

## 2. The Skyrme–Hartree–Fock Model of the Nuclear Ground State

*P.-G. Reinhard*

### 2.1 Introduction

Two decades ago, with the introduction of Skyrme forces [2.1], Hartree–Fock calculations became feasible in nuclear physics. Since then, they have been applied to a great variety of phenomena, including deformation properties, superheavy nuclei, vibrations, and heavy-ion collisions [2.2]. Nonetheless, their most straightforward application, the description of the ground state of spherical nuclei, remains a useful tool. It serves as the basis for many further applications in nuclear-structure physics; e.g., for studying refinements and variants of the force [2.3], for understanding electron-scattering data, for describing hyperons in nuclei, or for RPA vibrations of the ground-state. Thus, it is desirable to have a code optimized for speed. Such a code also provides a good example of the fast numerical techniques that are necessary for large scale applications.

The spherical Hartree–Fock code presented here now is more than 12 years old. It has been rewritten often to keep up with developing programming standards. We hope that it now represents a fairly modern FORTRAN style and that it is sufficiently commented. From the beginning the code was optimized for speed. For this purpose we have developed improved variants of the gradient iteration [2.4–2.7]. Finite-difference schemes with five-point precision are used for the wave functions. We find this the optimum choice in one-dimensional applications, although this decision may change for higher dimensions. The code has been applied intensively in least-squares fits for optimizing the Skyrme-force parameters [2.3], for calculating ground-state correlations [2.8], and for nuclear-excitation studies. Although we have omitted most of these branches, options, and extensions here, the present code is applicable to quite a body of nuclear-structure physics. We shall try to give a complete but short account of all necessary ingredients.

The chapter is outlined as follows. In Sect. 2.2, we describe Hartree–Fock theory with Skyrme-forces and the schematic treatment of pairing. In Sect. 2.3, we explain the numerical representation of wave functions and fields, and we discuss the shell ordering that defines the bookkeeping. In Sect. 2.4, we present the iteration schemes for solving the coupled equations. In Sect. 2.5, we explain the computation of the charge density and related quantities, and in Sect. 2.6 we give a short overview of the code structure, the input, and the installation.

## 2.2 Skyrme–Hartree–Fock Theory with Schematic Pairing

The Hartree–Fock equations and the pairing equations will be derived variationally from the total energy functional of the nucleus. Thus the whole model and all its variants can be presented by discussing the various contributions to the energy functional

$$E = E_{\text{Skyrme}} + E_{\text{Coulomb}} + E_{\text{pair}} - E_{\text{cm}} , \quad (2.1)$$

where  $E_{\text{Skyrme}}$ , the energy functional of the Skyrme force, is the leading part. The Coulomb energy,  $E_{\text{Coulomb}}$ , seems straightforward, but some approximations will be needed to achieve a simplicity similar to  $E_{\text{Skyrme}}$ . A schematic pairing is defined by  $E_{\text{pair}}$ , and finally a correction for the spurious center-of-mass motion of the mean field is subtracted with  $E_{\text{cm}}$ . We discuss each of these contributions in the following subsections.

### 2.2.1 The Skyrme Energy Functional

#### The Skyrme Force

The Skyrme force is an effective force for nuclear Hartree–Fock calculations that aims to parametrize the  $t$ -matrix for nucleon–nucleon scattering in the nuclear medium in a simple and efficient manner. It is a zero-range, density- and momentum-dependent force of the form

$$\begin{aligned} V_{\text{Skyrme}} = & t_0(1 + x_0 P_x) \delta(\mathbf{r}_i - \mathbf{r}_j) \\ & + \frac{1}{2} t_1(1 + x_1 P_x) \{ \mathbf{p}_{12}^2 \delta(\mathbf{r}_i - \mathbf{r}_j) + \delta(\mathbf{r}_i - \mathbf{r}_j) \mathbf{p}_{12}^2 \} \\ & + t_2(1 + x_2 P_x) \mathbf{p}_{12} \cdot \delta(\mathbf{r}_i - \mathbf{r}_j) \mathbf{p}_{12} \\ & + \frac{1}{6} t_3(1 + x_3 P_x) \rho^\alpha(\bar{\mathbf{r}}) \delta(\mathbf{r}_i - \mathbf{r}_j) \\ & + i t_4 \mathbf{p}_{12} \cdot \delta(\mathbf{r}_i - \mathbf{r}_j) (\boldsymbol{\sigma}_i + \boldsymbol{\sigma}_j) \times \mathbf{p}_{12} , \end{aligned} \quad (2.2)$$

where  $\mathbf{p}_{12} = \mathbf{p}_i - \mathbf{p}_j$  is the relative momentum,  $P_x$  the space exchange operator  $\mathbf{r}_i \leftrightarrow \mathbf{r}_j$ ,  $\boldsymbol{\sigma}$  the vector of Pauli spin matrices, and  $\bar{\mathbf{r}} = \frac{1}{2}(\mathbf{r}_i + \mathbf{r}_j)$ . The simplicity of the ansatz allows the expectation value of the energy for Slater determinants to be evaluated in terms of a few densities and currents:

$$\begin{aligned} \rho_q(\mathbf{r}) &= \sum_{\beta \in q} w_\beta \varphi_\beta(\mathbf{r})^\dagger \varphi_\beta(\mathbf{r}) \\ \mathbf{j}_q(\mathbf{r}) &= \frac{i}{2} \sum_{\beta \in q} w_\beta [\nabla \varphi_\beta(\mathbf{r})^\dagger \varphi_\beta(\mathbf{r}) - \varphi_\beta(\mathbf{r})^\dagger \nabla \varphi_\beta(\mathbf{r})] \\ \tau_q(\mathbf{r}) &= \sum_{\beta \in q} w_\beta \nabla \varphi_\beta(\mathbf{r})^\dagger \cdot \nabla \varphi_\beta(\mathbf{r}), \\ \nabla \mathbf{J}_q(\mathbf{r}) &= -i \sum_{\beta \in q} w_\beta \nabla \varphi_\beta(\mathbf{r})^\dagger \cdot \nabla \times \boldsymbol{\sigma} \varphi_\beta(\mathbf{r}), \end{aligned} \quad (2.3)$$

where  $\varphi_\beta$  is the single-particle wave function of state  $\beta$ , and the isospin label  $q$  runs over  $q \in \{\text{pr}, \text{ne}\}$  (pr = proton and ne = neutron). The occupation probability of the state  $\beta$  is denoted by  $w_\beta$ . Completely filled shells have  $w_\beta = 1$ , but fractional occupancies occur for nonmagic nuclei; these are determined by the pairing scheme discussed in Sect. 2.2.3.

### The Energy in Spherical Representation

In the following we restrict consideration to the (stationary) ground state of spherical nuclei. The single-particle wave functions can then be separated as

$$\varphi_\beta(\mathbf{r}) = \frac{R_\beta(r)}{r} \mathcal{Y}_{j_\beta l_\beta m_\beta}(\theta, \phi). \quad (2.4)$$

The functions  $\mathcal{Y}_{j_\beta l_\beta m_\beta}$  are spinor spherical harmonics [2.9]. The radial wave functions  $R_\beta$  are independent of the  $m_\beta$  quantum number. The factor  $r^{-1}$  has been separated to simplify the overlap integrals to simple  $r$ -integration,

$$\langle \varphi_\beta | \varphi'_\beta \rangle = \delta_{j_\beta j'_\beta} \delta_{l_\beta l'_\beta} \delta_{m_\beta m'_\beta} \int_0^\infty dr R_\beta(r) R'_\beta(r), \quad (2.5)$$

and to simplify the kinetic energy in the radial Schrödinger equation to a second derivative. All shells are assumed to be filled equally over the  $m_\beta$ -degeneracy, so that the densities become spherical. Altogether the resulting Skyrme energy functional becomes

