POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Title

Tesi di Laurea Magistrale in
Xxxxxxx Engineering - Ingegneria Xxxxxxx

Author: **Name Surname**

Student ID: 000000
Advisor: Prof. Name Surname
Co-advisors: Name Surname, Name Surname
Academic Year: 20XX-XX

Abstract

Here goes the Abstract in English of your thesis followed by a list of keywords. The Abstract is a concise summary of the content of the thesis (single page of text) and a guide to the most important contributions included in your thesis. The Abstract is the very last thing you write. It should be a self-contained text and should be clear to someone who hasn't (yet) read the whole manuscript. The Abstract should contain the answers to the main scientific questions that have been addressed in your thesis. It needs to summarize the adopted motivations and the adopted methodological approach as well as the findings of your work and their relevance and impact. The Abstract is the part appearing in the record of your thesis inside POLITesi, the Digital Archive of PhD and Master Theses (Laurea Magistrale) of Politecnico di Milano. The Abstract will be followed by a list of four to six keywords. Keywords are a tool to help indexers and search engines to find relevant documents. To be relevant and effective, keywords must be chosen carefully. They should represent the content of your work and be specific to your field or sub-field. Keywords may be a single word or two to four words.

Keywords: here, the keywords, of your thesis

Abstract in lingua italiana

Qui va l'Abstract in lingua italiana della tesi seguito dalla lista di parole chiave.

Parole chiave: qui, vanno, le parole chiave, della tesi

Contents

1 | Energy functional

1.1. Hartree-Fock theory

An empirical description of nuclear structure can be carried out using phenomenological models, as reported in section (REF).
A more rigorous approach needs to take into account the fact that the mean field which the nucleons interact with, is generated by the nucleons themselves, due to some microscopic interaction. Starting from the many-body Hamiltonian of the system, we will be able to extract a single particle hamiltonian, where the nucleon is subject to a mean field potential generated by the effective microscopic force.
The many-body hamiltonian of the system, made of A fermions, given by

$$\hat{H} = \hat{T} + \hat{V} = \sum_i -\frac{\hbar^2}{2m}\nabla_i^2 + \sum_{i<j} v_{ij}^{(2)} + \sum_{i<j<k} v_{ijk}^{(3)} \tag{1.1}$$

which acts on the slater determinant given by

$$\Psi = \frac{1}{\sqrt{A!}} \sum_{\{p\}} (-1)^p \varphi_{p(1)}(\boldsymbol{r}_1) \dots \varphi_{p(A)}(\boldsymbol{r}_A) \tag{1.2}$$

where φ_i are single-particle states. The slater determinant sums over all possible permutations of the A fermions on the single particle states, with a $-$ sign according to the parity of the permutation.

1.1.1. Hartree-Fock equations

It is possible to show [20] that the ground state of the many-body system, found by minimizing the functional

$$E[\Psi] = \frac{\langle \Psi | \hat{H} | \Psi \rangle}{\langle \Psi | \Psi \rangle} \tag{1.3}$$

Is found in the basis of eigenstates of the single-particle hamiltonian, found by setting to 0 the functional variation of $E[\Psi]$ in φ_i^* with the orthonormality constraint.

$$\frac{\delta}{\delta\varphi_i^*}\left(E[\Psi] - \lambda \int \varphi_j^*\varphi_i d\boldsymbol{r}\right) = 0 \tag{1.4}$$

Doing this yields the *Hartree-Fock equations*

$$-\frac{\hbar^2}{2m}\nabla^2\varphi_i + \sum_j^A \int \varphi_j^*(\boldsymbol{r}')v_{ij}(\boldsymbol{r},\boldsymbol{r}')\varphi_j(\boldsymbol{r}')\varphi_i(\boldsymbol{r})d\boldsymbol{r}' - \sum_j^A \int \varphi_j^*(\boldsymbol{r}')v_{ij}(\boldsymbol{r},\boldsymbol{r}')\varphi_j(\boldsymbol{r})\varphi_i(\boldsymbol{r}')d\boldsymbol{r}' = \varepsilon_i\varphi_i \tag{1.5}$$

Here, a couple of observations are in order.
The first interaction term, called Hartree term, arises from considering independent particles, and is also routinely found in classical physics. The second one, called Fock term, or exchange term is non-local and is given by the quantum mechanical nature of the system. From the standpoint of the solution of the eigenvalue problem, the Fock term is very problematic, and is usually avoided using finite-range interaction like the Gogny force [21], or contact forces like the Skyrme one [25], used in the present work. They render the exchange term as a local one.
Even if the interaction terms are local, the equation is still highly non-linear, since the mean field potential will be a function of the various eigenstates. The consequece is that the equation will be solved *self-consistently*, that is, by solving for the set of orbitals $\{\varphi_i\}$, using them to build the new mean field, and solving again, repeating the process until convergence.

1.1.2. Symmetries

Since the objective of this work is to solve the Hartree-Fock equations without spatial symmetry assumptions, it is useful to first understand how symmetries propagate along the self-consistent calculation.
We start by defining the creation and annihilation operators of the single particle hamiltonian eigenstates, $a_i^\dagger, a_i$, which abide the usual anticommutation relations of fermions

$$\{a_i, a_j^\dagger\} = \delta_{ij} \tag{1.6}$$

If we expand on a different, orthonormal complete basis $\{\chi_l\}$, we can write the corresponding creation and annihilation operators $c_l^\dagger, c_l$ as

$$\varphi_k = \sum_l D_{lk}\chi_l \tag{1.7}$$

$$a_k^\dagger = \sum_l D_{lk}c_l^\dagger \tag{1.8}$$

$$a_k = \sum_l D_{lk}^\dagger c_l \tag{1.9}$$

Since orthonormality is guaranteed for both sets, taking

$$\delta_{jk} = \langle\varphi_j|\varphi_k\rangle = \sum_{ll'} D_{l'j}^\dagger D_{lk} \langle\chi_l|\chi_l'\rangle \implies DD^\dagger = 1 \tag{1.10}$$

Symmetry propagation

COMPLETARE

1.1.3. Density Functional Theory

It shall be evident shortly, in section 1.2, that a more general approach to microscopic models has to be taken, in order to give a satisfactory description of the nuclear world. The framework that we'll briefly outline here is called Density Functional Theory (DFT). DFT was introduced by P. Hohenberg and W. Kohn in 1964 [10], by proving two theorems. The **first** HK theorem states that the energy of a fermion system, subject to an external potential V_{ext} can be expressed solely as a functional of the particle density ρ of the system.

$$E[\rho] = F[\rho] + \int V_{\text{ext}}\rho(\boldsymbol{r})d\mathbf{r} \tag{1.11}$$

While the **second** one states that the ground state of the system is found by minimizing its variation with respect to ρ.
HK theorems are fundamental but not constructive [4], since they do not provide a form for the functional F, which is intrinsic to the physics of the fermions at hand. COMPLETARE

1.2. Skyrme

Now that the theoretical framework is clear, we can investigate a plausible nucleonic interaction, which in the present work, takes the form of the Skyrme force.

1.2.1. Skyrme force

It was first proposed by Tony Skyrme in 1958 [25] as a zero range force between nucleons, and has been used successfully as the building block of theoretical nuclear structure. It comprises a two body attractive term that reads

$$\begin{aligned} v^{(2)}(\mathbf{r}_1,\mathbf{r}_2) = {} & t_0\left(1+x_0P_\sigma\right)\delta(\mathbf{r}) \\ & + \frac{1}{2}t_1\left(1+x_1P_\sigma\right)\left[\mathbf{P}'^2\delta(\mathbf{r})+\delta(\mathbf{r})\mathbf{P}^2\right] \\ & + t_2P_\sigma\mathbf{P}'\cdot\delta(\mathbf{r})\mathbf{P} \\ & + iW_0\boldsymbol{\sigma}\cdot[\mathbf{P}'\times\delta(\mathbf{r})\mathbf{P}] \end{aligned}$$

And a three body interaction, that is

$$v^{(3)}(\boldsymbol{r}_1,\boldsymbol{r}_2,\boldsymbol{r}_3) = t_3\delta(\boldsymbol{r}_1-\boldsymbol{r}_2)\delta(\boldsymbol{r}_2-\boldsymbol{r}_3)$$

which mimics the repulsive three-body force; without it, a collapse of the nuclear density would occur.
It's trivial to show that the three-body term is equivalent to a two-body, density-dependent interaction: [29]

$$v^{(3)}(\boldsymbol{r}_1,\boldsymbol{r}_2) = \frac{1}{6}t_3(1+P_\sigma)\delta(\boldsymbol{r})\rho(\boldsymbol{R}) \tag{1.12}$$

The different operators here are defined as

$$\mathbf{r}=\mathbf{r}_1-\mathbf{r}_2 \quad \mathbf{R}=\frac{\mathbf{r}_1+\mathbf{r}_2}{2} \tag{1.13}$$

which are respectively the relative position of two particles and their center of mass coordinate, assuming equal masses.

$$\mathbf{P}=\frac{-i(\nabla_1-\nabla_2)}{2} \tag{1.14}$$

which is the so called relative wave-number operator.

$$\boldsymbol{\sigma}=\boldsymbol{\sigma}_1+\boldsymbol{\sigma}_2 \tag{1.15}$$

being the total spin of the two interacting particles, and lastly

$$\mathbf{P}_\sigma=\frac{(1+\boldsymbol{\sigma}_1\cdot\boldsymbol{\sigma}_2)}{2} \tag{1.16}$$

which represents the spin-exchange operator. Primed operators refer to the adjoint acting on the left.
The zero-range characteristic takes the form of a Dirac delta $\delta(\mathbf{r})$, which allows the writing of the Fock term detailed in (REF) as a purely local one.
Taking the expectation value of the many body hamiltonian, in the Hilbert space of Slater determinants, yields an energy density which can be expressed as a function of $\rho_q, \tau_q, \boldsymbol{J}_q$ [29].

