symmetry restoration

Keyword: $\mathtt{PROJPARNUM}$
$0, 1 = \mathtt{IPRNUM}, \mathtt{NPNKNO}$

For $\mathtt{IPRNUM} = 1(2)$ and $\mathtt{NPNKNO} > 1$, and for diagonal (non-diagonal) GCM kernels, see section VI-3.2 [6], the code performs a projection on total particle number $A = \mathtt{IN\_FIX} + \mathtt{IZ\_FIX}$. The number of Gauss–Tchebyschev points used to perform the integration over gauge angle ϕ covering the domain of $0 \leqslant \phi < \pi$ is defined by $\mathtt{NPNKNO}$. $\mathtt{IPRNUM} > 0$ requires that it is equal to all other nonzero projection switches: $\mathtt{IPRROT}$, $\mathtt{IPRISO}$, $\mathtt{IPRVEC}$ and $\mathtt{IPRPTY}$, see section 2.16.6.
Keyword: $\mathtt{PROJVECNUM}$

$0, 1 = $ IPRVEC, NTZKNO

For IPRVEC $= 1(2)$ and NTZKNO > 1, and for diagonal (non-diagonal) GCM kernels, see section VI-3.2 [6], the code performs a projection on the doubled z-component of the isospin. $2T_z = $ IN_FIX-IZ_FIX. The number of Gauss–Tchebyschev points used to perform the integration over gauge angle ϕ_T covering the domain of $0 \leqslant \phi_T < \pi$ is defined by NTZKNO. IPRNUM > 0 requires that it is equal to all other nonzero projection switches: IPRROT, IPRISO, IPRNUM and IPRPTY, see section 2.16.6.

Keyword: PROJPARITY

$0, 0, +1 = $ IPRPTY, NPAKNO, IPAPRO

For IPRPTY $= 1(2)$ and NPAKNO $= 2$, and for diagonal (non-diagonal) GCM kernels, see section VI-3.2 [6], the code performs parity projection onto the positive-parity (for IPAPRO $= +1$) or negative-parity (for IPAPRO $= -1$) states. IPRPTY > 0 requires that it is equal to all other nonzero projection switches: IPRROT, IPRISO, IPRNUM and IPRVEC, see section 2.16.6.

Keyword: ONISHI

$0 = $ IONISH

For IONISH $= 0$ or 1, the code uses the Pfaffian or Onishi formula, respectively, see section 2.9, to compute the overlap kernels between the HFB wave functions involved in the symmetry projection (keywords PROJECTGCM, PROJPARITY, PROJPARNUM or PROJVECNUM). IPRGCM > 1 with IPAHFB > 1 and IONISH $= 0$ requires ITWOBA $= 1$ and NUQEVE $= 1$. In version (v3.06h) of the code HFODD, IPRGCM > 1 with IPAHFB > 1 and IONISH $= 0$ still requires ISIMPY $= 0$ and ISIQTY $= 0$.

Keyword: PROJ_J2_T2

$0, 0 = $ KETAJ2, KETAT2

For KETAJ2 $= 1$ (KETAT2 $= 1$) and IPRGCM > 0, the code computes the expectation value of the square of the total angular momentum (total isospin) in the projected states. This calculation is performed to control the precision of the angular-momentum (isospin) projection. KETAJ2 $= 1$ (KETAT2 $= 1$) is ignored unless NUBKNO $= 1$ (NBTKNO $= 1$). Either KETAJ2 $= 1$ or KETAT2 $= 1$ requires ISAKER $= 0$ or 2.

Keyword: KERNINVERS

$0, 0, = $ IKEINV, IKEKAR

For IKEINV > 0 or IKEKAR > 0, and for IPRGCM $= 1$, the 'right' wave function of the kernel calculated within the diagonal GCM mode, see section VI-3.2 [6], is initially transformed according to one of the D_{2h}^{T} symmetry operations enumerated for variables INIINV and INIKAR under keyword INI_INVERS in section VI-3.2 [6]. Compared to the operations requested by keyword INI_INVERS, which generates the transformed wave function that can later be used by employing keyword PROJECTGCM in the non-diagonal GCM mode, see section VI-3.2 [6], keyword KERNINVERS ensures that the phase relations between the 'right' and 'left' wave functions are properly maintained.

Keyword: NOBLOLIPKI

$0, 0, = $ LIPNON, LIPNOP

For LIPNON $= 1$ or LIPNOP $= 1$, the blocked neutron or proton orbitals are excluded from the calculation of the neutron or proton Lipkin parameters λ_2, respectively. Since the occupation probabilities of the blocked states are by definition equal to 1, their contributions to particle-number fluctuations should not be, in principle, counted. LIPNON $= 1$ requires LIPKIN $= 1$ and LIPNOP $= 1$ requires LIPKIP $= 1$.

Keyword: PROJE_DENS

$0 = $ IDENSU

For IDENSU = 1, the particle-number projection is performed with ignored gauge-angle dependence of the density in the p-h and p-p density-dependent terms.

3.4. Configurations

Keyword: AXIALIZE

　0 = IAXIAP

For |IAXIAP| = 1, the code axializes, see section 2.11, the particle-hole and pairing (for IPAHFB > 0) mean-fields. For IAXIAP = −1, the code in addition axializes the particle-hole density matrix and pairing tensor (for IPAHFB > 0). For |IAXIAP| = 1, ICONTI = 1 and IFCONT = 0, the code issues a warning to the effect that a smooth continuation of axialized wave functions may require continuation from fields, that is, IFCONT = 1, see section VI-3.8 [6]. In version (v3.06h) of the code HFODD, |IAXIAP| = 1 still requires IGOGPA = 0 and IAXIAP = −1 is incompatible with the AMP (NUBKNO > 1).