$$\begin{aligned} E_{\text{Skyrme}} = & 4\pi \int_0^\infty dr r^2 \left\{ \frac{\hbar^2}{2m} \tau + \frac{1}{2} t_0 (1 + \frac{1}{2} x_0) \rho^2 - \frac{1}{2} t_0 (\frac{1}{2} + x_0) \sum_q \rho_q^2 \right. \\ & + \frac{1}{12} t_3 (1 + \frac{1}{2} x_3) \rho^{\alpha+2} - \frac{1}{12} t_3 (\frac{1}{2} + x_3) \rho^\alpha \sum_q \rho_q^2 \\ & + \frac{1}{4} [t_1 (1 + \frac{1}{2} x_1) + t_2 (1 + \frac{1}{2} x_2)] \rho \tau \\ & - \frac{1}{4} [t_1 (\frac{1}{2} + x_1) - t_2 (\frac{1}{2} + x_2)] \sum_q \rho_q \tau_q \\ & - \frac{1}{16} [3t_1 (1 + \frac{1}{2} x_1) - t_2 (1 + \frac{1}{2} x_2)] \rho \nabla^2 \rho \\ & + \frac{1}{16} [3t_1 (1 + \frac{1}{2} x_1) + t_2 (1 + \frac{1}{2} x_2)] \sum_q \rho_q \nabla^2 \rho_q \\ & \left. - \frac{1}{2} t_4 [\rho \nabla \mathbf{J} + \sum_q \rho_q \nabla \mathbf{J}_q] \right\}, \quad (2.6) \end{aligned}$$

where  $\nabla^2 = \partial_r^2 + \frac{2}{r} \partial_r$  and  $\partial_r$  is shorthand for  $\frac{\partial}{\partial r}$ . Note that the current  $\mathbf{j}$  vanishes for stationary states. The densities in spherical representation are

$$\rho_q(r) = \sum_{n_\beta j_\beta l_\beta} w_\beta \frac{2j_\beta + 1}{4\pi} \left( \frac{R_\beta}{r} \right)^2,$$

$$\begin{aligned}
\tau_q(r) &= \sum_{n_\beta j_\beta l_\beta} w_\beta \frac{2j_\beta + 1}{4\pi} \left[ \left( \partial_r \frac{R_\beta}{r} \right)^2 + \frac{l(l+1)}{r^2} \left( \frac{R_\beta}{r} \right)^2 \right], \\
\nabla \mathbf{J}_q(r) &= \left( \partial_r + \frac{2}{r} \right) J_q(r), \\
J_q(r) &= \sum_{n_\beta j_\beta l_\beta} w_\beta \frac{2j_\beta + 1}{4\pi} [j_\beta(j_\beta + 1) - l_\beta(l_\beta + 1) - \frac{3}{4}] \frac{2}{r} \left( \frac{R_\beta}{r} \right)^2,
\end{aligned} \tag{2.7}$$

where the occupation weights  $w_\beta$  are also independent of  $m_\beta$ . The densities without an isospin label in (2.6) represent total densities, summed over both species:

$$\rho = \rho_{\text{pr}} + \rho_{\text{ne}}, \quad \tau = \tau_{\text{pr}} + \tau_{\text{ne}}, \quad \nabla \mathbf{J} = \nabla \mathbf{J}_{\text{pr}} + \nabla \mathbf{J}_{\text{ne}}$$

### The Hartree–Fock Equations

The Hartree–Fock equations for the radial wave functions  $R_\beta$  are obtained by varying the energy functional with respect to  $R_\beta$  under the constraint that the wave functions  $R_\beta$  be orthonormal. This yields the single-particle Schrödinger-like equation

$$h_q R_\beta = \epsilon_\beta R_\beta \tag{2.8}$$

with the mean-field Hamiltonian

$$h_q = \partial_r \mathcal{B}_q \partial_r + U_q + U_{ls,q} \mathbf{l} \cdot \boldsymbol{\sigma}, \tag{2.9}$$

where

$$\begin{aligned}
\mathcal{B}_q &= \frac{\hbar^2}{2m_q} + \frac{1}{8} [t_1(1 + \frac{1}{2}x_1) + t_2(1 + \frac{1}{2}x_2)]\rho \\
&\quad - \frac{1}{8} [t_1(\frac{1}{2} + x_1) - t_2(\frac{1}{2} + x_2)]\rho_q,
\end{aligned} \tag{2.10}$$

$$\begin{aligned}
U_q &= t_0(1 + \frac{1}{2}x_0)\rho - t_0(\frac{1}{2} + x_0)\rho_q \\
&\quad + \frac{1}{12}t_3\rho^\alpha \left[ (2 + \alpha)(1 + \frac{1}{2}x_3)\rho - 2(\frac{1}{2} + x_3)\rho_q \right. \\
&\quad \left. - \alpha(\frac{1}{2} + x_3) \frac{\rho_{\text{pr}}^2 + \rho_{\text{ne}}^2}{\rho} \right] \\
&\quad + \frac{1}{4} [t_1(1 + \frac{1}{2}x_1) + t_2(1 + \frac{1}{2}x_2)]\tau \\
&\quad - \frac{1}{4} [t_1(\frac{1}{2} + x_1) - t_2(\frac{1}{2} + x_2)]\tau_q \\
&\quad - \frac{1}{8} [3t_1(1 + \frac{1}{2}x_1) - t_2(1 + \frac{1}{2}x_2)]\Delta\rho \\
&\quad + \frac{1}{8} [3t_1(\frac{1}{2} + x_1) + t_2(\frac{1}{2} + x_2)]\Delta\rho_q \\
&\quad - \frac{1}{2}t_4(\nabla \mathbf{J} + \nabla \mathbf{J}_q) + U_{\text{Coul}},
\end{aligned} \tag{2.11}$$

$$U_{ls,q} = \frac{1}{4}t_4(\rho + \rho_q) + \frac{1}{8}(t_1 - t_2)J_q - \frac{1}{8}(x_1t_1 + x_2t_2)J, \tag{2.12}$$

and where the Coulomb contribution to the potential  $U_{\text{Coul}}$  will be given in the next subsection. The densities  $\rho$ ,  $\tau$ , and  $\nabla \mathbf{J}$  are given in (2.7). Note that (2.8) is nonlinear in the wave functions  $R_\beta$  via the density-dependent mean-field Hamiltonian (2.9–2.12).

## The Rearrangement Energy

Once the solution to the Hartree–Fock equations (2.8–2.12) has been found (possibly with pairing for the  $w_\beta$ ) one could evaluate the energy of the system straightforwardly from the functional (2.6). However, most of the energy information is already contained in the single-particle energies  $\epsilon_\beta$ . Thus, it is simpler to compute the energy via

$$E_{\text{Skyrme}} + E_{\text{Coulomb}} = \frac{1}{2} \left( E_{\text{kin}} + \sum_{\beta} w_{\beta} \epsilon_{\beta} \right) + E_{\text{rearr}} \quad (2.13)$$

$$E_{\text{kin}} = 4\pi \sum_q \int_0^\infty dr r^2 \frac{\hbar}{2m_q} \tau_q \quad (2.14)$$

$$E_{\text{rearr}} = -4\pi \int_0^\infty dr r^2 \frac{\alpha}{24} t_3 \rho^\alpha \left[ \left(1 + \frac{1}{2} x_3\right) \rho^2 - \left(\frac{1}{2} + x_3\right) (\rho_{\text{pr}}^2 + \rho_{\text{ne}}^2) \right] + E_{\text{Coul, rearr}}. \quad (2.15)$$

The combination of “half single-particle energies and half kinetic energy” is well known from standard Hartree–Fock schemes and automatically accounts for all two-body forces in the ansatz. The density-dependent term ( $\propto t_3$ ) additionally requires the so-called rearrangement energy,  $E_{\text{rearr}}$ . The density-dependent approximation to the Coulomb exchange also contributes to the rearrangement energy as discussed in the next subsection. The final energy is computed in the code according to (2.13–2.15).

### 2.2.2 The Coulomb Energy

The Coulomb interaction is a well-known piece of the nuclear interaction. However, its infinite range makes it very time consuming to evaluate the exchange part exactly, and it is unwise to spend most of the computing time on a small contribution. Therefore the Coulomb-exchange part is treated in the so-called Slater approximation, and we obtain for the Coulomb energy

$$E_{\text{Coul}} = \frac{1}{2} e^2 \int d^3r d^3r' \rho_C(\mathbf{r}) \frac{1}{|\mathbf{r} - \mathbf{r}'|} \rho_C(\mathbf{r}') + E_{\text{Coul, exch}} \quad (2.16)$$

$$E_{\text{Coul, exch}} = -\frac{3}{4} \left( \frac{3}{\pi} \right)^{1/3} 4\pi \int_0^\infty dr r^2 \rho_{\text{pr}}^{4/3}. \quad (2.17)$$

The contribution to the Hartree–Fock potential is easily computed by variation:

$$U_{\text{Coul}} = U_{\text{Coul, dir}} + U_{\text{Coul, exch}}, \quad (2.18)$$

$$-\Delta U_{\text{Coul, dir}} = 4\pi e^2 \rho_C, \quad (2.19)$$

$$U_{\text{Coul, exch}} = -\left( \frac{3}{\pi} \right)^{1/3} \rho_{\text{pr}}^{1/3}, \quad (2.20)$$

where the direct part is determined from solving the radial Poisson equation (2.19) rather than by integration. The density-dependent approach to the exchange term requires a contribution to the rearrangement energy

$$E_{\text{Coul, rearr}} = \frac{1}{4} \left( \frac{3}{\pi} \right)^{1/3} 4\pi \int_0^\infty dr r^2 \rho_{\text{pr}}^{4/3}. \quad (2.21)$$

In the direct term, the nuclear charge distribution  $\rho_C$  should enter. It should include folding with the finite size of the proton, but this folding again is a bit slow, and one would like to avoid it during the many repeated steps of an iteration. To a very good approximation one can replace  $\rho_C \rightarrow \rho_{\text{pr}}$  in the Coulomb energy (2.16). This approximation is toggled by IFRHOC in the code (see Sect. 2.6.2).