$$\langle H \rangle = \langle \Psi | H | \Psi \rangle = \int \mathcal{H}(\boldsymbol{r}) d\mathbf{r} \tag{1.17}$$

Modern parametrization

The Skyrme force has evolved from the original one to accomodate new nuclei, done through the addition of a few parameters, yielding the following form of the interaction [2]

$$\begin{aligned} v^{(2)}(\mathbf{r}_1, \mathbf{r}_2) = & \, t_0 \left(1 + x_0 P_\sigma\right) \delta(\mathbf{r}) \\ & + \frac{1}{2} t_1 \left(1 + x_1 P_\sigma\right) \left[\mathbf{P}'^2 \delta(\mathbf{r}) + \delta(\mathbf{r}) \mathbf{P}^2\right] \\ & + t_2 \left(1 + x_2 P_\sigma\right) \mathbf{P}' \cdot \delta(\mathbf{r}) \mathbf{P} \\ & + \frac{1}{6} t_3 \left(1 + x_3 P_\sigma\right) \left[\rho(\mathbf{R})\right]^\sigma \delta(\mathbf{r}) \\ & + i W_0 \boldsymbol{\sigma} \cdot \left[\mathbf{P}' \times \delta(\mathbf{r}) \mathbf{P}\right] \\ & + \frac{1}{6} t_3 \left(1 + x_3 P_\sigma\right) \left[\rho(\mathbf{R})\right]^\sigma \delta(\mathbf{r}) \end{aligned}$$

Here, the boundary between Hartree-Fock and DFT starts to thin out, as the exponent σ of the density makes that piece of the force a true three-body interaction only for the value $\sigma = 1$ [6].
On top of that, additional, empirical tweaking of the resulting energy density needed to reach satisfactory physical accuracy, such as the case for the spin-orbit couplings [19], prompts for the following, well established proceeding: use the Skyrme interaction as a starting guide for building the energy functional to employ Density Functional Theory.

1.2.2. Skyrme functional

The energy functional we want to minimize is rather complex, due to the rich phenomenology of nuclear interactions and numerical nuances.
The complete energy functional is

$$E_{\mathrm{HF}} = E_{\mathrm{Skyrme}} + E_{\mathrm{Coul}} + E_{\mathrm{Kin}} \tag{1.18}$$

We'll begin by looking at the Skyrme and kinetic parts, while later on give a treatment for the Coulomb one.

$$\langle H \rangle = \langle \Psi | \, H \, | \Psi \rangle = \int (\mathcal{E}_{\text{Skyrme}} + \mathcal{E}_{\text{Kin}}) d\mathbf{r} = \int \mathcal{E} d\mathbf{r} \tag{1.19}$$

Since the Skyrme interaction is time-reversal invariant (as well as the kinetic operator), this means that the total Hamiltonian must be time-even. Different additive contributions must be time-even as well, even if the densities from which they are calculated are not. Unless we reduce to the even-even nucleus case, where total angular momentum is defined at $J = 0$, giving vanishing time-odd densities. This allows us to write the functional in a simpler form as [28]

$$\mathcal{E}_{\text{Kin}} = \frac{\hbar^2}{2m}\tau \tag{1.20}$$

$$\mathcal{E}_{\text{Skyrme}} = \sum_{t=0,1} \left\{ C_t^{\rho}[\rho_0]\rho_t^2 + C_t^{\Delta\rho}\rho_t \nabla^2 \rho_t + C_t^{\nabla \cdot J}\rho_t \nabla \cdot \mathbf{J}_t + C_t^{\tau}\rho_t\tau_t \right\} \tag{1.21}$$

Here, $t = 0, 1$ refers to the isoscalar and isovector components of the densities, e.g.

$$\rho_0 = \rho_p + \rho_n$$
$$\rho_1 = \rho_p - \rho_n$$

Where

$$C_0^{\rho} = +\frac{3}{8}t_0 + \frac{3}{48}t_3\rho_0^{\sigma} \tag{1.22}$$

$$C_1^{\rho} = -\frac{1}{8}t_0(1 + 2x_0) - \frac{1}{48}t_3(1 + x_3)\rho_0^{\sigma} \tag{1.23}$$

$$C_0^{\tau} = +\frac{3}{16}t_1 + \frac{1}{16}t_2(5 + 4x_2) \tag{1.24}$$

$$C_1^{\tau} = -\frac{1}{16}t_1(1 + 2x_1) + \frac{1}{16}t_2(1 + 2x_2) \tag{1.25}$$

$$C_0^{\Delta\rho} = -\frac{9}{64}t_1 + \frac{1}{64}t_2(5 + 4x_2) \tag{1.26}$$

$$C_1^{\Delta\rho} = +\frac{3}{64}t_1(1 + 2x_1) + \frac{1}{64}t_2(1 + 2x_2) \tag{1.27}$$

$$C_0^{\nabla \cdot J} = -\frac{3}{4}W_0 \tag{1.28}$$

$$C_1^{\nabla \cdot J} = -\frac{1}{4}W_0 \tag{1.29}$$

As outlined in previous chapters (REF), we can now derive the Kohn-Sham equations, by constraining orthonormality and enforcing the variation of the functional to be zero.

What we end up with is

$$\left[-\nabla\left(\frac{\hbar^2}{2m_q^*(\mathbf{r})}\nabla\right) + U_q(\mathbf{r}) + \delta_{\mathrm{q,proton}}U_C(\mathbf{r}) - i\mathbf{B}_q(\mathbf{r})\cdot(\nabla\times\boldsymbol{\sigma})\right]\varphi_\alpha = \varepsilon_\alpha\varphi_\alpha \tag{1.30}$$

The index $q = n, p$ refers respectively to the neutron and proton quantities.
Where the different terms are given by

$$\frac{\hbar^2}{2m_q^*(\mathbf{r})} = \frac{\delta\mathcal{E}}{\delta\tau_q} \tag{1.31}$$

$$U_q(\mathbf{r}) = \frac{\delta\mathcal{E}}{\delta\rho_q} \tag{1.32}$$

$$\mathbf{B}_q(\mathbf{r}) = \frac{\delta\mathcal{E}}{\delta\mathbf{J_q}} \tag{1.33}$$

The coulomb field U_C, which is present only in the single particle equation for protons, doesn't come from the skyrme interaction, rather from the Coulomb part of the whole functional. It will be properly derived in section 1.3.
Following the rules for functional derivatives, outlined in the appendix (REF) we get

$$\frac{\hbar^2}{2m_q^*(\mathbf{r})} = + \frac{\hbar^2}{2m} \tag{1.34}$$

$$+ \frac{1}{8}[t_1(2+x_1) + t_2(2+x_2)]\rho(\mathbf{r}) \tag{1.35}$$

$$- \frac{1}{8}[t_1(1+2x_1) + t_2(1+2x_2)]\rho_q(\mathbf{r}) \tag{1.36}$$

$$\tag{1.37}$$

$$U_q(\mathbf{r}) = + \frac{1}{8}[t_1(2+x_1) + t_2(2+x_2)]\rho \tag{1.38}$$

$$+ \frac{1}{8}[t_2(1+2x_2) - t_1(1+2x_1)]\rho_q \tag{1.39}$$

$$+ \frac{1}{8}[t_1(2+x_1) + t_2(2+x_2)]\tau \tag{1.40}$$

$$+ \frac{1}{8}[t_2(1+2x_2) - t_1(1+2x_1)]\tau_q \tag{1.41}$$

$$+ \frac{1}{16}[t_2(2+x_2) - 3t_1(2+x_1)]\nabla^2\rho \tag{1.42}$$

$$+ \frac{1}{16}[3t_1(2x_1+1) + t_2(2x_2+1)]\nabla^2\rho_q \tag{1.43}$$

$$\tag{1.44}$$

$$\mathbf{B}_q(\mathbf{r}) = + \frac{1}{2}W_0[\nabla\rho + \nabla\rho_q] \tag{1.45}$$

$$- \frac{1}{8}(t_1x_1 + t_2x_2)\mathbf{J} + \frac{1}{8}(t_1 - t_2)\mathbf{J}_q \tag{1.46}$$

Unless otherwise specified, unindexed densities denote isoscalar quantities (sum of neutron's and proton's).