　Keyword: VACNONANEU

0, 1	=	NLSIZN, MXALIN
0, …, 0	=	LALSIZ(−MXALIN, 0),
		LALSIZ(−MXALIN + 2, 0),
		....
		LALSIZ(+MXALIN− 2, 0),
		LALSIZ(+MXALIN, 0)

After reading the first line, the code reads the second line that contains MXALIN + 1 numbers LALSIZ(:, 0) of neutrons in the Ω-blocks, from $\Omega = -\text{MXALIN}/2$ to $\Omega = +\text{MXALIN}/2$, see section 2.14. |NLSIZN| = 1, 2 or 3 stands for the Cartesian direction of x, y, or z, respectively. For NLSIZN > 0, the code distributes neutrons in the Ω-blocks according to the values of LALSIZ(i, 0). For LALSIZ(i, 0) > 0, the |LALSIZ(i, 0)| lowest neutron states are occupied in the block $\Omega = i/2$. For LALSIZ(i, 0) < 0, the |LALSIZ(i, 0)| − 1 lowest neutron states are occupied in the block $\Omega = i/2$, the state number |LALSIZ(i, 0)| is kept empty, and the state number |LALSIZ(i, 0)| + 1 is kept occupied. For NLSIZN < 0, the code does not fix occupations in the Ω-blocks but determines and prints the distribution of neutrons across the Ω-blocks. MXALIN must be odd. For IPAIRI = 0, IPNMIX = 0, ISIMPY = 0, ISIQTY = 0, IVACUM = 0 and NLSIZN > 0, the sum of numbers of neutrons in all Ω-blocks, that is, the sum of |LALSIZ(:, 0)|, must be equal to the number of neutrons given by IN_FIX. |NLSIZN| > 0 requires ISIMPY = 0 and ISIQTY = 0.

　Keyword: VACNONAPRO

0, 1	=	NLSIZP, MXALIP
0, …, 0	=	LALSIZ(−MXALIP, 1),
		LALSIZ(−MXALIP + 2, 1),
		....
		LALSIZ(+MXALIP− 2, 1),
		LALSIZ(+MXALIP, 1)

The same as in keyword VACNONANEU but for protons.

Keyword: `VACPARANEU`

0, 1	=	NLSIQN, MXALIN
0, ..., 0	=	LALSIQ(−MXALIN, 0, 0),
		LALSIQ(−MXALIN + 2, 0, 0),
		....
		LALSIQ(+MXALIN− 2, 0, 0),
		LALSIQ(+MXALIN, 0, 0)
0, ..., 0	=	LALSIQ(−MXALIN, 1, 0),
		LALSIQ(−MXALIN + 2, 1, 0),
		....
		LALSIQ(+MXALIN− 2, 1, 0),
		LALSIQ(+MXALIN, 1, 0)

After reading the first line, the code reads two lines that each contain $MXALIN + 1$ numbers of neutrons `LALSIQ(:, 0, 0)` and `LALSIQ(:, 1, 0)` in the positive-parity and negative-parity Ω-blocks, respectively, from $\Omega = -MXALIN/2$ to $\Omega = +MXALIN/2$, see section 2.14. $|NLSIQN| = 1, 2$ or 3 stands for the Cartesian direction of x, y, or z, respectively. For $NLSIQN > 0$, the code distributes neutrons in the positive-parity and negative-parity Ω-blocks according to the values of $LALSIQ(i, j, 0)$, where $j = 0(1)$ stands for the positive (negative) parity. For $LALSIQ(i, j, 0) > 0$, the $|LALSIQ(i, j, 0)|$ lowest neutron states are occupied in the block $\Omega = i/2$ for a given parity. For $LALSIQ(i, j, 0) < 0$, the $|LALSIQ(i, j, 0)| - 1$ lowest neutron states are occupied in the block $\Omega = i/2$ for a given parity, the state number $|LALSIQ(i, j, 0)|$ is kept empty, and the state number $|LALSIQ(i, j, 0)| + 1$ is kept occupied. For $NLSIQN < 0$, the code does not fix occupations in the Ω-blocks but determines and prints the distribution of neutrons across the Ω-blocks of both parities. MXALIN must be odd. For $IPAIRI = 0$, $IPNMIX = 0$, $ISIMPY = 0$, $ISIQTY = 1$, $IVACUM = 0$ and $NLSIQN > 0$, the sum of the number of neutrons in all Ω-blocks of a given parity, that is, the sum of $|LALSIQ(:, j, 0)|$, must be equal to the number of neutrons given by $KVASIQ(j, 0)$, see section IV-3.3 [4].[10] $|NLSIQN| > 0$ requires $ISIMPY = 0$ and $ISIQTY = 1$.

Keyword: `VACPARAPRO`

0, 1	=	NLSIQP, MXALIN
	=	LALSIQ(−MXALIN, 0, 1),
		LALSIQ(−MXALIN + 2, 0, 1),
		....
		LALSIQ(+MXALIN− 2, 0, 1),
0, ..., 0		LALSIQ(+MXALIN, 0, 1)
0, ..., 0	=	LALSIQ(−MXALIN, 1, 1),
		LALSIQ(−MXALIN + 2, 1, 1),
		....
		LALSIQ(+MXALIN− 2, 1, 1),
		LALSIQ(+MXALIN, 1, 1)

The same as in keyword `VACPARANEU` but for protons.

---

[10] In [4] p 175, description of keywords `VACPAR_NEU` and `VACPAR_PRO` should refer to variables `KVASIQ(0,0)`, `KVASIQ(1,0)` and `KVASIQ(0,1)`, `KVASIQ(1,1)`, respectively.

Keyword: VACLASTORB

$0, 0, =$ LASTAN, LASTAP

For LASTAN $= 1$ or LASTAP $= 1$, calculations are performed with only one, highest-energy neutron or proton orbital, respectively, occupied in each Ω-block, see section 2.14. This option can be used to perform the AMP of a single orbital, after the code is restarted from a converged solution. For LASTAN $= 1$, the option of using negative values of LALSIZ or LALSIQ, see keywords VACNONANEU or VACPARANEU, respectively, is not allowed. The same rule applies to protons.

Keyword: FILNON_NEU

$2, 1, 0 =$ KPFILZ(0), KHFILZ(0), KOFILZ(0)

Keyword FILNON_NEU is an analog of keyword FILSIG_NEU, see section VI-3.2 [6], and demands calculations performed within the filling approximation applied to neutrons in the no-symmetry case. Variables KPFILZ(0) and KHFILZ(0) contain indices of particle (empty) and hole (occupied) states, respectively. Variable KOFILZ(0) contains the number of particles put into the states between KHFILZ(0) and KPFILZ(0) by using partial occupation factors of KOFILZ(0)/(KPFILZ(0) $-$ KHFILZ(0) $+$ 1) for them. For KOFILZ(0) $= 0$, the filling approximation is inactive. KOFILZ(0) > 0 is incompatible with IPAIRI $= 1$, IFLIPI $\neq 0$ or KOFLIZ(0) $\neq 0$.