There are many applications in the literature which do not include the Coulomb-exchange part at all. We allow for this in the code with the switch IFEX, as explained in Sect. 2.6.2.

### 2.2.3 Pairing

A schematic pairing force is introduced through the energy functional

$$E_{\text{pair}} = - \sum_q G_q \left[ \sum_{\beta \in q} \sqrt{w_\beta(1 - w_\beta)} \right]^2, \quad (2.22)$$

where the pairing matrix elements  $G_q$  are constant within each species,  $q \in \{\text{pr}, \text{ne}\}$ . The BCS equations for the pairing weights  $w_\beta$  are obtained by varying the energy functional (2.1) with respect to  $w_\beta$ . This yields the standard BCS equations for the case of a constant pairing force. The occupation weights become

$$w_\beta = \frac{1}{2} \left( 1 - \frac{\epsilon_\beta - \epsilon_{F,q}}{\sqrt{(\epsilon_\beta - \epsilon_{F,q})^2 + \Delta_q^2}} \right). \quad (2.23)$$

The two remaining parameters, the pairing gap  $\Delta_q$  and the Fermi energy  $\epsilon_{F,q}$ , are determined by simultaneous solution of the gap equation and the particle-number condition:

$$\frac{\Delta_q}{G_q} = \sum_{\beta \in q} \sqrt{w_\beta(1 - w_\beta)}, \quad (2.24)$$

$$A_q = \sum_{\beta \in q} w_\beta, \quad (2.25)$$

where  $A_q$  is the desired number of protons ( $q = \text{pr}$ ) or neutrons ( $q = \text{ne}$ ). In the following we call this treatment the *constant-force approach*.

Often one simplifies the pairing treatment further by parametrizing the pairing gap  $\Delta_q$  directly [2.10]. In this case only the Fermi energy  $\epsilon_{F,q}$  has

to be adjusted such that the particle-number condition (2.25) is matched, and the gap equation (2.24) need not be solved. It does, however, serve to compute  $G_q$ , which then is needed to compute the pairing energy according to (2.22). In the following we call this procedure the *constant-gap approach*. A commonly accepted parametrization of the gap is

$$\Delta_q = (11.2 \text{ MeV})/\sqrt{A}, \quad (2.26)$$

where  $A = A_{\text{pr}} + A_{\text{ne}}$  is the total nucleon number of a nucleons.

Both variants of the pairing treatment are implemented in the code. They are switched by the sign of the pairing strength in the input data; for details see Sect. 2.6.2.

## 2.2.4 The Center-of-Mass Correction

The exact nuclear ground state should be a state with total momentum zero. The mean field, however, localizes the nucleus and thus breaks translational invariance [2.11], so that the center-of-mass of the whole nucleus oscillates in the mean-field. In principle, one should project a state with good total momentum zero out of the given mean-field state. A simple and reliable substitute for the projection is to subtract the zero-point energy of the nearly harmonic oscillations of the center of mass [2.7]. The correction is simply

$$\begin{aligned} E_{\text{cm}} &= \frac{\langle P_{\text{cm}}^2 \rangle}{2Am}, \\ \langle P_{\text{cm}}^2 \rangle &= \sum_{\beta} w_{\beta} \langle \varphi_{\beta} | \hat{p}^2 | \varphi_{\beta} \rangle \\ &\quad - \sum_{\alpha, \beta} (w_{\alpha} w_{\beta} + \sqrt{w_{\alpha}(1-w_{\alpha})w_{\beta}(1-w_{\beta})}) | \langle \varphi_{\alpha} | \hat{p} | \varphi_{\beta} \rangle |^2, \end{aligned} \quad (2.27)$$

where  $P_{\text{cm}} = \sum_i \hat{p}_i$  is the total momentum operator,  $A$  the nucleon number, and  $m$  the average nucleon mass. Note that this means evaluating a two-body operator, which leads to a double-sum over single-particle states. It would be too time consuming to evaluate such a small but involved correction at each iteration. Thus the center-of-mass correction (2.27) is computed *after variation*, and does not contribute to the Hartree–Fock equations (2.8–2.12). The correction is subtracted only a posteriori.

An even simpler approach to the center-of-mass correction was used in early applications. One can take the one-body part of (2.27),  $P_{\text{cm}}^2 \approx \sum_i \hat{p}_i^2$ , and thus express the center-of-mass correction as a modification of the nucleon mass

$$\frac{\hbar^2}{2m} \rightarrow \frac{\hbar^2}{2m} \left( 1 - \frac{1}{A} \right). \quad (2.28)$$

Both variants are implemented in the code. They are switched by IFZPE; see Sect. 2.6.2.

## 2.3 Representation of Wave Functions and Fields

### 2.3.1 The Radial Grid

After the spherical separation (2.4), the radial wave functions  $R_\beta$  and the densities and fields  $\rho_q, \tau_q$ , etc. depend only on the radial distance  $r$ . We represent them on an equidistant radial grid

$$f(r) \equiv f_i = f(r_i), \quad r_i = (i-1)\Delta_r, \quad i = 1, \dots, N_r, \quad (2.29)$$

where  $f$  stands for any one of the wave functions or fields.

The radial integration is done by the simple trapezoidal rule:

$$\int_0^\infty dr F(r) \approx \Delta_r \left( \frac{1}{2} F_1 + \sum_{i=2}^{N_r-1} F_i + \frac{1}{2} F_{N_r} \right) = \Delta_r \sum_{i=2}^{N_r} F_i, \quad (2.30)$$

where we have used the fact that all integrands that occur in spherical calculations vanish at the origin ( $i = 1$ ) and are negligible at the boundary ( $i = N_r$ ). We have found out in tests that the simple trapezoidal rule is very precise for integrands that vanish at the boundaries, comparable to or better than a five-point integration formula.

For the derivatives, we use five point-precision throughout. This implies

$$(\partial_r f)_i = \frac{1}{\Delta_r} \left( -\frac{1}{12} f_{i+2} + \frac{2}{3} f_{i+1} - \frac{2}{3} f_{i-1} + \frac{1}{12} f_{i-2} \right), \quad (2.31)$$

$$(\partial_r^2 f)_i = \frac{1}{\Delta_r^2} \left( -\frac{1}{12} f_{i+2} + \frac{4}{3} f_{i+1} - \frac{5}{2} f_i + \frac{4}{3} f_{i-1} - \frac{1}{12} f_{i-2} \right). \quad (2.32)$$

These formulas cease to work at the boundaries of the grid because they call the function  $f$  at points outside the grid, e.g.  $f_0$ . Fortunately, all functions have well-defined reflection symmetry about the lower boundary  $r = 0$ . The radial wave functions are  $R_\beta \propto r^{l_\beta+1} P(r^2)$ , where  $P$  can be any polynomial, and the densities and potentials are even, i.e.  $\propto P(r^2)$ . Thus we can define lower grid values by  $f_0 = \Pi f_2$  and  $f_{-i} = \Pi f_{i+2}$  with  $\Pi$  the radial parity of the field  $f$ . The solution of the boundary problem is less clean at the upper bound, but all wave functions and fields should be very small there. Thus, one may very well switch to a less precise treatment. We use three-point precision for the derivatives at the penultimate point,  $i = N_r - 1$ , and extrapolate to the upper boundary by  $\mathcal{F}_{N_r} = 2\mathcal{F}_{N_r-1} - \mathcal{F}_{N_r-2}$  with  $\mathcal{F}$  standing for  $\partial_r f$  or  $\partial_r^2 f$ . A special treatment is used for the second derivative in the inversion of the mean-field Hamiltonian, as will be discussed in connection with the inverse gradient step in Sect. 2.4.