1.3. Coulomb interaction

Unlike the Skyrme interaction, the Coulomb force is not local, giving rise to an unwanted integral operator in the Hamiltonian. A well known and widely used device is the Slater approximation [9], which gives a local exchange interaction.
In this approximation, the Coulomb energy reads

$$E_{\text{Coul}} = \int \mathcal{E}_{\text{Coul}}(\mathbf{r}) d\boldsymbol{r}$$

$$\mathcal{E}_{\text{Coul}}(\boldsymbol{r}) = \frac{e^2}{2}\left[\int \frac{\rho_p(\mathbf{r})\rho_p(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}' - \frac{3}{2}\left(\frac{3}{\pi}\right)^{\frac{1}{3}} \rho_p^{4/3}(\mathbf{r})\right]$$

Which gives

$$U_C(\mathbf{r}) = \frac{\delta \mathcal{E}_{\text{Coul}}}{\delta \rho_p} = \frac{e^2}{2}\left[\int \frac{\rho_p(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d^3\mathbf{r}' - 2\left(\frac{3}{\pi}\right)^{\frac{1}{3}} \rho_p^{1/3}(\mathbf{r})\right] \tag{1.47}$$

From a computational standpoint, the exchange part is trivial, while the direct one is more involved. One could compute the integral, but the complexity on a 3D mesh is $\mathcal{O}(N^6)$, rendering it unusable for fine meshes.
An alternative approach is to solve the poisson equation (from now on, V_c refers to the direct part only)

$$\nabla^2 V_c = 4\pi e^2 \rho_p \tag{1.48}$$

Given the proton density, we can impose Dirichlet boundary conditions, which can be extracted from a quadrupole expansion of the charge density [11]

$$V_c(\mathbf{r}) = 4\pi e^2 \sum_{\lambda=0}^{2} \sum_{\mu=-\lambda}^{\lambda} \frac{\langle Q_{\lambda\mu}\rangle\, Y_{\lambda\mu}}{r^{1+\lambda}} \text{ on } \partial\Omega \tag{1.49}$$

Where $\langle Q_{\lambda\mu}\rangle$ is defined as

$$\langle Q_{\lambda\mu}\rangle = \int r^\lambda Y^*_{\lambda\mu}(\mathbf{r}) \rho_p(\mathbf{r}) d^3\mathbf{r} \tag{1.50}$$

Since we expect a charge density confined to the nuclear shape, higher order terms in the expansion can be neglected, provided that the box is sufficiently large.
In a reference frame where the nucleus center of mass is at the origin, the expansion

reduces to

$$V_c(\mathbf{r}) = \frac{Ze^2}{r} + e^2 \sum_{\mu=-2}^{2} \frac{\langle Q_{2\mu} \rangle Y_{2\mu}}{r^3} \text{ on } \partial\Omega \tag{1.51}$$

Refer to appendix A.1 for the definition and numerical evaluation of the spherical harmonics $Y_{\lambda\mu}$.

1.4. Energy calculation

One, if not the most important physical quantity we want to compute is the total energy of the system.

Integrated energy

The obvious way would be to evaluate the functional for a given density. We will call this *integrated energy*.

$$E_{\text{int}} = E[\rho, \tau, J_{\mu\nu}] = \int \mathcal{E} d\mathbf{r}$$

Hartree-Fock energy

An alternative approach can be used, as in a stationary point $\delta E = 0$, the single particle eigenvalue equation 1.30 stands true, summarized as

$$(\hat{t} + U)\varphi_k = \varepsilon_k \varphi_k \tag{1.52}$$

We can multiply 1.52 on the left by φ_k^* and integrate to get

$$\int -\varphi_k^* \frac{\hbar^2}{2m} \nabla^2 \varphi_k d\boldsymbol{r} + \int \varphi_k^* U \varphi_k d\boldsymbol{r} = \int \varphi_l^* \varepsilon_k \varphi_k d\boldsymbol{r} \tag{1.53}$$

The integral on the right hand side of 1.53 evaluates to ε_k due to the orthonormality constraint. If we sum over all states k we get

$$\sum_k \left\{ \int -\varphi_k^* \frac{\hbar^2}{2m} \nabla^2 \varphi_k d\boldsymbol{r} + \int \varphi_k^* U \varphi_k d\boldsymbol{r} \right\} = \sum_k \varepsilon_k \tag{1.54}$$

$$\sum_k t_k + \int \rho U = \sum_k \varepsilon_k \tag{1.55}$$

Since U is calculated as 1.32, assuming that the functional has a power dependence from ρ of the form $\mathcal{E}_{\text{Skyrme}} = A\rho^{\sigma+1}$ as in our case, we get the *rearrangement energy*

$$\rho U = \rho\frac{\delta\mathcal{E}_{\text{Skyrme}}}{\delta\rho} = \rho(\sigma+1)A\rho^{\sigma} = (\sigma+1)A\rho^{\sigma+1} = \mathcal{E}_{\text{Skyrme}} + \sigma\mathcal{E}_{\text{Skyrme}} = \mathcal{E}_{\text{Skyrme}} - \mathcal{E}_{\text{rea}} \tag{1.56}$$

If we explicit ρU in equation 1.55 using 1.56, we get to

$$\sum_k t_k + \int(\mathcal{E}_{\text{Skyrme}} - \mathcal{E}_{\text{rea}})d\boldsymbol{r} = \sum_k \varepsilon_k$$

Isolating the Skyrme energy density

$$\int \mathcal{E}_{\text{Skyrme}}d\boldsymbol{r} = \sum_k(\varepsilon_k - t_k) + \int \mathcal{E}_{\text{rea}}d\boldsymbol{r} \tag{1.57}$$

and given the total energy of the system from 1.19

$$E = \sum_k t_k + \frac{1}{2}\int \mathcal{E}_{\text{Skyrme}}d\boldsymbol{r} \tag{1.58}$$

substituting 1.57 in 1.58 yields

$$E_{\text{HF}} = \frac{1}{2}\sum_k(\varepsilon_k + t_k) + \int \mathcal{E}_{\text{rea}}d\boldsymbol{r} = \frac{1}{2}\left(T + \sum_k \varepsilon_k\right) + E_{\text{rea}} \tag{1.59}$$

which will be called *Hartree-Fock energy* throughout this text.

Sidenote: The actual functional has a plethora of ρ terms, which can be summarized as

$$\mathcal{E}_{\text{Skyrme}} = \sum_j A_j\rho^{\sigma_j+1} \implies E_{\text{rea}} = -\sum_j \sigma_j A_j\rho^{\sigma_j+1}$$

This means that only terms with a $\sigma_j \neq 0, -1$ actually contribute to the rearrangement energy.

Since equation 1.59 is valid only for $\delta E = 0$, it's useful to check its equivalence with the integrated energy at convergence, so one can be sure to actually be in a stationary point.

2 | Numerical methods

This chapter is devoted to tackling the practical implementation of the nuclear Hartree-Fock method, with a focus on the discretized evaluation of the relevant PDEs, the approximate solution of the eigenvalue problem, and the solution of the self-consistent problem.

2.1. Finite differences

The framework used to numerically solve the relevant PDEs of the problem, is the one of finite differences. The core idea is to discretize the domain on a 3D mesh, use Taylor expansions to approximate differential operators and then solve the resulting system of linear equations.

2.1.1. 3D mesh

The first step of the process is representing the different fields in a numerical, discretized fashon. Generally speaking, we deal at most with 2-rank tensors, which vary in space and spin.
Discretizing the 3D cartesian space with a 3-index mesh, choosing a box which size on x, y, z is respectively $[-a_x, a_x]$, $[-a_y, a_y]$, $[-a_z, a_z]$, and a number of points N_x, N_y, N_z, the resulting lattice will be given by

$$V = \{(-a_x + ih_x, -a_y + ih_y, -a_z + ih_z)\} = \{(x_i, y_j, z_k)\}$$

Where the indices and step sizes are

$$\begin{aligned} i = 0, \dots, N_x - 1 \quad h_x &= \frac{2a_x}{N_x - 1} \\ j = 0, \dots, N_y - 1 \quad h_y &= \frac{2a_y}{N_y - 1} \\ k = 0, \dots, N_z - 1 \quad h_z &= \frac{2a_z}{N_z - 1} \end{aligned}$$

For ease of notation, we will assume $a = a_x = a_y = a_z = a$ and $N = N_x = N_y = N_z = N$, without losing generality.
Including the spin degree of freedom, we can finally represent fields in a numerical way through

$$\psi(\boldsymbol{r}, \sigma) \mapsto \psi(x_i, y_j, z_k, s) = \psi_{ijks} \tag{2.1}$$

Differential operators discretization

By using Taylor series, it's possible to write approximations to derivatives [31], in any point of the lattice, of any (reasonable) order of accuracy, involving only near neighbouring points. In the present work, 5-points derivates are used, meaning Taylor expansions are written for $\psi(x \pm h)$ and $\psi(x \pm 2h)$ to compute the differential operators. Formulae for first and second derivates are given in appendix A.2.
From the theory background of chapter 1, we discern two main kinds of PDEs, the Schrödinger like KS equation, and the Poisson equation. Both are similar to a diffusion equation, but the former involves the spin degree of freedom, while the latter requires careful treatment of the boundary conditions.

2.1.2. Schrödinger equation

Starting from the Schrödinger equation 1.30, reported here for clarity

$$\left[-\nabla\left(\frac{\hbar^2}{2m_q^*(\mathbf{r})}\nabla\right) + U_q(\mathbf{r}) + \delta_{\mathrm{q,proton}} U_C(\mathbf{r}) - i\mathbf{B}_q(\mathbf{r}) \cdot (\nabla \times \boldsymbol{\sigma}) \right] \psi = \varepsilon\psi$$

it can be compactly written as

$$f(\nabla^2\psi, \nabla\psi, \psi, \boldsymbol{r}, s) = \varepsilon\psi. \tag{2.2}$$

If f is linear in ψ, it would be possible to rewrite it as a linear combination of ψ on the mesh, after which we can use linear algebra methods to solve the problem.