Keyword: FILNON_PRO

$2, 1, 0 =$ KPFILZ(1), KHFILZ(1), KOFILZ(1)

The same as for keyword FILNON_NEU but it demands calculations performed within the filling approximation applied to protons in the no-symmetry case.

Keyword: MBLOCSIZ_N

0	$=$	NBBLOC
1, 0	$=$	INSIZN(1), IDSIZN(1),
1, 0	$=$	INSIZN(2), IDSIZN(2),
		....
1, 0	$=$	INSIZN(NBBLOC), IDSIZN(NBBLOC)

Keyword MBLOCSIZ_N generalizes the quasiparticle blocking requested by keyword BLOCKSIZ_N in the case of no symmetries, see section IV-3.3 [4], to multi-quasiparticle blocking, see section 2.8. After reading the first line, the code reads NBBLOC pairs of data, INSIZN(i) and IDSIZN(i), for $i = 1, \ldots,$ NBBLOC. For IDSIZN(i) $= +1$ or -1, the blocked quasiparticle state is selected by having the largest overlap with the INSIZN(i)th neutron single-particle eigenstate of the HFB mean-field Routhian or with its time-reversed partner, respectively. Note that for rotating states, the time-reversed eigenstate is not necessarily an eigenstate of the Routhian. For IDSIZN(i) $= 0$, the blocking of the ith quasiparticle is omitted. For any i, $|$IDSIZN(i)$| = 1$ requires ISIMPY $= 0$, IPARTY $= 0$, IPAHFB $= 1$ and IROTAT $= 1$. In a given run of the code HFODD, keyword MBLOCSIZ_N must not be simultaneously used with keyword BLOCKSIZ_N.

Keyword: MBLOCSIZ_P

0	$=$	NBBLOC
1, 0	$=$	INSIZP(1), IDSIZP(1),
1, 0	$=$	INSIZP(2), IDSIZP(2),
		....
1, 0	$=$	INSIZP(NBBLOC), IDSIZP(NBBLOC)

The same as for keyword MBLOCSIZ_N but for the proton multi-quasiparticle blocking.
Keyword: NUMBCUTOFF

$0, 0 = $ NCUTOF(0), NCUTOF(1)

For NCUTOF(0) > 0 and/or NCUTOF(1) > 0, the number of lowest neutron and/or proton single-particle states used in the two-basis method, see section VII-2.2.1 [7], are limited to NCUTOF(0) and/or NCUTOF(1), respectively. This option overrides the energy cutoff specified by variable ECUTOF read under keyword CUTOFF, see section IV-3.1 [4]. This option allows for the calculation of the transition densities when it is performed for different left and right HFB states, that is for IPRGCM > 1. Indeed, this type of calculation requires that the numbers of quasiparticle states defining the left and right HFB states are equal. NCUTOF(0) > 0 and/or NCUTOF(1) > 0 is incompatible with ITWOBA $= 0$ (for now), LIMQUA $= 1$ or LAMCUT $= 1$.

3.5. Numerical parameters

Keyword: ADPARBASIS

$0 = $ ILIBAS

In version (v3.06h) of the code HFODD, the names of variables defining the oscillator frequency $\hbar\omega_0$, which are read under keyword SURFAC_PAR, have changed and now read INBASI, IZBASI and R0PARM, see section II-3.5 [2]. This allows for dynamically linking the original variables INNUMB and IZNUMB to neutron and proton numbers IN_FIX and IZ_FIX, see section II-3.1 [2], depending on the value of variable ILIBAS, which has the allowed values of 0, 1, 2 and 3. Namely,

- For ILIBAS $= 1$ or ILIBAS $= 3$, INNUMB is set to IN_FIX; otherwise it is set to INBASI.
- For ILIBAS $= 2$ or ILIBAS $= 3$, IZNUMB is set to IZ_FIX; otherwise it is set to IZBASI.

For example, for ILIBAS $= 0$, the code defines $\hbar\omega_0$ (as before) by using variables read under keyword SURFAC_PAR, whereas for ILIBAS $= 3$, it does so by using the neutron and proton numbers IN_FIX and IZ_FIX.

Keyword: NEW_WIGNER

$0 = $ NEWWIG

For NEWWIG $= 0$ or 1, version (v3.06h) of the code HFODD uses the old and new method to calculate the Wigner d functions, respectively, see section 2.12.

Keyword: EVENQPNUMB

$0 = $ NUQEVE

For ITWOBA $= 1$ or IPRGCM > 0, NUQEVE $= 1$ enforces even numbers of single-particle and quasiparticle states, which is needed for the implementation of the Pfaffian method, see section 2.9. In version (v3.06h) of the code HFODD, NUQEVE $= 1$ still requires ISIMPY $= 0$.

3.6. Output parameters

Keyword: ALLNILABS

$0 = $ INUNIL

For INUNIL > 0 and IREVIE > 0, up to INUNIL $= 99$, non-dominant Nilsson labels are printed for each single-particle state on the REVIEWFILE, see section II-3.9 [2].

Keyword: PRINT_ELEC

$0 = $ IELPRI

Switch IELPRI defines the type of matrix elements of the electric, magnetic, surface or Schiff operators between angular-momentum-projected states, which are printed in version (v3.06h) of the code HFODD.

- For IELPRI $= 0$, the code prints reduced matrix elements of the operators defined in the code, see section IV-2.4 [4].
- For IELPRI $= 1$, the code prints reduced matrix elements of the standard operators, that is, the particular units defined in section IV-2.4 [4] are removed.
- For IELPRI $= 2$, the code prints the standard reduced transition rates $\mathrm{BE}\lambda(I_i \rightarrow I_f)$ and/or $\mathrm{BM}\lambda(I_i \rightarrow I_f)$.
- For IELPRI $= 3$, the code prints the standard spectroscopic matrix elements.

Keyword: PRINTMATEL

$1, 0 =$ ILIMAM, IALLAM

For ILIMAM $= 0(1)$, the code does not (does) calculate reduced kernels and/or matrix elements that are not required for printing, as defined by the ranges of angular momenta specified in variables ISLPRI, ISUPRI, see section VI-3.6 [6]. For IALLAM $= 0(1)$, the code does not (does) print reduced kernels and/or matrix elements between the 'left' (bra) angular momentum larger than the 'right' (ket) angular momentum. Those between smaller or equal angular momenta are always printed.