### 2.3.2 Storage of Densities and Potentials

All densities ( $\rho_q, \tau_q, J_q$ ) and potentials ( $\mathcal{B}_q, U_q, U_{ls,q}$ ) occur once for protons ( $q = \text{pr}$ ) and once for neutrons ( $q = \text{ne}$ ). We define one linear array for each field, e.g. RHO for the density  $\rho$ , and stack the neutron array above the proton array, so that



$$\begin{aligned}\rho_{\text{pr}} &\leftrightarrow \text{RHO}(1, \dots, N_r), \\ \rho_{\text{ne}} &\leftrightarrow \text{RHO}(N_r + 1, \dots, 2N_r),\end{aligned}\tag{2.33}$$

and similarly for all other fields.

### 2.3.3 Shell Ordering

The quantum number  $\beta$  for the single-particle states is, in fact, a composite quantum number,  $\beta = (q, n_\beta, j_\beta, l_\beta, m_\beta)$ . While the  $m_\beta$  is irrelevant for radial properties, there still remains a multiplet of four quantum numbers, which run over very different values. The typical arrangement in nuclear-shell models is described very well by the ordering of shells as given in the harmonic-oscillator shell model [2.12], as shown in Table 2.1. It is advantageous to pack the total angular momentum  $j$  and the orbital angular momentum  $l$  into one combined quantum number

$$\text{JS} = j + l + \frac{1}{2},\tag{2.34}$$

which in turn allows retrieving the separate angular momenta as  $l = \left\lceil \frac{\text{JS}}{2} \right\rceil$  and  $j = \left\lfloor \frac{\text{JS}+1}{2} \right\rfloor - \frac{1}{2}$ , where [...] means the “integer part”.

We adopt the oscillator shell ordering as given in Table 2.1 for our bookkeeping of states in the code. The single-particle states  $\beta$  are labeled by the shell number NSHL, and the single-particle energies  $\epsilon_\beta$  and the occupation weights  $w_\beta$  are stored in a linear array according to this ordering. Again, we store proton and neutron information in *one* linear array, and the array of neutron information is stacked above the proton information.

There are two ways to loop through the single-particle states in the code. First, one can count along the shell number NSHL. The associated angular momentum JS is then obtained from the bookkeeping field JSP(NSHL). Alternatively, one can do a double loop over JS and the radial quantum number  $n$ . In this case one needs the reverse pointer that produces the NSHL for given JS and  $n$ . This is provided in the pointer field NPLA, which is, in fact, a pointer for the wave functions. We refer to the next subsection for more details.

### 2.3.4 Storage of Wave Functions

The radial wave functions  $R_\beta$  are stored in one linear array WF in portions of length  $N_r$ . They are stacked above one another according to the oscillator shell ordering of Table 2.1. The neutron wave functions follow on top of the proton wave functions, so that the arrangement becomes

**Table 2.1.** Typical shell ordering according to the nuclear oscillator shell model. This holds for protons as well as for neutrons. Shell closures are indicated by a horizontal line. The ordering of states within two shell closures may differ from model to model. For simple bookkeeping we have chosen a subordering by decreasing the total angular momentum  $j$ . The “degeneracy” is the number of nucleons fitting in the shell, and “accumulated A” denotes the number of nucleons up to and including the listed shell, thus immediately displaying the magic nucleon numbers at the shell closures.

NSHL	$n$	$j$	$l$	JS	name	degeneracy	accumulated A
1	1	1/2	0	1	$1s_{\frac{1}{2}}$	2	2
2	1	3/2	1	3	$1p_{\frac{3}{2}}$	4	6
3	1	1/2	1	2	$1p_{\frac{1}{2}}$	2	8
4	1	5/2	2	5	$1d_{\frac{5}{2}}$	6	14
5	1	3/2	2	4	$1d_{\frac{3}{2}}$	4	18
6	2	1/2	0	1	$2s_{\frac{1}{2}}$	2	20
7	1	7/2	3	7	$1f_{\frac{7}{2}}$	8	28
8	1	5/2	3	6	$1f_{\frac{5}{2}}$	6	34
9	2	3/2	1	3	$2p_{\frac{3}{2}}$	4	38
10	2	1/2	1	2	$2p_{\frac{1}{2}}$	2	40
11	1	9/2	4	9	$1g_{\frac{9}{2}}$	10	50
12	1	7/2	4	8	$1g_{\frac{7}{2}}$	8	58
13	2	5/2	2	5	$2d_{\frac{5}{2}}$	6	64
14	2	3/2	2	4	$2d_{\frac{3}{2}}$	4	68
15	3	1/2	0	1	$3s_{\frac{1}{2}}$	2	70
16	1	11/2	5	11	$1h_{\frac{11}{2}}$	12	82
17	1	9/2	5	10	$1h_{\frac{9}{2}}$	10	92
18	2	7/2	3	7	$2f_{\frac{7}{2}}$	8	100
19	2	5/2	3	6	$2f_{\frac{5}{2}}$	6	106
20	3	3/2	1	3	$3p_{\frac{3}{2}}$	4	110
21	3	1/2	1	2	$3p_{\frac{1}{2}}$	2	112
22	1	13/2	6	13	$1i_{\frac{13}{2}}$	14	126
23	1	11/2	6	12	$1i_{\frac{11}{2}}$	12	138
24	2	9/2	4	9	$2g_{\frac{9}{2}}$	10	148
25	2	7/2	4	8	$2g_{\frac{7}{2}}$	8	156
26	3	5/2	2	5	$4d_{\frac{5}{2}}$	6	162
27	3	3/2	2	4	$4d_{\frac{3}{2}}$	4	166
28	4	1/2	0	1	$5s_{\frac{1}{2}}$	2	168
29	1	15/2	7	15	$1j_{\frac{15}{2}}$	16	184

$$\begin{aligned}
R_{\text{pr},1s\frac{1}{2}} &\equiv \text{WF}(1, \dots, N_r) \\
R_{\text{pr},1p\frac{3}{2}} &\equiv \text{WF}(N_r + 1, \dots, 2N_r) \\
R_{\text{pr},1p\frac{1}{2}} &\equiv \text{WF}(2N_r + 1, \dots, 3N_r) \\
&\dots \quad \dots \quad \dots \\
R_{\text{ne},1s\frac{1}{2}} &\equiv \text{WF}(\mathcal{A} + 1, \dots, \mathcal{A} + N_r) \\
R_{\text{ne},1p\frac{3}{2}} &\equiv \text{WF}(\mathcal{A} + N_r + 1, \dots, \mathcal{A} + 2N_r) \\
&\dots \quad \dots
\end{aligned} \tag{2.35}$$

where  $\mathcal{A} = N_r M_{\text{pr}}$  is the offset for neutron storage and  $M_{\text{pr}} =$  “number of proton shells”. The first position in WF for a wave function for a given state  $(q, n, j, l)$  is stored in the pointer field NPLA. It can easily be constructed from the scheme (2.35) with the shell ordering of Table 2.1 as  $\text{NPLA} = N_r * \text{NSHL} + 1$  for protons and  $\text{NPLA} = \mathcal{A} + N_r * \text{NSHL} + 1$  for neutrons. Note that the pointer field can also be used to access the number of a state in the fields for the  $\epsilon_\beta$  and for the  $w_\beta$ : the integer part of  $\text{NPLA}/N_r$  returns just NSHL with the appropriate offset.