Linearity Breaking down each part of the equation, the kinetic term

$$\nabla\left(\frac{\hbar^2}{2m_q^*(\mathbf{r})}\nabla\right)\psi = \frac{\hbar^2}{2m_q^*(\mathbf{r})}\nabla^2\psi + \nabla\left(\frac{\hbar^2}{2m_q^*(\mathbf{r})}\right) \cdot \nabla\psi \tag{2.3}$$

Is evidently linear in ψ.
The spin-orbit coupling, which most generally reads

$$\begin{aligned}\hat{h}_{\mathrm{SO}} &= \boldsymbol{f}(\boldsymbol{r}) \cdot (\nabla \times \boldsymbol{\sigma}) \\ &= f_x(\boldsymbol{r})(\sigma_z \partial_y - \sigma_y \partial_z) + f_y(\boldsymbol{r})(\sigma_x \partial_z - \sigma_z \partial_x) + f_z(\boldsymbol{r})(\sigma_y \partial_x - \sigma_x \partial_y)\end{aligned}$$

where $\hat{h}_{\mathrm{SO}}$ acts on the spinor ψ_{ijk}. Since $\boldsymbol{\sigma}$ is the vector of Pauli matrices acting as linear operators on ψ_{ijk}, this portion of the equation is linear in ψ.
Finally, the mean field terms U_q, U_c

$$(U_q + \delta_{\mathrm{q,proton}} U_c)\psi$$

are just multiplicative, hence linear.
Given that the whole equation is linear in ψ, we can evaluate it on the chosen mesh, using finite differences to approximate the differential operators, yielding a linear eigenvalue problem of the form

$$\sum_{n}^{N_x \cdot N_y \cdot N_z \cdot 2} A_{mn}\psi_n = E\psi_m \tag{2.4}$$

Boundary conditions

We expect the nucleus to be a localized object, prompting null Dirichlet boundary conditions for the Schrödinger equation. Near the boundaries, the derivatives will involve points outside the box, setting these points to zero, is equivalent to solving

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \psi_{-2} \\ \psi_{-1} \\ \psi \\ \psi_N \\ \psi_{N+1} \end{bmatrix} = E \begin{bmatrix} \psi_{-2} \\ \psi_{-1} \\ \psi \\ \psi_N \\ \psi_{N+1} \end{bmatrix} \tag{2.5}$$

From this system of equations, we get for points outside the boundary:

$$\begin{cases} \psi_{-2} = 0 \\ \psi_{-1} = 0 \\ \dots \end{cases} \tag{2.6}$$

Meaning that ψ outside the box will automatically be set to zero if the A matrix is built assuming those points to be zero when computing its coefficients.

2.1.3. Poisson equation

The other fundamental PDE we need to solve is the Poisson equation encountered in section (REF). Dropping the c and p subscripts, it reads

$$\nabla^2 V = 4\pi e^2 \rho$$

It's simpler than the Schrödinger equation, as it only involves a laplacian and it's not an eigenvalue problem. The right is side is given, and the solution is found by inverting the coefficients' matrix.

Boundary conditions

Unlike the Schrödinger equation, we do not expect the solution to rapidly decay near the boundaries; as reported in section 1.3, we have fixed, non-null boundary conditions, which we have to properly impose on the system.
We can choose a direction, say x, and look at the discretized equation at the boundaries $x = \pm a$. Since the indeces j, k won't vary, we can omit them, and ignore the other derivates in the following equations.

$$\begin{aligned} \nabla^2 V &= \partial_{xx} V + \partial_{yy} V + \partial_{zz} V \\ &= \frac{-V_{i-2} + 16V_{i-1} - 30V_i + 16V_{i+1} - V_{i+2}}{12h^2} + \cdots = 4\pi e^2 \rho_i \end{aligned} \tag{2.7}$$

Near a boundary, say $i = 0$, the formula calls for points outside the box, known as *ghost points*. Since they are not part of the linear system, but they are known, we can bring them on the right side of equation 2.7.

$$\frac{-30V_0 + 16V_1 - V_2}{12h^2} = 4\pi e^2 \rho_0 + \frac{V_{-2} - 16V_{-1}}{12h^2} = \tilde{\rho}_0 \tag{2.8}$$

The same procedure must be applied to all equations involving ghost points, e.g. for $i = 1$

$$\frac{+16V_0 - 30V_1 + 16V_2 - V_3}{12h^2} = 4\pi e^2 \rho_1 + \frac{V_{-1}}{12h^2} = \tilde{\rho}_1. \tag{2.9}$$

The proper system to solve will then be

$$AV = \tilde{\rho} \tag{2.10}$$

Where A is constructed as previously specified. Solving with $\tilde{\rho}$ on the right hand side will force the solution to abide boundary conditions.

On higher order approximations and performance

Higher and higher order approximations for derivatives involve more points that are further away. This increases accuracy by reducing the finite differences error, but it also decreases matrix sparseness.
Algorithms like Conjugate Gradient, as we'll see in the next section, and linear algebra computations, benefit from matrix sparseness, since multiplication by zero is readily known. The implication is that performance *vs* accuracy is a tradeoff that isn't univocal to every problem.
In the present work, the golden choice has been 5-point derivatives; but it's not definitive, as the lattice points rapidly cap depending on the system's memory.
As an example, take a seemingly harmless grid, made of 50 points in each direction. The resulting matrix will be $50 \times 50 \times 50 \times 2 = 2.5 \times 10^5$ both in columns and rows. It may be the case in the future, that higher order derivatives will be needed, to compensate for the limitation brought by the $\mathcal{O}(h^n)$ polynomial accuracy of the method in the step size.

2.2. Eigenvalue problem

This section is devoted to the approximate solution of the eigenvalue problem, by exploring the idea behind the Conjugate Gradient (CG) algorithm and its use in the General Conjugate Gradient (GCG).
Eigenvalue problems are ubiquitous in physics and engineering, and while solving one for a small matrix is trivial, it still requires roughly $O(n^3)$ [8] operations to do so. More often than not, real computational applications result in large-scale matrices, which are completely out of question for exact eigenvalues calculations, thus requiring the use of approximate algorithms.
We'll start by skimming through the basics of the most modern and used algorithms, by understanding shared numerical techniques and addressing their limitations, and then move on to the GCG.

2.2.1. Conjugate Gradient and numerical techniques

Iterative eigensolvers often share some common numerical techniques, like the Rayleigh-Ritz procedure, shift-and-invert, and approximate linear systems solvers. We will begin by describing these recurring techniques.

Conjugate Gradient and numerical techniques

Solving linear systems of the form

$$Ax = b \tag{2.11}$$

is crucial in many eigensolvers. The Conjugate Gradient (CG) is perhaps the most famous iterative solver in this sense, especially in regards to sparse matrices, as we'll see in a moment. CG applies to cases where A is an $n \times n$, positive-definite, symmetric matrix, and x and b are n-dimensional vectors.
Many generalizations to this method exist, which relax the requirements on the matrix, like BiCGSTAB, CGRES and so on [23]. We'll describe the working principle of CG, but the same applies to all the others, with slight variations.

Steepest descent method The quadratic form $f(x)$ derived from system 2.11 is

$$f(x) = \frac{1}{2}x^T Ax - b^T x \tag{2.12}$$

If A is symmetric, positive-definite, the shape of $f(x)$ is convex and has a global minimum for

$$\nabla_x f(x) = Ax_m - b = 0 \implies Ax_m = b \tag{2.13}$$

This implies that the extremum of the quadratic form is the also the solution of the linear system 2.11.
We can employ the well-known gradient descent technique [24] to find such point: starting from a guess x_0, we compute the direction where f decreases the most (the residual r_i), compute the step size for maximal decrease, and update x_i at each iteration accordingly, repeating until convergence.

$$d_i = r_i = = b - Ax_i \tag{2.14}$$

$$x_{i+1} = x_i + \alpha_i r_i \tag{2.15}$$

$$\text{with } \alpha_i \text{ such that } \frac{\mathrm{d}f}{\mathrm{d}\alpha_i} = 0 \implies \alpha_i = \frac{r_i^T r_i}{r_i^T A r_i} \tag{2.16}$$

This is a powerful but highly inefficient procedure. We are not ensuring that the search direction doesn't end up with components in subspaces that were explored already.
It can be proven [24] that the norm of the error $e_i = x_i - x_m$ is minimal at each iteration if the search directions d_i are chosen to be A-orthogonal to the next error, i.e. $d_i^T A e_{i+1} = 0$. This makes the algorithm converge at the exact solution in n steps, but most importantly it allows to truncate the iterations without a large error on the approximation x_i.

In this case, the algorithm is called Conjugate Gradient Method and is formulated as

$$\alpha_i = \frac{r_i^T r_i}{d_i^T A d_i} \tag{2.17}$$

$$x_{i+1} = x_i + \alpha_i d_i \tag{2.18}$$

$$r_{i+1} = r_i - \alpha_i A d_i \tag{2.19}$$

$$\beta_{i+1} = \frac{r_{i+1}^T r_{i+1}}{r_i^T r_i} \tag{2.20}$$

$$d_{i+1} = r_{i+1} + \beta_{i+1} d_i \tag{2.21}$$

Where iterations are truncated if the norm of the residual r_i is smaller than a certain threshold. CG represents a great method for sparse matrices, because it can be proven to be of complexity $O(m)$, where m is the number of non-zero elements in A [24].

Preconditioning

The CG method convergence is known to be limited by the modulus of the condition number of A [24], given by

$$\kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}. \tag{2.22}$$

If we were able to find a good *preconditioner* M, symmetric and positive-definite, such that $\kappa(M^{-1}A) \lll \kappa(A)$, and M^{-1} is easy to compute, then the algorithm would converge much faster, by solving $M^{-1}Ax = M^{-1}b$, since x is also the solution of $Ax = b$.

$$x = (M^{-1}A)^{-1}M^{-1}b = A^{-1}MM^{-1}b = A^{-1}b \tag{2.23}$$

Without delving the details of the preconditioner implementation, detailed in [24], note that, in general, $M^{-1}A$ is neither positive-definite nor symmetric, which requires a decomposition of the type $M = EE^T$ to be used.
The catch with preconditioning is that M has no unique recipe. Preconditioners are widely spread across numerical analysis, so many methods have been explored and implemented [18].

