Tasks for "Nuclear density functional theory: Shapes and radii"

Wouter Ryssens
*Institut d'Astronomie et d'Astrophysique, Université Libre de Bruxelles, Campus de la Plaine CP 226, 1050 Brussels, Belgium**
(Dated: November 2022)

In this set of training exercices, you will study the deformation and root-mean-square (rms) radii of Uranium isotopes with nuclear energy density functionals. You will explore the link between deformation and nuclear radii by repeatedly solving the mean-field equations for different shapes and different nuclei for the time-tested SLy4 parameterization of the Skyrme EDF [3]. To do this, you will employ the EV8 code of Refs. [1, 2], which solves the Skyrme-HF+BCS equations on a three-dimensional code mesh. You will learn how to (i) initialize and run the code, (ii) judge its convergence and (iii) understand its outputs regarding total energy and nuclear shape.

I. PRELIMINARY STEPS

1. Get the code and all auxiliary files from https://github.com/wryssens/EV8. The `ev8` and `nil8` codes can be found in codes/, the relevant files for this training in LISA-training/.
2. Create a clean directory and copy the `ev8.f`, `nil8.f`, and `compile.sh` files there.
3. In order to compile the code, we will use the `compile.sh` bash script.
   - Verify that you have a FORTRAN compiler such as `gfortran` installed.
   - Optional: if you wish to use a compiler that is not gfortran, you should modify the script.
   - Execute the script and verity that you now also have `ev8.exe` and `nil8.exe` in your working directory.
4. Inspect the param8.h file. It tells you that we have compiled the code(s) to work with a Cartesian mesh with $mx = my = mz = 12$ points in every direction, with a $dx = 1.2$ fm spacing. Furthermore, we have selected to represent $mw = 220$ single-particle states. These conditions are generally not good enough to really accurately represent large atomic nuclei like Uranium isotopes. Those of you that want to do more realistic calculations can set $mx = my = mz = 18$, $dx = 0.8$ fm and $mx = 300$, but you should be prepared for long CPU times.

II. BASIC CODE OPERATION

The basic operation of both `nil8` and `ev8` is the following

```
./code.exe < input.data > output.out
```

where `input.data` is an input file you prepared and `output.out` is a file where you want to save the output. For the creation of the input data, I provide examples for both nil8 and ev8, for which you should modify only the relevant parts. As we will be running the code several times for different nuclei in different conditions, I strongly recommend you to (i) write a bash script to automate multiple code runs for multiple nuclei and (ii) to choose a clear labelling scheme for your outputs.

Aside from the input and output, running `nil8` also requires the presence of a file named `fort.12` in the working directory. This file contains a nuclear configuration and determines the starting guess of the self-consistent iterations. `nil8` serves to generate this starting point and will create a file named `fort.13`. It is sufficient to rename this file to `fort.12` to be able to start `EV8`. As an optional challenge, consider automating the gathering of information from the output files using, for example, the 'grep' bash command.

For more practical information on the code and its inputs, you can always refer to the slides of the training or the published papers, Ref. [1] and particularly the later sections in Ref. [2]. When you are truly stuck, you can always ask me at wryssens@ulb.be.

* wryssens@ulb.be

III. TASKS

1. Run `nil8` twice: once taking `nil8.prolate.data` and once taking `nil8.oblate.data` as input. After each run, verify that the code created a `fort.13` file and rename it with an appropriate name.

2. Run `EV8` twice for ^{238}U, each time for 100 iterations, starting from these starting guesses. Copy the parameters from `EV8.example.data` and modify the iteration numbers and particle number(s) accordingly. Verify that you are dealing with a prolate shape and an oblate shape depending on the initialization. Did these calculations converge? Repeat these calculations for larger amounts of iterations, until you are satisified with the convergence. Which configuration is the ground state?

3. Now repeat this process for very exotic proton-rich ^{218}U. What shapes do you obtain? Do calculations started from prolate/oblate configurations give different results? Did you expect this outcome?

4. Moving to the other extreme, repeat these calculations for the very neutron-rich ^{276}U. What do these results make you suspect about this particular number of neutrons?

5. Let us now do things a bit more systematically. Initializing everything with a prolate shape, perform a calculation for all even-even Uranium isotopes from ^{220}U up to ^{232}U. Combining these results with those you obtained for the ground states of ^{218}Pu, ^{234}U and ^{236}U, make separate plots of the following quantities as a function of neutron number: (i) rms charge radii, (ii) isotopic shifts with respect to ^{238}U and (iii) quadrupole deformation. Include experimental information on the radii and isotopic shifts from Ref. [4] and experimental information on the quadrupole deformation from Ref. [5]. Which one is better described, the rms charge radii or the isotopic shift? Why? Is the quadrupole deformation well described?

6. For the results you obtained in the previous question, plot the charge radius as a function of the quadrupole deformation. Why does this not depend perfectly quadratically on the deformation? Plot the hexadecapole (β_{40}) and hexacontetrapole (β_{60}) deformation as a function of the neutron number.

7. What about the neutron charge radii? Do they differ strongly from the charge radii? Plot the difference of the neutron and proton radii, also known as the 'neutron skin', as a function of neutron number. Are there also significant differences between the quadrupole deformation of protons and neutrons?

8. Repeat question 5, but this time initialize the calculations with oblate shapes. Make a combined plot with (i) the charge radii for the prolate solutions and (ii) the charge radii for the oblate solutions. What do you conclude?

Optional questions that will cost you a significant amount of computational time:

- Mapping the shell closures: perform calculations for all even-even nuclei from ^{212}U to ^{282}U. How do the charge radii behave?

- How do any of these results depend on the interaction? Change the parameterization from SLy4 to SLyIII$_{0.7}$ (`EV8` keyword= 'SLyIII0.7') [], and redo the calculations and plots of question 5.

---

[1] P. Bonche, H. Flocard, and P. H. Heenen, *Solution of the Skyrme HF+BCS Equation on a 3D Mesh*, Computer Physics Communications **171**, 49 (2005).

[2] 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 (2015).

[3] E. Chabanat, P. Bonche, P. Haensel, J. Meyer, and R. Schaeffer, *A Skyrme Parametrization from Subnuclear to Neutron Star Densities Part II. Nuclei Far from Stabilities*, Nuclear Physics A **635**, 231 (1998).

[4] I. Angeli and K. P. Marinova, *Table of Experimental Nuclear Ground State Charge Radii: An Update*, Atomic Data and Nuclear Data Tables **99**, 69 (2013).

[5] S. Raman, C. W. Nestor, and P. Tikkanen, *Transition probability from the ground to the first-excited 2+ state of even–even nuclides*, Atomic Data and Nuclear Data Tables **78**, 1 (2001).