## 2.4 Iterative Solution of the Hartree–Fock Equations

### 2.4.1 The Gradient Iteration

The Hartree–Fock equations (2.8–12) are nonlinear in the wave functions, so that an iterative solution is needed. It is advisable then also to use an iterative scheme for solving the differential equation (2.8) because the intermediate solutions for the wave functions are superseded anyway. The simplest and most robust iteration scheme to find an eigenvalue of  $h$  is  $\varphi_\beta^{(n+1)} = h\varphi_\beta^{(n)}$ . This scheme will converge towards  $h_{\text{max}}$ , the maximum eigenvalue of  $h$ . However, we are interested rather in the opposite, the minimum eigenvalue  $h_{\text{min}}$ . To this end, we construct from  $h$  a step-operator  $\mathcal{S}$  such that  $h_{\text{min}}$  corresponds to  $\mathcal{S}_{\text{max}}$ . This yields the gradient iteration

$$\varphi_\beta^{(n+1)} = \mathcal{O}\{\mathcal{S}^{(n)}\varphi_\beta^{(n)}\}, \tag{2.36}$$

where  $\mathcal{O}$  denotes orthonormalization of all  $\varphi_\beta$  in ascending order of the single-particle states  $\beta$  and where the (simple) gradient step is

$$\mathcal{S}_{\text{grad}}^{(n)} = 1 - \delta h^{(n)}, \quad \text{where} \quad \delta < \frac{2}{h_{\text{max}}}. \tag{2.37}$$

The restriction on  $\delta$  guarantees that  $h_{\text{min}}$  produces  $\mathcal{S}_{\text{max}}$ . Note that  $h^{(n)}$  and subsequently  $\mathcal{S}_{\text{grad}}^{(n)}$  depend on the iteration because the Hartree–Fock equations are nonlinear equations in  $\varphi_\beta^{(n)}$ . The step (2.37) is called the (simple) gradient step because  $h\varphi_\beta$  is used to approach the goal and this is just the gradient of the potential-energy landscape,  $h\varphi_\beta = \partial E / \partial \varphi_\beta^+$ .

The gradient step is simple and robust, but can become extremely slow owing to large  $h_{\max}$  in a good numerical representation of the  $\varphi_\beta$ . For example with the kinetic energy in a finite-difference scheme with five-point precision we obtain  $h_{\max} \approx \frac{\hbar^2}{2m\Delta r^2}(\frac{16}{3} + l_{\max}(l_{\max} + 1))$ , which ranges from 2000 MeV to 5000 MeV in typical nuclear applications, where we are looking for  $h_{\min}$  in the range 10 – 50 MeV. Thus, one typically needs a few thousand iterations to obtain satisfactory precision from the simple gradient iteration.

### 2.4.2 The Damped Gradient Iteration

The problem with the large  $h_{\max}$  and the resulting slow convergence of the gradient step can easily be localized for the case of the nuclear shell model: it is the kinetic energy which rises to annoyingly large values of thousands of MeV, whereas the potential energy typically stays within 100 MeV. This suggests improving the step by damping the components with high kinetic energy and leads to the damped gradient step

$$\mathcal{S}^{(n)\text{damp}} = 1 - \delta \left( \frac{p^2}{2m} + E_0 \right)^{-1} (h^{(n)} - \langle \varphi_\beta^{(n)} | h^{(n)} | \varphi_\beta^{(n)} \rangle), \quad (2.38)$$

where  $\delta$  and  $E_0$  are numerical parameters to be adjusted for optimal speed and stability. Typical values are  $\delta \approx 1$  and  $E_0 \approx 40$  MeV, which is about the variation of the potential. The subtraction of the expectation value  $\langle h \rangle$  is a shortcut to project the step orthogonal to the solution already reached; it guarantees that the step decays properly as one approaches the solution.

The damped gradient step (2.38) is a substantial improvement over the simple gradient step. One usually obtains five-digit precision within about 30–50 iterations. The method has been applied successfully in many different problems with different representations of the fields (finite differences, fast Fourier, B-splines) and different dimensions (spherical 1D, axially symmetric 2D, full Cartesian 3D) [2.6, 7, 13]. For applications in more than one dimension, one can reduce the potentially time-consuming inversion of the kinetic energy by a separable approach,  $(1 + \frac{p^2}{2mE_0}) \approx \prod_{i=x,y,z} (1 + \frac{p_i^2}{2mE_0})$ . Thus one only needs successive inversions in one dimension, which makes the damped gradient an extremely fast scheme in higher dimensions.

### 2.4.3 The Inverse Gradient Step

In one-dimensional calculations, it is even conceivable to invert the full mean-field Hamiltonian  $h$ , that is, replace  $\frac{p^2}{2m}$  in the damped gradient step (2.38) by  $h$ . One finds that optimum convergence is achieved for parameter combinations  $\delta$  and  $E_0$  such that the step becomes a mere inversion of  $h$ . This leads to the inverse gradient step

$$\mathcal{S}_{\text{inv}}^{(n)} = \left( h^{(n)} - U_{\min}^{(n)} \right)^{-1}, \quad (2.39)$$

where  $U_{\min}^{(n)}$  is the minimum value of the mean-field potential.  $U_{\min}$  is needed to make the inverted operator positive definite. This step is even a bit faster than the damped gradient step. An additional advantage is that it requires fewer operations per iteration and that it is free of numerical parameters. It should be noted that the step may be made even faster by using  $\epsilon_{\beta}^{(n)} + \delta$  rather than  $U_{\min}^{(n)}$ . But this is potentially dangerous and can lead to instabilities.

The inverse gradient step (2.39) is our preferred step for spherical calculations of the nuclear ground state. We nevertheless provide both steps, the inverse gradient step and the damped gradient step, in the code. This gives the user the opportunity to get experience with both schemes. The switch for the steps in the code is the stepsize  $\delta = \text{X0DAMP}$ : the inverse gradient step (2.39) is used for  $\text{X0DAMP} = 0$  whereas a finite value switches to the damped gradient step (2.38).

The wave-function iteration with the inverse gradient step is so fast that the change in the potential fields gives rise to unwanted oscillations. We damp them by mixing only a fraction  $\text{ADDNEW}$  of the new potential  $U^{(n)}$  with the previous potential  $U^{(n-1)}$ .

#### 2.4.4 Efficient Inversion with Five-Point Precision

The five-point formula (2.32) for the second derivative makes the kinetic energy a sparse matrix with two off-diagonal lines. This could easily and quickly be inverted by a Gaussian elimination scheme. However, there is an even faster scheme [2.14]. We write the five-point formula for the second derivative as

$$\begin{aligned} \Delta_5 &= D_3^{-1} \Delta_3 , \\ (D_3 f)_i &= \frac{1}{12} f_{i+1} + \frac{5}{6} f_i + \frac{1}{12} f_{i-1} \\ (\Delta_3 f)_i &= f_{i+1} - 2f_i + f_{i-1} . \end{aligned} \tag{2.40}$$

The inversion is written as an inhomogenous linear equation. For the kinetic energy inversion (e.g.,  $\psi = (\frac{p^2}{2m} + E_0)^{-1} \varphi$ ), we write

$$\left( \frac{p^2}{2m} + E_0 \right) \psi = \varphi ,$$

to be solved for  $\psi$ . We insert the second derivative in the approach (2.40) and operate with  $D_3$  from left. Thus we have to solve

$$\left( -\frac{\hbar^2}{2m} \Delta_3 + E_0 D_3 \right) \varphi = D_3 \psi . \tag{2.41}$$

The matrix to be inverted is now sparse with only one off-diagonal line. This, of course, can be inverted even faster by a Gaussian elimination scheme. Thus we obtain five-point precision in nearly the same time as three-point precision.

### 2.4.5 Pairing Iteration

The pairing equations are solved by Newton’s tangential iteration. We illustrate it for the “constant-gap approach”, where we have to determine the Fermi energy  $\epsilon_F$  such that the particle-number condition (2.25) is met for a given set of single-particle energies  $\epsilon_\beta$ . There may be a deviation from the goal for some given Fermi energy  $\epsilon_{F,q}^{(n)}$ . We linearize the error and determine a correction  $\delta\epsilon_{F,q}$  by

$$A_q = \sum_{\beta \in q} w_\beta + \sum_{\beta \in q} \frac{\partial w_\beta}{\partial \epsilon_{F,q}} \delta\epsilon_{F,q} , \quad (2.42)$$

such that the new  $\epsilon_{F,q}^{(n+1)} = \epsilon_{F,q}^{(n)} + \delta\epsilon_{F,q}$  comes much closer to the proper Fermi energy. The derivative in (2.42) is determined numerically. As the tangential step has a tendency to overshoot, we stabilize it by reducing it to 90%. Thus, we iterate

$$\epsilon_{F,q}^{(n+1)} = \epsilon_{F,q}^{(n)} + 0.9 \frac{A_q - \sum_{\beta \in q} w_\beta}{\sum_{\beta \in q} \frac{\partial w_\beta}{\partial \epsilon_{F,q}}} . \quad (2.43)$$

This step still suffers in that it sometimes shoots completely out of a physically reasonable region. As a further safety measure, we limit the step in  $\epsilon_{F,q}$  to a maximum change of 3 MeV.