Rayleigh-Ritz procedure

A common denominator of all these algorithms is the search of good approximations for the correct eigenvectors in a certain subspace $\mathcal{K}$. The method is called Rayleigh-Ritz (RR) procedure [23], and is here outlined.
Suppose to have a matrix A of size $n \times n$, with entries in $\mathbf{C}$ and a collection of vectors

k which form a subspace $\mathcal{K} \subset \mathbf{C}^n$, where $\mathcal{K}$ is an $n \times k$ matrix. Generally speaking, n is large, while k is much smaller.
The best approximation of the true eigenvectors of A in $\mathcal{K}$ can be computed by solving the small scale eigenvalue problem

$$\mathcal{K}^\dagger A \mathcal{K} C = C\Lambda \tag{2.24}$$

Resulting in matrices $\mathcal{K}^\dagger A\mathcal{K}$ and C being of size $k \times k$. Computing $\mathcal{K}C$ gives a matrix of size $n \times k$, whose column vectors are the best approximations of the eigenvectors of A in $\mathcal{K}$, with their corresponding eigenvalues in the entries of the diagonal matrix Λ.

Shift and Invert

The power iteration is the technique on which Krylov subspace search methods are based [8]. By repeatedly applying matrix A to a vector x, it gets skewed towards the eigenvector whose eigenvalue is of largest magnitude λ_n.
Assume A is a hermitian matrix, thus diagonalizable. This means we can write an arbitrary vector $x^{(0)}$ as a linear combination of the eigenvectors $\{v_i\}$ of A.

$$x^{(0)} = \sum_i^n \alpha_i v_i \tag{2.25}$$

If we apply A to $x^{(0)}$ k times, we get

$$x^{(k)} = A^k x^{(0)} = \sum_i^n \alpha_i A^k v_i = \sum_i^n \alpha_i \lambda_i^k v_i \tag{2.26}$$

It can be proven that taking the ratio of the j-th component of $x_j^{(k)}$ and $x_j^{(k-1)}$ converges to λ_n

$$\lim_{k\to\infty} \frac{x_j^{(k)}}{x_j^{(k-1)}} = \lambda_n \tag{2.27}$$

which means, that for large enough k, we have the relation

$$Ax^{(k)} \approx \lambda_n x^{(k)} \tag{2.28}$$

So $x^{(k)}$ is an approximation of the eigenvector v_n of A whose eigenvalue is λ_n.

Smallest eigenvalue If instead of the largest eigenvalue, we were intereseted in the smallest one (in magnitude) λ_0, then we would need to apply the inverse matrix A^{-1} to

$x^{(k)}$, which would change the ratio 2.27 to

$$\lim_{k\to\infty} \frac{x_j^{(k)}}{x_j^{(k-1)}} = \lambda_0 \tag{2.29}$$

Assume for a moment that we're solving a nuclear single-particle hamiltonian, where we have a certain number of bound states of negative energy and a much larger number of unbound states with positive energy. In this case, the inverse power iteration would converge to the states which energy is closer to zero, avoiding the interesting ones on the bottom of the spectrum.
The solution is, before inverting, to shift the matrix by a quantity σ that is very close to the lowest eigenvalue we want to compute, call it λ_σ (eigenvector v_σ). Now, the eigenvalue of lowest magnitude of $(A - \sigma I)$ is $\lambda_\sigma - \sigma$ and by applying $(A - \sigma I)^{-1}$ to $x^{(k)}$, we'll get the approximation to the eigenvector v_σ.

2.2.2. Iterative eigensolvers

Now that the main techniques used by iterative eigensolvers have been laid out, we can look at three general methods, which are the most commonly used ones.

Jacobi-Davidson

The Jacobi-Davidson method [26] performs the RR procedure on a subspace which is enriched at each iteration by a correction to the previous eigenvectors.
Given an approximation (u, θ) of an eigenpair of matrix A, if the residual

$$r = Au - \theta u \tag{2.30}$$

is ≈ 0, then the eigenpair converged. Otherwise, we want to fine a correction t such that

$$r = A(u + t) - (\theta + \delta\theta)(u + t) = 0 \tag{2.31}$$

Linearizing this equation in t gives

$$(A - \theta I)t = -r \tag{2.32}$$

To avoid singularity of the equation near convergence, since u approximately spans a subspace of $\ker(A-\theta I)$, and enrich the subspace search with a useful orthogonal correction,

we project the problem onto the orthogonal subspace of u, which finally gives

$$(I - uu^\dagger)(A - \theta I)(I - uu^\dagger)t = -r \tag{2.33}$$

Algorithm 2.1 Jacobi-Davidson method for $Ax = \lambda x$

1: Choose normalized initial vectors $\{u_k\}$, set $V = [u_1, \ldots, u_k]$
2: **repeat**
3: Compute Ritz pair: $T = V^\dagger AV$, solve $Ty = \theta y$
4: Set $u = Vy$, residual $r = Au - \theta u$
5: **if** $\|r_k\| < \varepsilon \ \forall k$ **then**
6: **return** (θ, u)
7: **end if**
8: Solve approximately $(I - u_k u_k^\dagger)(A - \theta I)(I - u_k u_k^\dagger)t_k = -r_k$ using preconditioned iterative solver, ensuring $t_k \perp u_k$
9: Normalize: $v_k = t_k / \|t_k\|$
10: Expand subspace, setting $V = [V, v]$
11: **until** convergence for $k = 1, \ldots,$ nev

Although simple, this method is computationally efficient only by using preconditioning, which is known to be unstable in many cases [23].

Lanczos

Lanczos algorithm [13] is probably the most used iterative eigensolver in regards to hermitian matrices. It's a Krylov subspace search method, meaning the Rayleigh-Ritz procedure is done on a subspace formed as

$$\mathcal{K} = \{v_1, Av_1, A^2v_1, \ldots, A^{k-1}v_1\} \tag{2.34}$$

which exploits the power iteration. After orthogonalizing the new approximation to the previous one and diagonalizing the small scale problem, we end up with the new best approximations to the eigenvectors of A.

Algorithm 2.2 Lanczos Method for Computing nev Lowest Eigenpairs of Hermitian A

Choose normalized initial vector v_1, set $\beta_0 = 0$, $m =$ subspace size.
repeat
  for $j = 1, 2, \ldots, m$ **do**
    $w \leftarrow Av_j - \beta_{j-1}v_{j-1}$
    $\alpha_j \leftarrow v_j^* w$
    $w \leftarrow w - \alpha_j v_j$
    $\beta_j \leftarrow \|w\|$
    if $\beta_j = 0$ **then**
      break
    end if
    $v_{j+1} \leftarrow w/\beta_j$
  end for
  Form tridiagonal matrix $T_m = \text{tridiag}(\beta_{1:m-1}, \alpha_{1:m}, \beta_{1:m-1})$
  Compute eigen-decomposition $T_m y_k = \theta_k y_k$, for $k = 1, \ldots, \text{nev}$
  Form Ritz approximations $x_k = V_m y_k$, where $V_m = [v_1, \ldots, v_m]$
  Compute residual norms $r_k = \|Ax_k - \theta_k x_k\|$ for all k
until convergence for $k = 1, \ldots, \text{nev}$

Lanczos is extremely efficient, memory and CPU wise for extremal eigenvalues, but this limits its applicability to discretized unbound operators, where unwanted eigenvalues may dominate the Krylov subspace, and most importantly, for applications where the inner part of the spectrum is of interest, as in the case of Hartree-Fock-Bogoliubov (HFB).
A shift-and-invert strategy would be unfeasible in the case of large scale problems, since all Lanczos steps need to be performed exactly to avoid instabilities, a well known problem in the Arnoldi generalization [23].

LOBPCG

The last algorithm of this short list is LOBPCG, it's the newest and most sofisticated one of the three.
Introduced by A. V. Knyazev in 1991 [12], it's a block, preconditioned conjugate gradient method, explicitly targeted at solving large-scale eigenvalue problems, and it's been used in modern solutions of the Schrödinger/KS equation in recent years [14, 16, 17, 30]. In this algorithm, the Rayleigh-Ritz procedure is done on a subspace formed as

$$V = [X, W, P] \tag{2.35}$$

Where X is the current best eigenvectors approximation, P is the block of previous search directions and W is a block formed by preconditioning P.
We won't go into the details of LOBPCG, since GCG shares with it many aspects, like blocking and search directions calculation.
LOBPCG works very well for large scale problems, but it has limitations. For one, it's not possible arbitrarily select the portion of the matrix spectrum to calculate, which is required for problems where variational collapse happens, like in HFB or the Dirac equation, which manifests particle/antiparticle solutions [14]. To solve this, an additional filtering step is required [14, 16], which introduces a computational cost in the algorithm. Lastly, LOBPCG may fail when poor conditioning is present or when high precision on the eigenvalues is required [15].

2.2.3. General Conjugate Gradient

The General Conjugate Gradient is an iterative eigensolver designed with the aim of improving LOBPCG, it's a blocked algorithm, which uses the inverse power method and previous search directions to generate the search subspace. GCG is proven to be faster and more stable than LOBPCG [15].
A slightly different implementation of the algorithm is employed in the present work, detailed in algorithm 2.3, to improve applicability to HF calculations and reduce the computational cost.

Eigenvalue problem The original algorithm aims at solving the general eigenvalue problem $AX = \lambda BX$. Since in our case $B = I$, it is omitted from the algorithm, reducing the computational cost of P generation, and orthogonalization of V.

Blocking The algorithm allows to save converged eigenpairs (implemented) and work on a subset of the active eigenvectors. Since in a self-consistent calculation the matrix changes rapidly and at each HF iteration, it will be the case that the maximum number of iterations is reached before convergence, so we must work at all times on the remaining unconverged eigenvectors.

Orthogonalization The original paper [15] suggests an improved orthogonalization procedure; being beyond the scope of this work, the simpler Gram-Schmidt [1] orthogonalization is used in the present work.