Keyword: PRINTALLRM

$0 =$ IPRALL

For IPRALL $= 1$, the code prints all reduced kernels and/or matrix elements irrespective of restrictions otherwise imposed by switches IPRGCM, IELPRI or IAXIAL.

Keyword: REDMATSAVE

$0 =$ IWRIRM

For IWRIRM $= 1$, an ASCII file with the reduced kernels and/or matrix elements, see keyword REDMATFILE, is saved on disk after the AMP is performed. ILIMAM $= 1$ allows for saving the reduced kernels and/or matrix elements that are not required for printing, see keyword PRINTMATEL. IWRIRM $= 1$ requires IPRGCM > 0 and is incompatible with IFTEMP $= 1$.

Keyword: REDMATFILE

HFODD.RED $=$ FILRED

CHARACTER$*68$ is the file name of the ASCII with reduced kernels and/or matrix elements. Must start at the 13th column of the data line.

Keyword: EFF_G_FACT

$0, 1., 1., 1. =$ IGYROS, GYRORP, GYRSPN, GYRSPP

For IGYROS $= 1$, the standard single-particle orbital and spin gyroscopic factors, $g_l^{\nu,\mathrm{s.p.}}$ and $g_s^{\nu,\mathrm{s.p.}}$, respectively, for neutrons and protons, $\nu =$ n, p, which are used to calculate the magnetic moments, see section IV-2.4 [4], are multiplied by the corresponding effective gyroscopic factors and read:

$$
\begin{aligned}
g_l^p &= g_l^{p,\mathrm{s.p.}} * g_l^{p,\mathrm{eff}} = +1.000 * \mathrm{GYRORP}, \\
g_s^n &= g_s^{n,\mathrm{s.p.}} * g_s^{n,\mathrm{eff}} = -3.826 * \mathrm{GYRSPN}, \\
g_s^p &= g_s^{p,\mathrm{s.p.}} * g_s^{p,\mathrm{eff}} = +5.586 * \mathrm{GYRSPP}.
\end{aligned}
\tag{95}
$$

Keyword: QUASIPSAVE

$-1 =$ IWRIQU

For IWRIQU $= 1$, a binary quasiparticle file, see keyword QUASIPFILE, is saved on disk after each iteration is completed. The file contains quasiparticle wave functions.

For `IWRIQU` $= 0$, the file is saved only once, after all iterations are completed. For `IWRIBA` $= -1$, the file is never saved. `IWRIQU` $= 0$ or 1 requires `IPAHFB` > 0 and is incompatible with `IFTEMP` $= 1$ or `IF_RPA` $= 1$. In version (v3.06h) of the code HFODD, `IWRIQU` $= 0$ or 1 still requires `ISIMPY` $= 1$ and `IPNMIX` $= 0$.

Keyword: `QUASIPFILE`

`HFODD.QUA = FILQUA`

`CHARACTER*68` is the file name of the binary file that contains quasiparticle wave functions. Must start at the 13th column of the data line.

3.7. Starting, performing, stopping and restarting iterations

Keyword: `MAXANTICON`

$0., 0 =$ `EPSCON, NUCONS`

For `NUCONS` > 0, iterations stop when the changes in the stability energy, equation (I-37) [1], stay below `EPSCON*EPSITE` for `NUCONS` consecutive iterations. This option aims to stop iterations when the convergence is extremely slow or when the wave function infinitely alternates between two solutions both having the same value of the stability energy. Note that the alternating signs of the stability energy are recognized by the ping-pong divergence condition, see sections III-2.6 and III-3.1 [3] and keyword `PING_PONG`. `NUCONS` > 0 requires `EPSCON` > 0.

Keyword: `QUASISTABI`

$0 =$ `IQPSTA`

For `IQPSTA` $= 1$, expression (I-37) [1] for the stability energy is replaced by expression (89), which is suitable for the HFB calculations. `IQPSTA` $= 1$ requires `IPAHFB` > 0. In version (v3.06h) of the code HFODD, `IQPSTA` $= 1$ still requires `IPNMIX` $= 0$.

Keyword: `SLOWALLFIL`

$0.5, 0 =$ `SLOWAL, I_SLOW`

For `I_SLOW` $= 1$, the rate of convergence is slowed down by a factor of `SLOWAL` by mixing the mean field and pairing matrices on the harmonic-oscillator basis instead of mixing the mean-field potentials on the Gauss–Hermite spatial nodes, see keywords `SLOW_DOWN`, section II-3.5 [2], `SLOWLIPKIN`, section VI-3.2 [6], `SLOW_PAIR`, section IV-3.4 [4] and `SLOWLIPMTD`, section VIII-3.1.2 [8]. `I_SLOW` $= 1$ is incompatible with `IBROYD` $= 1$.

Keyword: `BASIS_SAVE`

$-1 =$ `IWRIBA`

For `IWRIBA` $= 1$, the basis file is saved on disk after each iteration is completed. The file contains Bohr deformation parameters that can be used to restart calculations with the basis deformation parameters equal to those read from the basis file, see section 2.13 and keyword `CONT_BASIS`. For `IWRIBA` $= 0$, the file is saved only once, after all iterations are completed. For `IWRIBA` $= -1$, the file is never saved.

Keyword: `REPBASFILE`

`HFODD.BAP = FILBAP`

`CHARACTER*68` is the file name of the basis file. Must start at the 13th column of the data line. For `IBCONT` $= 1$, see keyword `CONT_BASIS`, an ASCII basis file with the name defined in `FILBAP` must exist, and will be read. If the filenames `FILBAP` and `FILBAC` are identical, the basis file will be subsequently overwritten as a new basis file.

Keyword: `RECBASFILE`

`HFODD.BAC = FILBAC`

`CHARACTER*68` is the file name of the basis file. Must start at the 13th column of the data line. For `IWRIBA` $= 0$ or 1, an ASCII basis file is saved, see keyword `BASIS_SAVE`.

Keyword: CONT_BASIS

0 = IBCONT

For IBCONT = 1, the Bohr deformation parameters stored on the basis file are used to restart calculations with the basis deformation parameters equal to those read from the basis file, see section 2.13. For IBCONT = 1, an ASCII basis file with the name defined in FILBAP must exist, and it will be read. IBCONT = 1 is incompatible with ICONTI = 0.