The iteration for the “constant-force approach” proceeds very similarly. We then have to compute two steps,  $\delta\epsilon_{F,q}$  and  $\delta\Delta_q$ , from a twofold variation of the two equations (2.24) and (2.25). The extension is obvious and will not be outlined in detail here.

The Hartree–Fock and pairing equations are coupled, and they are solved by simultaneous iteration of the wave functions and the occupation weights. The procedure is outlined in Sect. 2.6.

## 2.5 The Form Factor and Related Observables

### 2.5.1 Composing the Nuclear Charge Density

The nuclear charge density is a most useful observable for analyzing nuclear structure: it provides information about the nuclear shape and can be determined by clear-cut procedures from the cross section for elastic electron scattering [2.15]. To compute the observable charge density from the Hartree–Fock results, one has to take into account that the nucleons themselves have an intrinsic electromagnetic structure [2.16]. Thus one needs to fold the proton and neutron densities from the Hartree–Fock method with the intrinsic charge density of the nucleons. Folding becomes a simple product in Fourier space, so that we transform the densities to the so-called form factors

$$F_q(k) = 4\pi \int_0^\infty dr r^2 j_0(kr) \rho_q(r) , \quad (2.44)$$

**Table 2.2.** The coefficients of the isospin-coupled Sachs form factors of the nucleons; the  $b_i$  are given in  $\text{fm}^{-1}$ .

	$a_1$	$a_2$	$a_3$	$a_4$	$b_1$	$b_2$	$b_3$	$b_4$
$E, I = 0$	2.2907	-0.6777	-0.7923	0.1793	15.75	26.68	41.04	134.2
$E, I = 1$	0.3681	1.2263	-0.6316	0.0372	5.00	15.02	44.08	154.2
$M$	0.6940	0.7190	-0.4180	0.0050	8.50	15.02	44.08	355.4

where  $j_0$  is the spherical Bessel function of zeroth order. In fact, the form factor is closer to experiment because in the Born approximation it directly represents the amplitude for scattering at momentum transfer  $\hbar k$ . The charge form factor is then given by

$$F_C(k) = \sum_q [F_q(k)G_{E,q}(k) + F_{ls,q}(k)G_M(k)] \exp(\tfrac{1}{8}(\hbar k)^2/\langle P_{\text{cm}}^2 \rangle), \quad (2.45)$$

where  $F_{ls,q}$  is the form factor of the spin-orbit current  $\nabla \mathbf{J}$  (accounting for magnetic contributions to the charge density),  $G_{E,q}$  are the electric form factors of the nucleons,  $G_M$  is the magnetic form factor of the nucleons (assumed to be equal for both species), and the exponential factor takes into account an unfolding of the spurious vibrations of the nuclear center of mass in the harmonic approximation (the  $\langle P_{\text{cm}}^2 \rangle$  therein is the same as in the zero-point energy (2.27)).

The intrinsic nucleon form factors are taken from electron scattering on protons and deuterons. They are given as the so-called Sachs form factors for isospin 0 and 1 in the form of a dipole fit, [2.17]

$$G_s = \sum_{\nu=1}^4 \frac{a_{s,\nu}}{1 + k^2/b_{s,\nu}} \quad \text{for } s \in \{(E, I = 0), (E, I = 1), M\}. \quad (2.46)$$

The actual coefficients are displayed in Table 2.2. The isospin-coupled form factors are recoupled for our purposes to proton and neutron form factors, and the relativistic Darwin correction [2.16] has to be added, yielding

$$G_{E,pr/ne} = \tfrac{1}{2}(G_{E,I=0} \pm G_{E,I=1}) / \sqrt{1 + \frac{\hbar^2 k^2}{2m_{pr/ne}}}. \quad (2.47)$$

Note that the nucleon structure is taken into account only approximately because we fold with the free form factors of the nucleons thus neglecting medium distortion and off-shell effects of the nucleon.

The charge density is obtained from the charge form factor by the inverse Fourier-Bessel transform

$$\rho_C(r) = \frac{1}{2\pi^2} \int dk k^2 j_0(kr) F_C(k). \quad (2.48)$$

Other information can be drawn directly from the form factor, as we will see in the next subsection.

### 2.5.2 Radii and Surface Thickness

Most of the information that is contained in the form factor at low  $q$  can be described by two parameters: the diffraction radius  $R$  and the surface thickness  $\sigma$  [2.18]. The diffraction radius is determined from the first zero of the form factor

$$R = 4.493/k_0^{(1)} \quad \text{where} \quad F_C(k_0^{(1)}) = 0. \quad (2.49)$$

It parametrizes the overall diffraction pattern, which resembles that of a box with radius  $R$ . The diffraction radius is the box equivalent radius.

Upon closer inspection one realises that the nuclear form factor decreases more rapidly than the box form factor. This is due to the finite surface thickness of nuclei. If one models the nuclear surface by folding the box distribution with a Gaussian,  $\exp(-\frac{1}{2}q^2\sigma^2)$ , one can determine the surface parameter  $\sigma$  by comparing the heights of the first maxima of the box-equivalent and the Hartree-Fock results:

$$\sigma = \frac{2}{k_m} \log(F_{box}(k_m)/F_C(k_m)) \quad \text{with} \quad F_{box}(k) = 3 \frac{j_1(k_m R)}{k_m R}, \quad (2.50)$$

where  $k_m = 5.6/R$  is the momentum where  $F_{box}$  has its first maximum.

Alternative information on the overall nuclear size is given by the root-mean-square (r.m.s.) radius  $r$ , which is determined from the curvature of the form factor at  $k \rightarrow 0$  as

$$r = 3 \left( \frac{d^2}{dk^2} F_C(k) \right)_{k=0} / F_C(0). \quad (2.51)$$

It does not carry much new information because it can be related to  $R$  and  $\sigma$  approximately as  $r = \sqrt{R^2 + 3\sigma^2}$  [2.18]. The parametrization in terms of  $R$  and  $\sigma$  is preferable because the diffraction radius  $R$  is very stable against correlations and depends smoothly on the particle number  $A$ , whereas the surface thickness  $\sigma$  is sensitive to shell effects and correlations from low energy modes.

### 2.5.3 Handling Form Factors on a Grid

In practice, the nucleon densities are given on a finite radial grid (see Sect. 2.3). All information on densities is retained if one computes and stores the form factor on the reciprocal lattice,

$$F(k) \equiv F(k_j) \quad , \quad k_j = (j-1) \frac{\pi}{\Delta_r N_r} \quad , \quad j = 1, \dots, N_k, \quad (2.52)$$

where usually  $N_k = N_r$ . In practice, one chooses the grid in coordinate space finer than necessary for a grid in momentum space since the finite difference formulas for the kinetic energy are inferior to a Fourier representation. Thus, one can easily drop the high Fourier components and choose  $N_k \approx \frac{1}{2}N_r$ .



The Fourier-Bessel forward and inverse transformations (2.44) and (2.48) are evaluated by the simple trapezoidal rule, which corresponds to the usual method of Fourier transformations between reciprocal lattices.

The form factor is thus given on a grid  $\{k_j, j = 1, \dots, N_k\}$ . But all three form parameters defined in the previous subsection require access to the form factor  $F_C$  at any value of  $k$ . We compute these intermediate values by Fourier-Bessel interpolation

$$F(k) = \frac{\sin(kR_{\max})}{kR_{\max}} \sum_{j=1}^{N_k} (-)^{j+1} \left[ 1 - \frac{kR_{\max}}{j\pi} \right]^{-1} F(k_j), \quad (2.53)$$

where  $R_{\max} = N_r \Delta_r$  is the length of the coordinate grid. The interpolation runs into problems with round-off errors near  $k \approx j\pi/R_{\max}$ ; we switch to a Taylor expansion in powers of  $k - k_j$  near those critical points.