Preconditioning The use of a preconditioner is beyond the scope of this work, a simple diagonal preconditioner is used.

Shift update The shift update is either fixed, in case of known spectrum, e.g. for HFB `shift` $= 0$, or adaptive [15], so that the inverse power step can find the correct eigenvalues, using the update formula 2.36

$$\texttt{shift} = (\lambda_{\text{nev}} - 100\lambda_1)/99 \tag{2.36}$$

Where λ_{nev} is the largest eigenvalue of the RR procedure and λ_1 is the smallest.

Convergence Convergence on an eigenpair is checked by computing the norm of the corresponding column vector of the residual matrix R and comparing it against a threshold.

Algorithm 2.3 GCG Algorithm

1: **Input:** Matrix A, number of desired eigenpairs `nev`, X_{guess} initial guess with $\text{col}(X_{\text{guess}}) = k \geq \text{nev}$, `max_iter` maximum number of iterations.
2: Initialize block $X = [X_{\text{c}}, X_{\text{a}}] \leftarrow [X_{\text{guess}}]$
3: Initialize blocks P and W with k null vectors
4: Solve the Rayleigh Ritz problem $X^{\dagger}AXC = C\Lambda$
5: Update $X = XC$
6: Initialize `shift`, Initialize `iter` $= 0$
7: **while** $\text{col}(X_c) <$ `nev` and `iter` $<$ `max_iter` **do**
8: Solve approximately $(A + \texttt{shift} \cdot I)W = X\Lambda$ with some CG steps, initial value X to generate W
9: Orthogonalize $V = [X, P, W]$
10: Solve the Rayleigh Ritz problem $V^{\dagger}(A + \texttt{shift} \cdot I)VC = C\Lambda$
11: Update $X_{\text{new}} \leftarrow VC$ and $\Lambda_{\text{new}} = \Lambda - \texttt{shift} \cdot I$
12: Compute the residual $R = AX_{\text{new}} - \Lambda X$
13: Check convergence on k-th column of R, update X_c and X_a accordingly
14: Update `shift` and `iter`
15: **end while**
16: **Output:** Approximate eigenpairs (Λ, X)

2.3. Code implementation details

In this last section regarding numerical methods, a few important generalities about the actual code implementation of the HF method are discussed. Mainly, the implementation of the Augmented Lagrangian Method to enforce spatial constraints on the HF

solution, the pseudocode of the entire self-consistent procedure, and the choice of optimal parameters for the functional minimization.

2.3.1. Constraints

The HF ground state is a great starting point to get theoretical results regarding many nuclear properties; however, for a plethora of applications, calculating the state of the nuclear system under certain constraints becomes necessary.
Effectively, a need to explore the energy surface arises. This means minimizing the energy functional, under a series of constraints of the type $\langle \mathcal{Q} \rangle = q_0$, where q_0 is the desired expectation value of an operator $\mathcal{Q}$.
This is an equality-constrained optimization problem (ECP), formulated as

$$\min_{|\Psi\rangle} E \tag{2.37}$$

$$\text{constrainted to } \langle \Psi | \mathcal{Q} | \Psi \rangle = \langle \mathcal{Q} \rangle = q_0 \tag{2.38}$$

Which yields the Lagrangian

$$E' = E + \lambda(\langle \mathcal{Q} \rangle - q_0) \tag{2.39}$$

where λ is a Lagrange multiplier determined by the condition $\langle \mathcal{Q} \rangle = q_0$. After finding the minimum of E', it's trivial to show that for a given λ, we get [7]

$$\frac{\mathrm{d}E}{\mathrm{d}\langle \mathcal{Q} \rangle} = -\lambda \tag{2.40}$$

If λ is properly adjusted, a certain value for q_0 can be obtained. From a numerical standpoint, λ needs to be tuned at each iteration to reach the desired value. This method was the one used in early constrained Hartree-Fock calculations [5].
Although the method is simple, it's often the case that it fails. Moreover, for the same value of λ, many, possibly infinite values of $\langle \mathcal{Q} \rangle$ can be obtained, for which we are only allowed to get the one with the most stable solution.
A different method is provided by the Quadratic Penalty Method (QPM). Briefly speaking, instead of a Lagrange multiplier, we add a quadratic contribution to the functional, such that

$$E' = E + \frac{c}{2}(\langle \mathcal{Q} \rangle - q_0)^2. \tag{2.41}$$

This is a straightforward method; intuitively one penalizes (hence the name), any solution for which $\langle \mathcal{Q} \rangle \neq q_0$ by increasing its energy. However, the success of such procedure is heavily influenced by the choice of c, often leading to instabilities for large values.

What happens is that for small values of c, the penalty may be insufficient to reach the target q_0, while for large values, the penalty may be so big that the self consistent calculation oscillates and fails.

Augmented Lagrangian Method

A modern, robust approach, used by HF/HFB codes [3, 22] is given in the form of the Augmented Lagrangian Method (ALM) [27]. Its main idea is to combine the precision of the ECP with the accuracy of the QPM.
Without delving into cumbersome mathematical details, we'll see how the algorithm is practically implemented in the code.
Given the functional

$$E' = E + \lambda(q - q_0) + \frac{c}{2}(q - q_0)^2 \tag{2.42}$$

where $q = \langle \mathcal{Q} \rangle$, the resulting mean field potential will be given by

$$U' = U + \lambda\mathcal{Q} + c(q - q_0)\mathcal{Q} \tag{2.43}$$

$$= U + c(q - q_0(\lambda))\mathcal{Q} \tag{2.44}$$

Where $q_0(\lambda)$ is updated at each iteration with the formula

$$q_0(\lambda) = q_0 - \frac{\lambda}{c} \tag{2.45}$$

$$\lambda^{(i+1)} = \lambda^{(i)} + \mu c(q - q_0) \tag{2.46}$$

Here, a slight deviation from the original ALM is present. Since the original work [27] doesn't provide guidance regarding what is considered an *iteration*, we employ the strategy [3] of using a damping factor $\mu \in [0, 1]$, so λ can be updated at each HF iteration for fast convergence, without large oscillations or instabilities of any kind.
This method is what powers the deformation curves that are shown in section (REF), allowing to explore the energy surface with arbitrary precision in reaching the value of q_0 at convergence, provided that enough HF iterations are performed.
Note that, since $\lambda^{(0)} = 0$, for $\mu = 0$ ALM reduces to the standard QPM.

2.3.2. Self-consistent procedure pseudocode

HF Code
- Parse user input
- Generate mesh
- Solve deformed Woods-Saxon

- Generate harmonic oscillator guess
- Diagonalize Woods-Saxon hamiltonian
- Normalize wavefunctions
- HF iterations
  - Solve BCS self-consistent equations
    - Identify time-reversal states
    - Compute pairing matrix elements
    - Solve for pairing gaps and chemical potential
    - Compute occupation factors v_i^2
  - Generate iteration data
    - Generate new densities, mix with previous ones
    - Generate mean fields
    - Generate constraint fields, mix with previous ones
  - Compute single-particle hamiltonian
  - Diagonalize h
  - Normalize wavefunctions
  - Compute energies, constraints residuals,
    - Hartree-Fock energy
    - Constraints residuals
  - Check convergence
    - Integrated energy
- Generate output
  - Compute multipole moments
  - Compute single-particle states quantum numbers
  - Write program output

2.3.3. Optimal parameters choice

For a given set of tolerances of the convergence criterion, the paramters that need to be chosen carefully are essentially the inverse power step tolerance and the number of maximum GCG iterations.

Inverse power step tolerance

The inverse power step tolerance is one of the biggest bottlenecks of the computational cost of the solver. A careful choice is needed for correct eigenvalues convergence, while mainting the computational cost at bay.
From figure 2.1, it's clear that at least a tolerance of 10^{-3} is needed for good convergence, while tolerances $\geq 10^{-4}$ stop offering increasing returns, rendering a choice between 10^{-4} and 10^{-5} an optimal one.

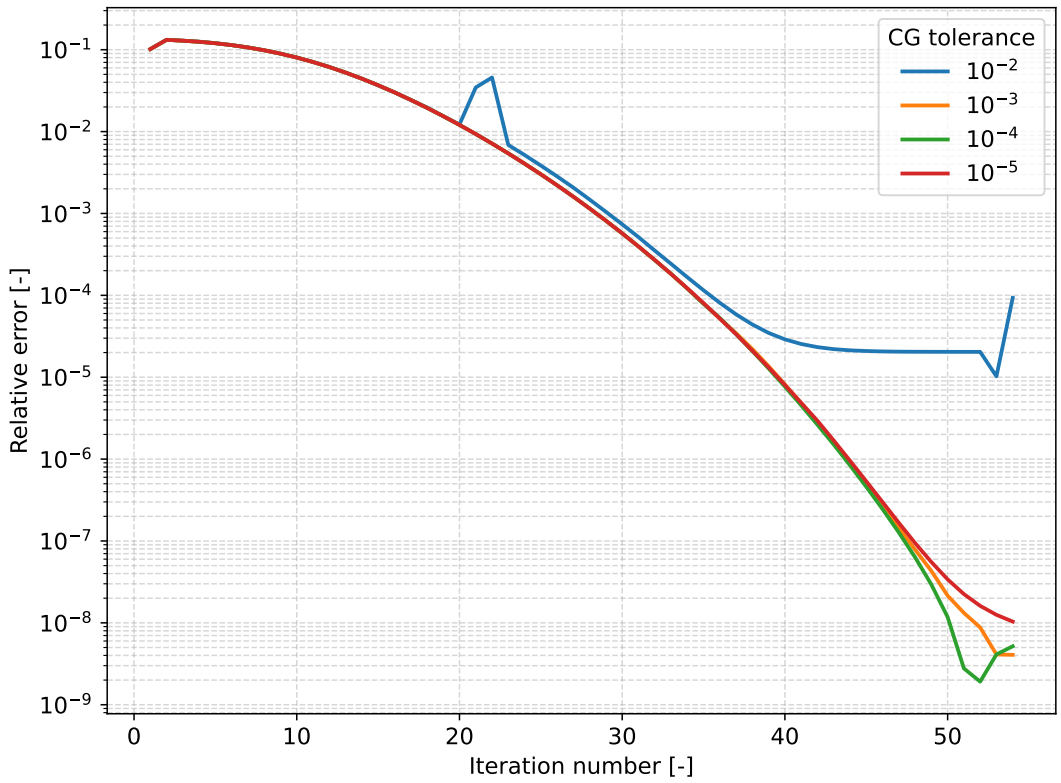

Figure 2.1: HF calculation convergence with varying CG tolerance. ^{16}O

Inner GCG iterations

The number of inner GCG iterations, here named 'inverse power steps' to avoid confusion, is slightly more nuianced than the CG tolerance. One could think that a higher number of iterations would bring to convergence faster since the precision on the eigenalues increases, but this is not the case.