3.8. Miscellaneous

Keyword: ANTISYMPAI

0 = KAPASY

For KAPASY = 1, the pairing tensor is antisymmetrized, which removes its possible nonzero symmetric component that can appear for the quasiparticle cutoff, see section IV-3.1 [4] and [78, 79].

Keyword: FERMICUT

0 = LAMCUT

For LAMCUT = 1, the quasiparticle cutoff, see section IV-3.1 [4], is applied relatively to the proton and neutron Fermi energies, and not relatively to the zero of the equivalent single-particle spectrum, which is the default.

Keyword: LAN4SCALED

0 = LANSCA

For LANSCA = 1, the Landau parameters, see sections IV-2.8 and IV-3.1 [4], are used for scaled, see section II-3.2 [2], coupling constants of the functional and not for the unscaled ones, which is the default.

Keyword: NOZEROPAIR

0 = NOZEPA

For NOZEPA = 1, the zero-range pairing force is neglected regardless of the values of pairing strengths defined in section IV-3.1 [4].

3.9. New features of previously implemented keywords

Keyword: OPTI_GAUSS

1 = IOPTGS

Apart from the value of IOPTGS = 1, implemented in section II-3.5 [2], whereupon expression (I-94) was used to calculate the orders of the Gauss–Hermite integrations in the three Cartesian directions, that is, NXHERM = 2*NXMAXX + 2, NYHERM = 2*NYMAXX + 2, NZHERM = 2*NZMAXX + 2, the version (v3.06h) of the code HFODD now accepts a new value of IOPTGS = 2, for which NXHERM = 3*NXMAXX + 2, NYHERM = 3*NYMAXX + 2, NZHERM = 3*NZMAXX + 2. The former (latter) values ensure exact Gauss–Hermite integrations of the two-body (three-body) zero-range terms with second-order gradients. The latter values are thus suitable for calculations described in section 2.2. For IOPTGS = 0 or IREAWS = 1, the above expressions do not overwrite values read under keyword GAUSHERMIT.

Keyword: MULTCONSTR

2, 0, 0.01, 42.0, 1 = LAMBDA, MIU, STIFFQ, QASKED, IFLAGQ

Apart from the value of IFLAGQ = 1, previously implemented in section II-3.7 [2], the version (v3.06h) of the code HFODD now accepts a new value of IFLAGQ = −1 mentioned in section 2.13, whereupon the value of QASKED for λ = LAMBDA and μ = MIU is used for a definition of the HO basis, whereas the corresponding constraint is ignored. For IFLAGQ = 0, the value of QASKED is ignored. Recall that unless IFLAGQ = 0 is explicitly set in the input data file for LAMBDA = 2 and MIU = 0, the constraint on Q_{20} = 42b would be active by default.

Keyword: FREQBASIS

1.0, 1.0, 1.0, 0 = BASINX, BASINY, BASINZ, INPOME

Apart from the value of INPOME = 1, previously implemented in section VIII-3.1.4 [8], and the default value of INPOME = 0, version (v3.06h) of the code HFODD now accepts the new values of INPOME = 2, ..., 7 described in section 2.13.

Keyword: REVIEW

2 = IREVIE

Apart from the values of IREVIE = 0, 1 and 2, previously implemented in section II-3.9 [2], version (v3.06h) of the code HFODD now accepts new values of IREVIE = −2, and 3, ..., 8, which allows for printing on the REVIEW file the following *additional* information:

- For IREVIE = −2, the quasiparticle data are printed.
- For IREVIE ⩾ 3, the *x* and *z* single-particle alignment data are printed.
- For IREVIE ⩾ 4, the proton-neutron single-particle data are printed.
- For IREVIE ⩾ 5, the proton-neutron single-particle alignment data are printed.
- For IREVIE ⩾ 6, the integration points are printed.
- For IREVIE ⩾ 7, the integration weights are printed.
- For IREVIE = 8, the densities are printed.

Keyword: BASIS_SIZE

15, 301, 800.0 = NOSCIL, NLIMIT, ENECUT

Apart from the values of NLIMIT > 0, previously implemented in section II-3.6 [2], version (v3.06h) of the code HFODD now accepts the value of NLIMIT = 0, whereupon the value of NLIMIT is instantly recalculated to NLIMIT = ((NOSCIL + 1)*(NOSCIL + 2)*(NOSCIL + 3))/6, which corresponds to the number of states of a spherical HO with the total number of quanta not exceeding NOSCIL.

Keyword: CONTLIPKIN

0 = ILCONT

Apart from the values of ILCONT = 0 or 1, previously implemented in section VI-3.8 [6], version (v3.06h) of the code HFODD now accepts the value of ILCONT = 2. This value does not request reading the LIPKIN FILE, but allows for reading the Lipkin–Nogami parameters λ_2 from the RECORD FILE stored in a previous run. In conjunction with reading the FIELDS FILE (IFCONT = 1), ILCONT = 2 thus allows for a smooth continuation of the Lipkin–Nogami calculations. ILCONT > 0 is incompatible with either of LIPKIN = LIPKIP = 0 or IPCONT = 0.

Keyword: HFB

0 = IPAHFB

Apart from the values of IPAHFB = 0 or 1, previously implemented in section IV-3.2 [4], version (v3.06h) of the code HFODD now accepts the value of IPAHFB = 2, whereupon the HFB densities are summed in the canonical basis. IPAHFB > 0 requires IPAIRI = 1, see section II-3.3 [2], and IPAHFB = 2 is incompatible with ITWOBA = 1, IMFHFB = 1, IFSHEL > 0 or IPNMIX = 1. However, the HFB method was already implemented in all symmetries and thus in version (v3.06h) of the code HFODD the restriction to ISIMPY = 1, specified in section IV-3.2 [4], was lifted.

Keyword: PROJECTGCM

0, 0, 0, 1, 1, 0, 1, 1, 0
IPRROT, IPROMI, IPROMA,   NUAKNO, NUBKNO, KPROJE,
                              IFRWAV, ITOWAV, IWRWAV

The name of the first variable was changed to IPRROT and thus this variable was made independent from the first variable read under keyword PROJECTGCM, see section 2.16.6. For IPRROT = 0, the remaining input data read under keyword PROJECTGCM are now ignored. For IPRGCM = 2, apart from a positive value of IFRWAV, previously implemented in section VI-3.2 [6], version (v3.06h) of the code HFODD now accepts its negative value, whereupon the calculation of the GCM kernels is performed between states with labels −IFRWAV and ITOWAV only, and not between *all* states with labels from a positive label IFRWAV to ITOWAV. In version (v3.06h) of the code HFODD, the AMP was implemented for the GCM kernels, so for NUAKNO ≠ 1 or NUBKNO ≠ 1, IPRGCM = 2 is now allowed.