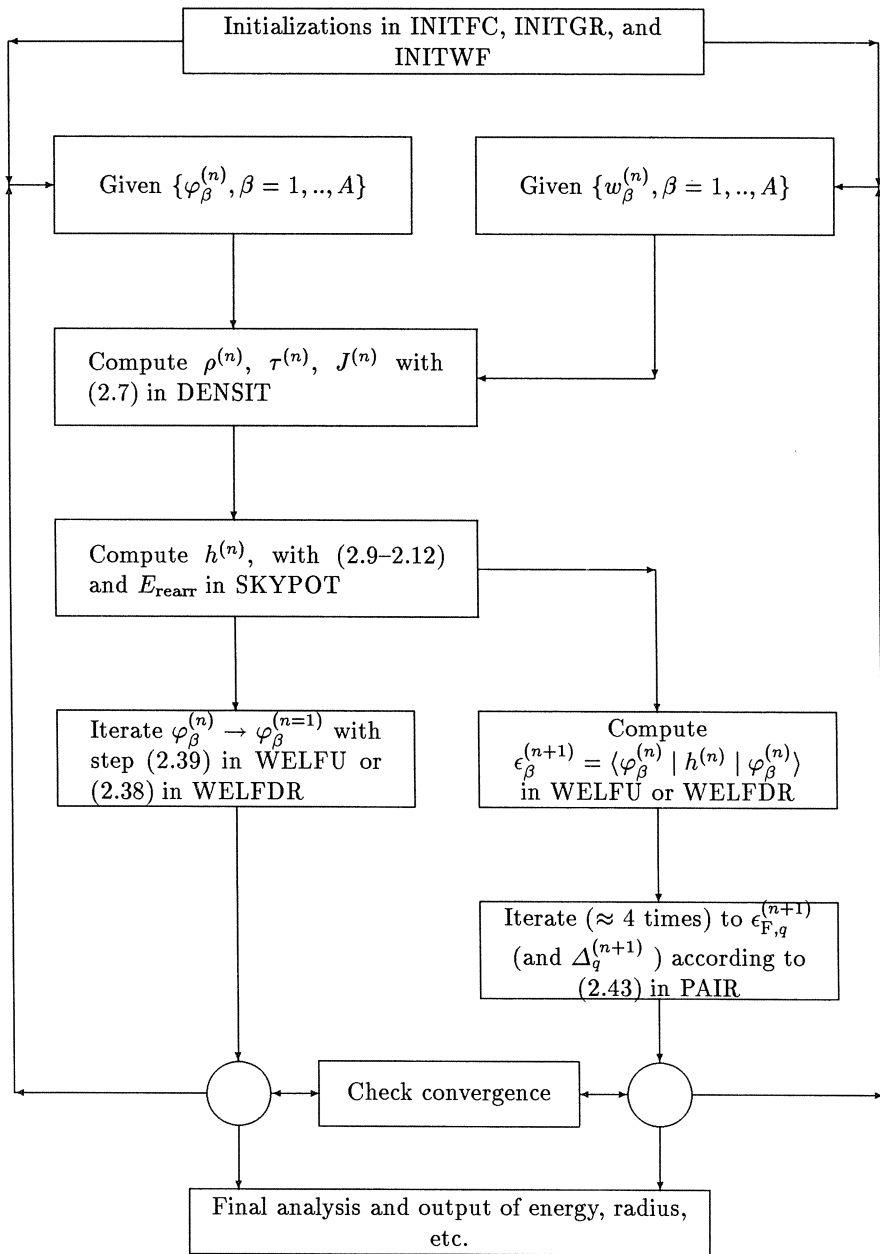
## 2.6 Instructions and Suggestions

At this point, papers usually have conclusions. For a presentation of a code, the appropriate end is an invitation to use the code and to experiment with it. Therefore, in this final section we give a short overview of the code structure and of the necessary input.

### 2.6.1 Structure of the Code

The main program, **HAFOMN**, serves only as a short master routine that sets default values for some of the parameters, contains all **READ** statements, and then quickly calls the principal subroutine **HAFOSU** which performs all other tasks, such as initialization, iteration, final analysis, and printing. This strategy to delegate everything except reading to a “master subroutine” was chosen to make the whole Hartree-Fock package easily callable from other routines. We used it, for example, extensively as the core of a least-squares fitting routine. All transfer to and from **HAFOSU** is done via common blocks. For detailed information on the I/O see the comments at the beginning of **HAFOSU**. A full explanation of each variable in each of the **COMMON** blocks is given at the top of the master routine **HAFOMN**.

The coupled Hartree-Fock and pairing equations are solved by an interleaved simultaneous iteration, preceded by an initialization phase and followed by an analysis (and output) phase. The structure of the code within **HAFOSU** is shown diagrammatically in Fig. 2.1. The decision to iterate a few pairing steps per wave-function step seems to introduce some asymmetry. The code under most circumstances also works with one pairing step per iteration. But the pairing step is orders of magnitude faster than the wave-function step. Thus, it is not worthwhile to be sparing with pairing steps at the (even small) risk of slowing down the whole iteration.



**Fig. 2.1.** Schematic flow chart of the code.

### 2.6.2 Examples for Input Data

In Table 2.3, we show a typical set of input lines to be placed into the file FOR005. Although all input variables are explained in the code, here we briefly discuss the input lines. We give input for three successive calculations with varying options and varying forces. The first set uses the force “Fit  $Z_\sigma$ ”, the second set “Skyrme M\*”, and the third set the old fashioned “Skyrme 3”; for definitions, references, and a variety of other parametrizations see Ref. [2.3]. In the first set, we use default input parameters wherever possible by entering no value at those places; the double magic nucleus  $^{208}\text{Pb}$  is calculated. In the second set, all parameters are given explicitly but still with their default values;  $^{124}\text{Sn}$  is computed. It has a magic proton number but needs pairing for the neutrons. Note that the low ITPR will produce error messages from the pairing routine, PAIR, during the first few iterations. One should not be worried by this because pairing converges well in the later iterations. In the third set, we have varied some options. Note that the pairing forces PAIRFP and PAIRFN are set to 0, which switches off the pairing iteration and calls for further input, namely the lines which explicitly specify the occupation weights  $w_\beta$ . The nucleus  $^{17}\text{O}$  is computed here in the filling approximation. All three sets start with IWUNIT = 0, which will produce output in file FOR006. In the following, we briefly explain each input variable:

**IWUNIT** Determines the output unit ( 0  $\rightarrow$  file FOR006, 1  $\rightarrow$  terminal ) or terminates the program ( -1 ).

**NGRID** =  $N_r$  = number of radial grid points. If NGRID = 0 is entered, the program automatically computes a reasonable grid size according to the number of nucleons (assuming normal nuclear sizes). *Default* = 0.

**RSTEP** =  $\Delta_r$  = grid spacing in units of fm. *Default* = 0.3, which suffices for 0.1% precision in energy and radius.

**IWR** = print level. Possible values are -2, -1, 0, 1. The higher the value the more detailed the printed output. *Default* = -1.

**IFPLOT** = switch for output on a plot file FOR011 (*Default* = 0):

- 0  $\rightarrow$  no plot file is written,
- 1  $\rightarrow$  a short plot file is written up to the single-particle energies,
- 2  $\rightarrow$  the densities are given in addition,
- 3  $\rightarrow$  the charge form factor is also given.

**DABR** = relative error in the radius which must be achieved before the iteration terminates. *Default* = 0.0001.

**DABE** = relative error in the energy which must be achieved before the iteration terminates. *Default* = 0.0001.

**ITMAX** = maximum number of iterations. *Default* = 80

**EPSPR** = precision of the pairing iteration.

**ITPR** = number of pairing steps per wave-function step. *Default* = 4.

**Table 2.3.** Example of input lines in file FOR005. The code reads only the numbers on the left half of the input lines. The right half is added as a comment to simplify changes in the input data.

0,	IWUNIT 'FIT Z-sigma '
''	NGRID,RSTEP
''	IWR,IFPLOT
''''	ITMAX,DABR,DABE,ITPR,EPSPR
'''	XODAMP,EODAMP,ADDNEW
1,1,1,0,0,	IF:EX,TM,ZPE,DFMS,RHOC
-1983.76,362.252,-104.27,11861.4,123.69,	TO,T1,T2,T3,T4
1.1717,0.,0.,1.7620,	X0,X1,X2,X3
0.25,1.0,,	POWER,ACOUL,PAIRFP,PAIRFN
82,126,,	NPROT,NNEUT,NMAXP,NMAXN
0,	IWUNIT 'Skyrme M*'
00,0.3,	NGRID,RSTEP
-1,0,	IWR,IFPLOT
80,0.0001,0.0001,4,0.0001,	ITMAX,DABR,DABE,ITPR,EPSPR
0.0,40.0,0.5,	XODAMP,EODAMP,ADDNEW
1,1,1,0,0,,	IF:EX,TM,ZPE,DFMS,RHOC
-2645.0,410.0,-135.0,15595.0,130.0,	TO,T1,T2,T3,T4
0.09,0.,0.,0.0,	X0,X1,X2,X3
0.16666667,1.0,-1.1E11,-1.1E11,	POWER,ACOUL,GAPP,GAPN
50,74,00,00,	NPROT,NNEUT,NMAXP,NMAXN
0,	IWUNIT 'Skyrme 3 '
00,0.3,	NGRID,RSTEP
-1,1,	IWR,IFPLOT
80,0.00001,0.00001,4,,	ITMAX,DABR,DABE,ITPR,EPSPR
1.00,40.0,0.5,	XODAMP,EODAMP,ADDNEW
1,0,0,0,0,	IF:EX,TM,ZPE,DFMS,RHOC
-1128.75,395.0,-95.0,14000.0,120.0,	TO,T1,T2,T3,T4
0.45,0.,0.,1.0,	X0,X1,X2,X3
1.0,1.0,0.0,0.0,	POWER,ACOUL,GAPP,GAPN
08,09,03,04,	NPROT,NNEUT,NMAXP,NMAXN
1,1,1,	WEIGHT(PROTONS)
1,1,1,.16666667,	WEIGHT(NEUTRONS)
-1,	TO STOP