In figures 2.2 and 2.3, the convergence of the HF calculation is plotted for different number of steps, respectively, for the spherical nucleus ^{16}O and the deformed nucleus ^{24}Mg.

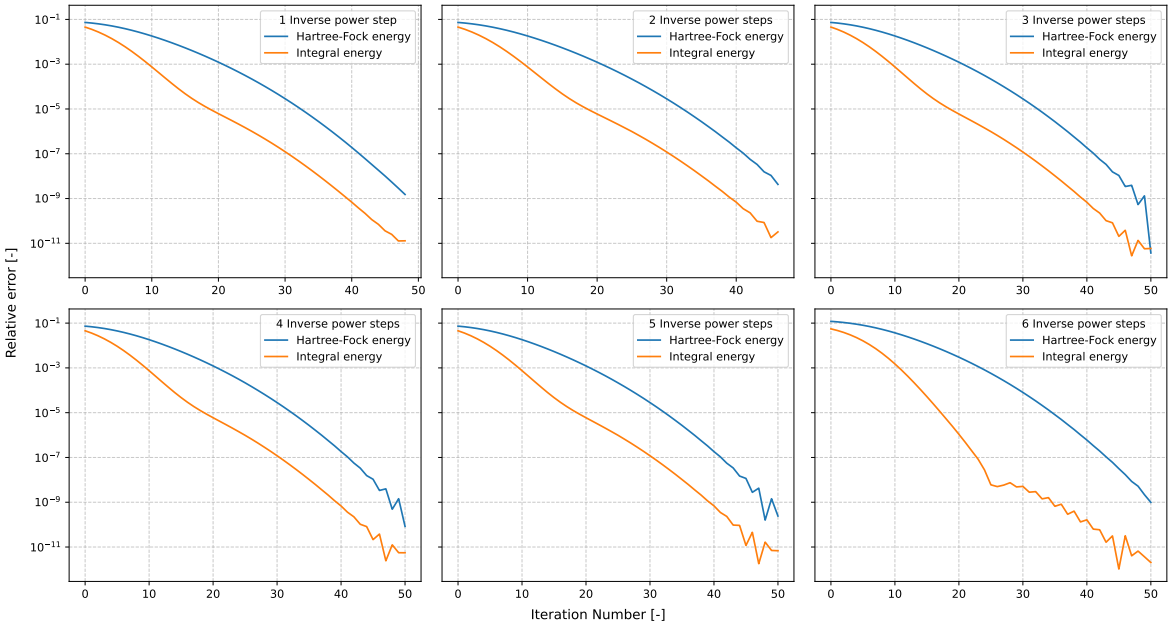

Figure 2.2: HF calculation convergence with varying number of inverse power steps for the spherical nucleus. ^{16}O

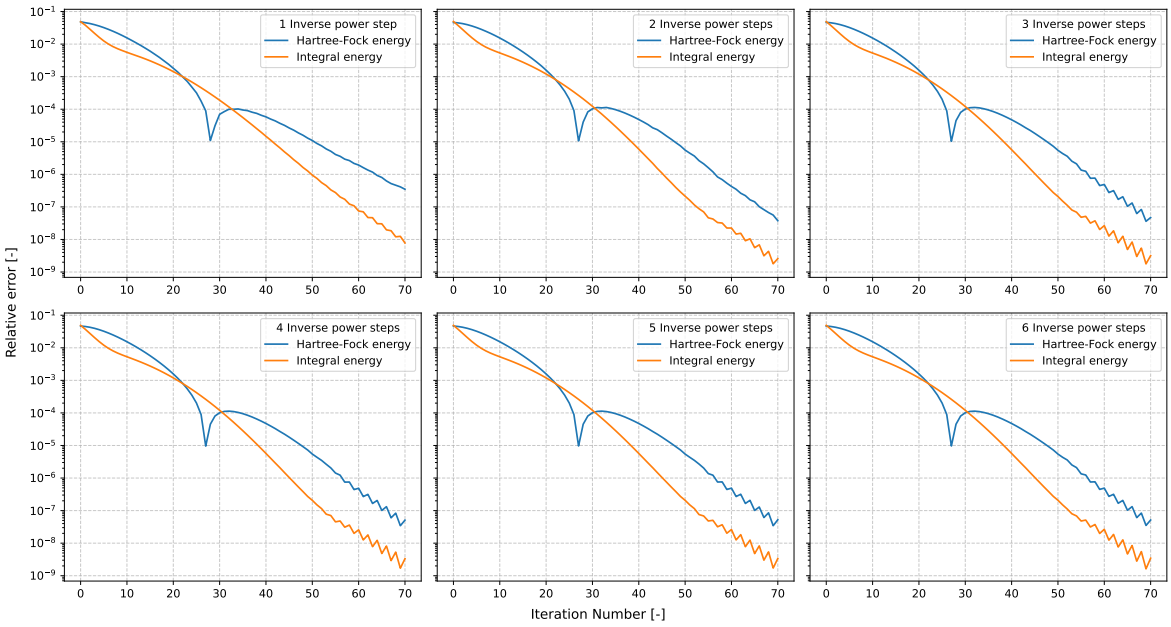

Figure 2.3: HF calculation convergence with varying number of inverse power steps for the deformed nucleus. ^{24}Mg

It's evident that in both cases, a steps number greater than 3 leads to a more unstable convergence, while in the case of the spherical nucleus, just one step is enough to quickly,

and safely, reach convergence.
This is likely due to the fact that at each HF iteration the hamiltonian changes and a great number of steps leads to solutions too biased towards the current matrix eigenvalues, at the expense of the next iteration; however, in the case of deformed nuclei, due to sharp shape changes at the start of the calculation, just one step may not be enough to sustain the pace at which the Hamiltonian changes, hence the quicker convergence with more steps.

Bibliography

[1] Å. Björck. Numerics of gram-schmidt orthogonalization. *Linear Algebra and Its Applications*, 197:297–316, 1994.

[2] E. Chabanat, P. Bonche, P. Haensel, J. Meyer, and R. Schaeffer. A skyrme parametrization from subnuclear to neutron star densities. *Nuclear Physics A*, 627(4):710–746, 1997. ISSN 0375-9474. doi: https://doi.org/10.1016/S0375-9474(97)00596-4. URL https://www.sciencedirect.com/science/article/pii/S0375947497005964.

[3] M. Chen, T. Li, B. Schuetrumpf, P.-G. Reinhard, and W. Nazarewicz. Three-dimensional skyrme hartree-fock-bogoliubov solver in coordinate-space representation. *Computer Physics Communications*, 276:108344, 2022. ISSN 0010-4655. doi: https://doi.org/10.1016/j.cpc.2022.108344. URL https://www.sciencedirect.com/science/article/pii/S0010465522000625.

[4] G. Colò. Nuclear density functional theory. *Advances in Physics: X*, 5(1):1740061, 2020. doi: 10.1080/23746149.2020.1740061. URL https://doi.org/10.1080/23746149.2020.1740061.

[5] R. Y. Cusson, P. G. Reinhard, M. R. Strayer, J. A. Maruhn, and W. Greiner. Density as a constraint and the separation of internal excitation energy in tdhf. *Zeitschrift für Physik A Atoms and Nuclei*, 320:475–482, 1985. doi: 10.1007/BF01415725. URL https://doi.org/10.1007/BF01415725.

[6] J. Erler, P. Klüpfel, and P.-G. Reinhard. Misfits in skyrme–hartree–fock. *Journal of Physics G: Nuclear and Particle Physics*, 37(6):064001, mar 2010. doi: 10.1088/0954-3899/37/6/064001. URL https://doi.org/10.1088/0954-3899/37/6/064001.

[7] H. Flocard, P. Quentin, A. Kerman, and D. Vautherin. Nuclear deformation energy curves with the constrained hartree-fock method. *Nuclear Physics A*, 203(3):433–472, 1973. ISSN 0375-9474. doi: https://doi.org/10.1016/0375-9474(73)90357-6. URL https://www.sciencedirect.com/science/article/pii/0375947473903576.

[8] G. H. Golub and C. F. van Loan. *Matrix Computations*. JHU Press, fourth edition, 2013. ISBN 1421407949 9781421407944. URL http://www.cs.cornell.edu/cv/GVL4/golubandvanloan.htm.

[9] H.-Q. Gu, H. Liang, W. H. Long, N. Van Giai, and J. Meng. Slater approximation for coulomb exchange effects in nuclear covariant density functional theory. *Phys. Rev. C*, 87:041301, Apr 2013. doi: 10.1103/PhysRevC.87.041301. URL https://link.aps.org/doi/10.1103/PhysRevC.87.041301.