Keyword: PROJECTISO

0, 2, 1, 1.E-6, 0, 0 = IPRISO, ISOSAD, NBTKNO, EPSISO, ICSKIP, IFERME

The name of the first variable was changed to IPRISO and thus this variable was made independent from the first variable read under keyword PROJECTGCM, see section 2.16.6. For IPRISO = 0, the remaining input data read under keyword PROJECTISO are now ignored.

Keyword: SAVEKERNEL

0 = ISAKER

Apart from the values of ISAKER = 0 or 1, previously implemented in section VI-3.2 [6], version (v3.06h) of the code HFODD now accepts the value of ISAKER = 2, whereupon the kernel file is stored on the disk in a new format. Although the old format, requested by ISAKER = 1, is still supported, it should be considered obsolete. The use of the old format is not recommended because some newly developed features may then be improperly stored. In particular, ISAKER = 1 is incompatible with KETAJ2 = 1, KETAT2 = 1, NPNKNO > 1, NTZKNO > 1 or NPAKNO > 1.

For ISAKER = 2 and IPAKER = 1 (see section VI-3.2 [6]), for all values of indices 't' the code attempts to read the kernel files Nxxxxxt-Lyyy-Rzzz-//FILKER, where // denotes concatenated strings. The one-, three- or five-digit indices are:

- 'xxxxx' is the consecutive index of the kernel file, which is equal to KFIKER (see section VII-3.2 [7]).
- 't' is the number from 0–9 of the consecutive file having the given index 'xxxxx'.
- 'yyy' is the number of the left wave function.
- 'zzz' is the number of the right wave function.

In the work directory, the file names for all indices 'xxxxxt' are scanned, starting from xxxxx0. The kernels stored in these files are read into memory and are not recalculated. Those that have not been found in the kernel files are calculated and stored in the kernel file with the lowest available index 't'. In this way, one can submit many parallel jobs, see the keyword PARAKERNEL (see section VI-3.2 [6]). The results are then collected in different kernel files with indices 't' attributed automatically. If any of the jobs is terminated before completing its task, the same input data can be resubmitted and the calculation automatically continues from the point where it has been interrupted. Once all the kernels have been calculated (with IPAKER = 1), which requires large CPU time, AMP can be performed (with IPAKER = 0) within a very small CPU time by reading, again automatically, all the created kernel files

with indices 'xxxxxt'. At the AMP stage, KFIKER denotes the maximum index 'xxxxx' of the kernel files that were stored. Note that if at the AMP stage any kernels are missing, the code will attempt to calculate them. For $\text{ISAKER} = 2$, this feature can be overridden by using $\text{IPAKER} = -1$ instead of $\text{IPAKER} = 0$, whereupon if any kernels are missing, the code will stop. $\text{ISAKER} = 2$ requires $\text{IPRGCM} > 0$ and $0 < \text{KFIKER} < 99\,999$.

Keyword: PARAKERNEL

$0, 1, 1, 1, 1 = \text{IPAKER, NUASTA, NUASTO, NUGSTA, NUGSTO}$

Apart from the values of $\text{IPAKER} = 0$ or 1, previously implemented in section VI-3.2 [6], version (v3.06h) of the code HFODD now accepts the value of $\text{IPAKER} = -1$. For $\text{ISAKER} = 2$, $\text{IPAKER} = -1$ is equivalent to $\text{IPAKER} = 0$. However, if in the earlier parallel runs (with $\text{IPAKER} = 1$) any kernels were not yet calculated, the code will stop instead of attempting to calculate them.

4. Fortran source files

The FORTRAN source of version (v3.06h) of the code HFODD is provided in the file hf306h.f, and its accompanying modules are:

- hfodd_sizes_7.f90: static array size declarations. Contains all PARAMETER statements controlling the sizes of all statically allocated arrays (and some of the dynamically allocated arrays) used in the code. The static allocations are maintained in the code because the compiler optimization options are often more efficient when the dimensions of the arrays are known to the compiler. In practice, only a few size declarations need to be defined by the user. In module hfodd_sizes_7.f90, these declarations are collected at the beginning of the module and read:

NDMAIN	Maximum number of HO shells
NDBASE	Maximum number of HO basis states
NDSTAT	Maximum number of HF states without spin
NDXHRM	Maximum number of Gauss−Hermite nodes in x direction
NDYHRM	Maximum number of Gauss−Hermite nodes in y direction
NDZHRM	Maximum number of Gauss−Hermite nodes in z direction
NDPROI	Maximum doubled spin in the AMP
NDAKNO	Maximum number of nodes in the α and γ Euler AMP angles
NDBKNO	Maximum number of nodes in the β Euler AMP angle
NDPROT	Maximum doubled isospin in the IP
NDATKN	Maximum number of nodes in the α_T and γ_T Euler IP angles
NDBTKN	Maximum number of nodes in the β_T Euler IP angles

Any version 5 of the module, hfodd_sizes_5.f90, can be upgraded to version 7 by copying a few lines of the code located at the end of module hfodd_sizes_7.f90.

The code HFODD can be perfectly well run with array sizes smaller than the maximum ones specified above; the only consequence would be non-optimal memory usage. If a given requested array size exceeds the maximum, the code stops and prints the new maximum size that has to be used at compilation.

- hfodd_modules_35.f: definitions of memory-consuming modules. Defines, among others, the matrices of the Bogolyubov transformation, the eigenvectors of the HF and HFB equations, etc.
- hfodd_hfbtho_201.f90: hfbtho DFT solver based on version 200d published in [80].