- X0DAMP** =  $\delta$  = stepsize of damped gradient step (2.38). For *X0DAMP* = 0 the inverse gradient step (2.39) is run. *Default* = 0.
- E0DAMP** =  $E_0$  = energy parameter in kinetic-energy damping (2.38). *Default* = 50. Hint: try to explore optimum combinations of *X0DAMP* and *E0DAMP*; be aware that these may depend a bit on the force used.
- ADDNEW** = weight for admixing the new potentials  $U^{(n)}$  to the previous potentials  $U^{(n-1)}$  to stabilize the iteration.
- IFEX** = switch for Coulomb exchange according to (2.20) and (2.21):  
 1  $\rightarrow$  exchange is taken into account,  
 0  $\rightarrow$  exchange is omitted.
- IFTM** = switch for  $t_1$  and  $t_2$  contribution in the spin-orbit potential  $U_{ls,q}$ :  
 0  $\rightarrow$  the term is dropped,  
 1  $\rightarrow$  the term is taken into account.
- IFZPE** = switch for the handling of the center-of-mass correction:  
 1  $\rightarrow$  full correction according to (2.27),  
 0  $\rightarrow$  approximate correction according to (2.28).
- IFDFMS** = switch for equal or different nucleon masses:  
 0  $\rightarrow \frac{\hbar^2}{2m_{pr}} = \frac{\hbar^2}{2m_{ne}} = 20.7525 \text{ MeV fm}^2$ ,  
 1  $\rightarrow \frac{\hbar^2}{2m_{pr}} = 20.735 \text{ MeV fm}^2$  and  $\frac{\hbar^2}{2m_{ne}} = 20.721 \text{ MeV fm}^2$ .
- IFRHOC** = switch for proton folding in Coulomb potential:  
 1  $\rightarrow$  charge density is used in Coulomb potential (2.18),  
 0  $\rightarrow$  proton density is used in Coulomb potential.
- T0...X3** = parameters of the Skyrme force,  $t_0 \dots x_3$ .
- POWER** =  $\alpha$  = parameter of the Skyrme force.
- ACOL** = fractional charge of the protons. *Default* = 1.
- PAIRFP** = Pairing force for protons:  
 > 0  $\rightarrow$  constant-force approach with  $G_{pr} = \text{PAIRFP}$ ,  
 >  $10^{10}$   $\rightarrow$  constant-force approach with  $G_{pr} = 22 \text{ MeV/A}$ ,  
 < 0  $\rightarrow$  constant-gap approach with  $\Delta_{pr} = \text{ABS}(\text{PAIRFP})$ ,  
 <  $-10^{10}$   $\rightarrow$  constant-gap approach with  $\Delta_{pr} = 11.2 \text{ MeV}/\sqrt{A}$ ,  
 = 0  $\rightarrow$  override pairing and use **WEIGHT** from input line.
- PAIRFN** = pairing force for neutrons. Switches as for **PAIRFP**, but uses  $G_{ne} = 29 \text{ MeV/A}$  in case of **PAIRFN** >  $10^{10}$ .
- NPROT** = number of protons.
- NNEUT** = number of neutrons.
- NMAXP** =  $M_{pr}$  = number of proton shells. The code automatically computes the appropriate **NMAXP** for given **NPROT** if **NMAXP** = 0 is entered. *Default* = 0
- NMAXN** =  $M_{ne}$  = number of neutron shells. The code automatically computes the appropriate **NMAXN** for given **NNEUT** if **NMAXN** = 0 is entered. *Default* = 0

### 2.6.3 Technical Notes

The program has been tested on a PC with Prospero FORTRAN and RM-FORTRAN, on a CDC under NOS with the FTN5 compiler, and on an IBM 3090 with the VFORT compiler. One can just compile it and run. The program should also run immediately on a VAX. Slight differences between the machines occur only for the I/O. The assignment to the unit “CON” for direct output on the terminal (see the `OPEN` statement for `UNIT=6` in `HAFOMN`) is specific for the PC and needs to be changed for each machine.

Each routine contains the same `PARAMETER` statement in the header. All `PARAMETERs` in the `PARAMETER` statement are explained in the header of the main program `HAFOMN`. All critical dimensioning is done via parameters. Thus the program can be easily readjusted for different circumstances by a global change of the appropriate parameter. We recommend collecting the `PARAMETER` statement and each `COMMON` block separately into an `INCLUDE` file and inserting the corresponding `INCLUDE` statement instead of the `PARAMETER` or `COMMON` sequence.

The code only uses generic names in function calls and never uses real constants in a calling list or an `IF` statement. Thus one can easily switch to double precision (although we rarely find it necessary). To that end each routine is headed by an `IMPLICIT DOUBLE PRECISION` which is switched off by a `CDOUB` in the first columns. One merely has to change all `CDOUB` into five blanks to convert the code to double precision.

## References

- [2.1] T.H.R. Skyrme, Nucl. Phys. **9** (1959) 615  
D. Vautherin and D. M. Brink, Phys. Rev. **C5** (1972) 626
- [2.2] P. Quentin and H. Flocard, Ann. Rev. Nucl. Part. Sci. **28** (1978) 523  
K. Goeke and P.-G. Reinhard, Lecture Notes in Physics, vol **171** (1982)
- [2.3] J. Friedrich and P.-G. Reinhard, Phys. Rev. **C33** (1986) 335
- [2.4] H. Pfeiffer, P.-G. Reinhard, and D. Drechsel, Z. Phys. **A292** (1979) 375
- [2.5] P.-G. Reinhard and R. Y. Cusson, Nucl. Phys. **A378** (1982) 418
- [2.6] M. R. Strayer, R. Y. Cusson, A. S. Umar, P.-G. Reinhard, D. A. Bromley, and W. Greiner, Phys. Lett. **B135** (1984) 261  
R. Y. Cusson, P.-G. Reinhard, M. R. Strayer, J. A. Maruhn, and W. Greiner, Z. Phys. **A320** (1985) 475
- [2.7] P.-G. Reinhard, F. Grümmer, and K. Goeke, Z. Phys. **A317** (1984) 339
- [2.8] P.-G. Reinhard and J. Friedrich, Z. Phys. **321** (1985) 619
- [2.9] A. R. Edmonds, *Angular Momentum in Quantum Mechanics* (Princeton University Press, Princeton 1957)
- [2.10] J. Blocki and M. Flocard, Nucl. Phys. **A273** (1976) 45
- [2.11] P. Ring and P. Schuck, *The Nuclear Many Body Problem* (Springer, Berlin, Heidelberg 1980)

- [2.12] J. M. Eisenberg and W. Greiner, *Nuclear Models* (North-Holland, Amsterdam 1971)
- [2.13] S.-J. Lee et al., Phys. Rev. Lett. **57** (1986) 2916
- [2.14] R. Zurmühl, *Praktische Mathematik* (Springer, Berlin, Heidelberg 1963) Chap. VII, Sect. 3
- [2.15] B. Dreher, J. Friedrich, K. Merle, G. Lühns and H. Rothaas, Nucl. Phys. **A235** (1974) 219
- [2.16] J. L. Friar, J. W. Negele, Adv. Nucl. Phys. **8** (1975) 219
- [2.17] V. H. Walther, private communication; G. G. Simon, C. Schmitt, F. Borkowski, and V. H. Walther, Nucl. Phys. **A333** (1980) 318
- [2.18] J. Friedrich and N. Voegeler, Nucl. Phys. **A373** (1982) 191