[10] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, Nov 1964. doi: 10.1103/PhysRev.136.B864. URL https://link.aps.org/doi/10.1103/PhysRev.136.B864.

[11] J. D. Jackson. *Classical Electrodynamics*. John Wiley & Sons, New York, 3rd edition, 1998. ISBN 978-0471309321.

[12] A. V. Knyazev. *A Preconditioned Conjugate Gradient Method for Eigenvalue Problems and its Implementation in a Subspace*, pages 143–154. Birkhäuser Basel, Basel, 1991. ISBN 978-3-0348-6332-2. doi: 10.1007/978-3-0348-6332-2_11. URL https://doi.org/10.1007/978-3-0348-6332-2_11.

[13] C. Lanczos. Solution of systems of linear equations by minimized iterations. *J. Res. Nat. Bur. Standards*, 49(1):33–53, 1952.

[14] B. Li, Z. Ren, and P. Zhao. Efficient solution for the dirac equation in 3d lattice space with the conjugate gradient method. *Physical Review C*, 102(4):044307, 2020.

[15] Y. Li, H. Xie, R. Xu, C. You, and N. Zhang. A parallel generalized conjugate gradient method for large scale eigenvalue problems. *CCF Transactions on High Performance Computing*, 2(2):111–122, jun 2020. ISSN 2524-4930. doi: 10.1007/s42514-020-00029-6. URL https://doi.org/10.1007/s42514-020-00029-6. The corresponding computing package can be downloaded from the web site: https://github.com/pase2017/GCGE-1.0.

[16] L. Lin, S. Shao, and W. E. Efficient iterative method for solving the dirac–kohn–sham density functional theory. *Journal of Computational Physics*, 245:205–217, 2013. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2013.03.030. URL https://www.sciencedirect.com/science/article/pii/S0021999113002088.

[17] T. Nottoli, I. Gianni, A. Levitt, and F. Lipparini. A robust, open-source implementation of the locally optimal block preconditioned conjugate gradient for large eigenvalue problems in quantum chemistry. *Theoretical Chemistry Accounts*, 142

(8):69, 2023. ISSN 1432-2234. doi: 10.1007/s00214-023-03010-y. URL https://doi.org/10.1007/s00214-023-03010-y.

[18] J. W. Pearson and J. Pestana. Preconditioners for krylov subspace methods: An overview. *GAMM-Mitteilungen*, 43(4):e202000015, 2020.

[19] P.-G. Reinhard and H. Flocard. Nuclear effective forces and isotope shifts. *Nuclear Physics A*, 584(3):467–488, 1995. ISSN 0375-9474. doi: https://doi.org/10.1016/0375-9474(94)00770-N. URL https://www.sciencedirect.com/science/article/pii/037594749400770N.

[20] P. Ring and P. Schuck. *The nuclear many-body problem*. Springer Science & Business Media, 2004.

[21] L. M. Robledo, T. R. Rodríguez, and R. R. Rodríguez-Guzmán. Mean field and beyond description of nuclear structure with the gogny force: a review. *Journal of Physics G: Nuclear and Particle Physics*, 46(1):013001, dec 2018. doi: 10.1088/1361-6471/aadebd. URL https://doi.org/10.1088/1361-6471/aadebd.

[22] W. Ryssens, V. Hellemans, M. Bender, and P.-H. Heenen. Solution of the skyrme–hf+bcs equation on a 3d mesh, ii: A new version of the ev8 code. *Computer Physics Communications*, 187:175–194, 2015. ISSN 0010-4655. doi: https://doi.org/10.1016/j.cpc.2014.10.001. URL https://www.sciencedirect.com/science/article/pii/S0010465514003361.

[23] Y. Saad. *Numerical Methods for Large Eigenvalue Problems*. Manchester University Press, Manchester, UK, 1992.

[24] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. 1994. URL http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf.

[25] T. Skyrme. The effective nuclear potential. *Nuclear Physics*, 9(4):615–634, 1958. ISSN 0029-5582. doi: https://doi.org/10.1016/0029-5582(58)90345-6. URL https://www.sciencedirect.com/science/article/pii/0029558258903456.

[26] G. L. G. Sleijpen and H. A. Van der Vorst. A jacobi–davidson iteration method for linear eigenvalue problems. *SIAM Review*, 42(2):267–293, 2000. doi: 10.1137/S0036144599363084. URL https://doi.org/10.1137/S0036144599363084.

[27] A. Staszczak, M. Stoitsov, A. Baran, and W. Nazarewicz. Augmented lagrangian method for constrained nuclear density functional theory. *The European Physical*

Journal A, 46:85–90, 2010. doi: 10.1140/epja/i2010-11018-9. URL https://doi.org/10.1140/epja/i2010-11018-9.

[28] P. Stevenson and M. Barton. Low-energy heavy-ion reactions and the skyrme effective interaction. *Progress in Particle and Nuclear Physics*, 104:142–164, 2019.

[29] D. Vautherin and D. M. Brink. Hartree-fock calculations with skyrme's interaction. i. spherical nuclei. *Phys. Rev. C*, 5:626–647, Mar 1972. doi: 10.1103/PhysRevC.5.626. URL https://link.aps.org/doi/10.1103/PhysRevC.5.626.

[30] J. Woo, S. Kim, and W. Y. Kim. Gaussian-approximated poisson preconditioner for iterative diagonalization in real-space density functional theory. *The Journal of Physical Chemistry A*, 127(17):3883–3893, 2023. doi: 10.1021/acs.jpca.2c09111. URL https://doi.org/10.1021/acs.jpca.2c09111. PMID: 37094552.

[31] P.-b. Zhou. *Finite Difference Method*, pages 63–94. Springer Berlin Heidelberg, Berlin, Heidelberg, 1993. ISBN 978-3-642-50319-1. doi: 10.1007/978-3-642-50319-1_3. URL https://doi.org/10.1007/978-3-642-50319-1_3.

A | Appendix

A.1. Spherical harmonics

Spherical harmonics, of order λ, μ, are defined as

$$Y_{\lambda\mu}(\theta, \phi) = (-1)^{\mu} \sqrt{\frac{2\lambda + 1}{4\pi} \frac{(\lambda - \mu)!}{(\lambda + \mu)!}} P_{\lambda}^{\mu}(\cos\theta)\, e^{i\mu\phi}. \tag{A.1}$$

Being able to provide the expression for arbitrary μ, λ through an algorithm is important in the current framework, to solve the Poisson equation and investigate nuclear properties. The major challenge is to generate the associated Legendre polynomials P_{λ}^{μ}. They can be expressed in the form (for positive μ)

$$P_{\lambda}^{\mu}(x) = (1 - x^2)^{\mu/2} \frac{\mathrm{d}^{\mu} P_{\lambda}(x)}{\mathrm{d}x^{\mu}}, \tag{A.2}$$

where $x = \cos\theta$ and

$$P_{\lambda}(x) = \frac{1}{2^{\lambda}\lambda!} \frac{\mathrm{d}^{\lambda}(x^2 - 1)^{\lambda}}{\mathrm{d}x^{\lambda}}. \tag{A.3}$$

To compute the arbitrary λ, μ associated Legendre polynomial we can employ a recursive approach, setting $\lambda = \mu$

$$P_{\mu}^{\mu}(x) = (2\mu - 1)!!\, (1 - x^2)^{\mu/2}, \tag{A.4}$$

where $(2\mu - 1)!! = 1 \cdot 3 \cdot 5 \ldots (2\mu - 1)$ denotes the double factorial. Once $P_{\mu}^{\mu}(x)$ is known, the next element with $\lambda = \mu + 1$ reads

$$P_{\mu+1}^{\mu}(x) = x(2\mu + 1) P_{\mu}^{\mu}(x). \tag{A.5}$$

All higher orders are then generated using the standard upward recurrence relation in λ:

$$(\lambda - \mu + 1)\, P_{\lambda+1}^{\mu}(x) = (2\lambda + 1)\, x\, P_{\lambda}^{\mu}(x) - (\lambda + \mu)\, P_{\lambda-1}^{\mu}(x), \tag{A.6}$$

valid for all $\lambda \geq \mu + 1$.

A.1.1. Algorithm

1. Compute the base case P_{μ}^{μ} from the closed-form formula.
2. If $\mu = \lambda$ the procedure ends, otherwise
3. Evaluate $P_{\mu+1}^{\mu}$, if $\lambda = \mu + 1$ the procedure ends, otherwise
4. Apply the recurrence relation $P_{\lambda+1}^{\mu}$ until the desired degree is reached

This ought to be applied only for $\mu \geq 0$. For $\mu < 0$ the procedure is carried out using $-\mu$ and in the end using the relation

$$Y_{\lambda-\mu} = (-1)^{\mu} Y_{\lambda\mu}^{*} \tag{A.7}$$

A.2. 5-point derivatives

The first and second derivatives of a function $\psi(x)$ in $x = x_i$, using 5-points formulae, read

$$\psi'(x_i) = \frac{\psi_{i-2} - 8\psi_{i-1} + 8\psi_{i+1} - \psi_{i+2}}{12h} \tag{A.8}$$

$$\psi''(x_i) = \frac{-\psi_{i-2} + 16\psi_{i-1} - 30\psi_i + 16\psi_{i+1} - \psi_{i+2}}{12h^2} \tag{A.9}$$

List of Figures

List of Tables

List of Symbols

Variable	Description	SI unit
$\boldsymbol{u}$	solid displacement	m
$\boldsymbol{u}_f$	fluid displacement	m

Acknowledgements

Here you might want to acknowledge someone.