J. Phys. G: Nucl. Part. Phys. **48** (2021) 102001

Guide

- `hfodd_interface_5.f90`: interface between the HFBTHO and HFODD solvers. Contains the routine to transform the HFB matrix from the HO basis used in HFBTHO (hfodd) to the basis used in HFODD (HFBTHO).
- `hfodd_functional_4.f90`: interface to UNEDF functionals.
- `hfodd_mpiio_6.f90`: IO interface in MPI calculations. Contains the routine to read input data for parallel HFODD calculations.
- `hfodd_mpimanager_5.f90`: MPI toolkit. Defines the list of MPI tasks based on the data read in the parallel input file `hfodd_mpiio.d`.
- `hfodd_shell_5.f`: toolkit for the shell correction.
- `hfodd_SLsiz_4.f`: toolkit to incorporate ScaLAPACK capabilities.
- `hfodd_fission_9.f90`: toolkit for fission calculations. Contains several routines to compute fission fragment properties such as charge, mass, total energy, interaction energy; the routines needed to use a constraint on the number of particles in the neck; the routines used for the quantum localization method.
- `hfodd_pairs_2.f90`: toolkit for defining various derived types related to pairs, lists of pairs and lists of lists of pairs, as well as the routines needed to manipulate these objects.
- `hfodd_pnp_8.f90`: toolkit for particle number projection (not supported in the present version (v3.06h) of the code HFODD).
- `hfodd_fits_16.f90`: fit module. Allows the code HFODD to work as a routine in an external program.
- `hfodd_lipcorr_31.f90`: toolkit for the Lipkin method and Pfaffian overlap calculations.
- `hfodd_tgrad_18.f90`: toolkit for the three-body gradient terms.
- `hfodd_wigner_6.f90`: toolkit for the Wigner functions.

The FORTRAN source of version (v3.06h) of the code HFODD contains numerous undocumented and untested features that are under development. The user should not attempt to activate or reverse-engineer these features, because this can certainly lead to unpredictable behavior of the code and even damage to the computer hard drive.

Acknowledgments

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://webfiles.york.ac.uk/HFODD/.

ORCID iDs

J Dobaczewski ⬥ https://orcid.org/0000-0002-4158-3770
P Bączyk ⬥ https://orcid.org/0000-0003-4356-7982
M Bender ⬥ https://orcid.org/0000-0001-8707-3410
K Bennaceur ⬥ https://orcid.org/0000-0002-6722-491X
J Bonnard ⬥ https://orcid.org/0000-0002-5164-3343
A Idini ⬥ https://orcid.org/0000-0001-7624-8975
M Konieczka ⬥ https://orcid.org/0000-0003-2325-3767
M Kortelainen ⬥ https://orcid.org/0000-0001-6244-764X
L Próchniak ⬥ https://orcid.org/0000-0002-6577-6438
W Satuła ⬥ https://orcid.org/0000-0003-0203-3773

References

[1] Dobaczewski J and Dudek J 1997 *Comput. Phys. Commun.* **102** 166–82
[2] Dobaczewski J and Dudek J 1997 *Comput. Phys. Commun.* **102** 183–209
[3] Dobaczewski J and Dudek J 2000 *Comput. Phys. Commun.* **131** 164–86
[4] Dobaczewski J and Olbratowski P 2004 *Comput. Phys. Commun.* **158** 158–91
[5] Dobaczewski J and Olbratowski P 2005 *Comput. Phys. Commun.* **167** 214–6
[6] Dobaczewski J *et al* 2009 *Comput. Phys. Commun.* **180** 2361–91
[7] Schunck N, Dobaczewski J, McDonnell J, Satuła W, Sheikh J A, Staszczak A, Stoitsov M and Toivanen P 2012 *Comput. Phys. Commun.* **183** 166–92
[8] Schunck N *et al* 2017 *Comput. Phys. Commun.* **216** 145–74
[9] Gogny D 1975 *Nucl. Phys.* A **237** 399–418
[10] Koepf W and Ring P 1989 *Nucl. Phys.* A **493** 61–82
[11] Dobaczewski J *et al* 2009 hfodd (v2.40h) user's guide (arXiv:0909.3626)
[12] Dobaczewski J *et al* 2021 hfodd (v3.06h) User's Guide to be submitted to arXiv, online only publication
[13] Skyrme T H R 1959 *Nucl. Phys.* **9** 615–34
[14] Vautherin D and Brink D M 1972 *Phys. Rev.* C **5** 626–47
[15] Anguiano M, Egido J L and Robledo L M 2001 *Nucl. Phys.* A **696** 467–93
[16] Robledo L M 2007 *Int. J. Mod. Phys.* E **16** 337–51
[17] Dobaczewski J, Stoitsov M V, Nazarewicz W and Reinhard P G 2007 *Phys. Rev.* C **76** 054315
[18] Lacroix D, Duguet T and Bender M 2009 *Phys. Rev.* C **79** 044318
[19] Bender M, Duguet T and Lacroix D 2009 *Phys. Rev.* C **79** 044319
[20] Satuła W and Dobaczewski J 2014 *Phys. Rev.* C **90** 054303
[21] Duguet T, Bender M, Bennaceur K, Lacroix D and Lesinski T 2009 *Phys. Rev.* C **79** 044320
[22] Sadoudi J, Bender M, Bennaceur K, Davesne D, Jodon R and Duguet T 2013 *Phys. Scr.* **T154** 014013
[23] Sadoudi J, Duguet T, Meyer J and Bender M 2013 *Phys. Rev.* C **88** 064326
[24] Beiner M, Flocard H, Van Giai N and Quentin P 1975 *Nucl. Phys.* A **238** 29–69
[25] Bally B, Avez B, Bender M and Heenen P-H 2014 *Phys. Rev. Lett.* **113** 162501
[26] Stancu F, Brink D M and Flocard H 1977 *Phys. Lett.* B **68** 108–12
[27] Perlińska E, Rohoziński S G, Dobaczewski J and Nazarewicz W 2004 *Phys. Rev.* C **69** 014316
[28] Lesinski T, Bender M, Bennaceur K, Duguet T and Meyer J 2007 *Phys. Rev.* C **76** 014312
[29] Zalewski M, Dobaczewski J, Satuła W and Werner T R 2008 *Phys. Rev.* C **77** 024316
[30] Bender M, Bennaceur K, Duguet T, Heenen P H, Lesinski T and Meyer J 2009 *Phys. Rev.* C **80** 064302
[31] Hellemans V, Heenen P H and Bender M 2012 *Phys. Rev.* C **85** 014326
[32] Dobaczewski J and Dudek J 1995 *Phys. Rev.* C **52** 1827–39
[33] Miller G A and van Oers W H T 1995 Charge independence and charge symmetry *Symmetries and Fundamental Interactions in Nuclei* ed W C Haxton and E M Henley (Singapore: World Scientific)

[34] Henley E M and Miller G A 1979 *Mesons in Nuclei* ed M Rho and D H Wilkinson (Amsterdam: North-Holland) p 407
[35] Nolen J A Jr and Schiffer J P 1969 *Annu. Rev. Nucl. Sci.* **19** 471
[36] Ormand W E and Brown B A 1989 *Nucl. Phys.* A **491** 1
[37] Colo G, Sagawa H, Van Giai N, Bortignon P F and Suzuki T 1998 *Phys. Rev.* C **57** 3049–54
[38] Brown B A, Richter W A and Lindsay R 2000 *Phys. Lett.* B **483** 49–54
[39] Kaneko K, Sun Y, Mizusaki T and Tazaki S 2013 *Phys. Rev. Lett.* **110** 172505
[40] Ormand W E, Brown B A and Hjorth-Jensen M 2017 *Phys. Rev.* C **96** 024323
[41] Bączyk P, Dobaczewski J, Konieczka M, Satuła W, Nakatsukasa T and Sato K 2018 *Phys. Lett.* B **778** 178–83
[42] Roca-Maza X, Colò G and Sagawa H 2018 *Phys. Rev. Lett.* **120** 202501
[43] Bączyk P, Satuła W, Dobaczewski J and Konieczka M 2019 *J. Phys. G: Nucl. Part. Phys.* **46** 03LT01
[44] Sato K, Dobaczewski J, Nakatsukasa T and Satuła W 2013 *Phys. Rev.* C **88** 061301
[45] Sheikh J A, Hinohara N, Dobaczewski J, Nakatsukasa T, Nazarewicz W and Sato K 2014 *Phys. Rev.* C **89** 054317
[46] Satuła W and Wyss R 2001 *Phys. Rev. Lett.* **86** 4488–91
[47] Satuła W and Wyss R 2001 *Phys. Rev. Lett.* **87** 052504
[48] Chabanat E, Bonche P, Haensel P, Meyer J and Schaeffer R 1998 *Nucl. Phys.* A **635** 231–56
[49] Bartel J, Quentin P, Brack M, Guet C and Håkansson H-B 1982 *Nucl. Phys.* A **386** 79–100
[50] Konieczka M, Bączyk P and Satuła W 2016 *Phys. Rev.* C **93** 042501
[51] Raimondi F, Bennaceur K and Dobaczewski J 2014 *J. Phys. G: Nucl. Part. Phys.* **41** 055112
[52] Dobaczewski J, Bennaceur K and Raimondi F 2012 *J. Phys. G: Nucl. Part. Phys.* **39** 125103
[53] Bennaceur K, Dobaczewski J and Raimondi F 2014 *EPJ Web Conf.* **66** 02031
[54] Davesne D, Pastore A and Navarro J 2013 *J. Phys. G: Nucl. Part. Phys.* **40** 095104
[55] Bennaceur K, Idini A, Dobaczewski J, Dobaczewski P, Kortelainen M and Raimondi F 2017 *J. Phys. G: Nucl. Part. Phys.* **44** 045106
[56] Girod M and Grammaticos B 1983 *Phys. Rev.* C **27** 2317–39
[57] Egido J L, Robledo L M and Chasman R R 1997 *Phys. Lett.* B **393** 13–8
[58] Gonzalez-Boquera C, Centelles M, Viñas X and Robledo L M 2021 *Phys. Rev.* C **103** 064313
[59] Tian Y, Ma Z Y and Ring P 2009 *Phys. Lett.* B **676** 44–50
[60] Nikšić T, Ring P, Vretenar D, Tian Y and Ma Z y 2010 *Phys. Rev.* C **81** 054318
[61] Nikšić T, Paar N, Vretenar D and Ring P 2014 *Comput. Phys. Commun.* **185** 1808–21
[62] Romero A M, Dobaczewski J and Pastore A 2020 *J. Phys.: Conf. Ser.* **1643** 012144
[63] Carlsson B, Dobaczewski J, Toivanen J and Veselý P 2010 *Comput. Phys. Commun.* **181** 1641–57
[64] Veselỳ P, Dobaczewski J, Michel N and Toivanen J 2011 Finite-range separable pairing interaction within new N3LO DFT approach *AIP Conf. Proc.* vol 1377 (American Institute of Physics) pp 456–8
[65] Onishi N and Yoshida S 1966 *Nucl. Phys.* **80** 367–76
[66] Robledo L M 2009 *Phys. Rev.* C **79** 021302
[67] Bertsch G F and Robledo L M 2012 *Phys. Rev. Lett.* **108** 042505
[68] Sheikh J, Dobaczewski J, Ring P, Robledo L and Yannouleas C 2019 Symmetry restoration in mean-field approaches (arxiv:1901.06992)
[69] Bally B and Bender M 2021 *Phys. Rev.* C **103** 024315
[70] Ring P and Schuck P 1980 *The Nuclear Many-Body Problem* (Berlin: Springer)
[71] Tajima N 2015 *Phys. Rev.* C **91** 014320
[72] Feng X M, Wang P, Yang W and Jin G R 2015 *Phys. Rev.* E **92** 043307
[73] Côté J and Pearson J M 1978 *Nucl. Phys.* A **304** 104–26
[74] Ryssens W, Hellemans V, Bender M and Heenen P-H 2015 *Comput. Phys. Commun.* **187** 175
[75] Dobaczewski J, Engel J, Kortelainen M and Becker P 2018 *Phys. Rev. Lett.* **121** 232501
[76] Dobaczewski J, Nazarewicz W and Reinhard P-G 2014 *J. Phys. G: Nucl. Part. Phys.* **41** 074001
[77] Dobaczewski J, Flocard H and Treiner J 1984 *Nucl. Phys.* A **422** 103–39
[78] Dobaczewski J, Borycki P J, Nazarewicz W and Stoitsov M 2005 *Eur. Phys. J.* A **25** 541–2
[79] Dobaczewski J and Nazarewicz W 2013 Hartree–Fock–Bogoliubov solution of the pairing Hamiltonian in finite nuclei *50 Years of Nuclear BCS* ed R A Broglia and V Zelevinsky (Singapore: World Scientific) p 40 arXiv:1909.13041
[80] Stoitsov M V, Schunck N, Kortelainen M, Michel N, Nam H, Olsen E, Sarich J and Wild S 2013 *Comput. Phys. Commun.* **184** 1592–604