CHE 5834/6834 Nuclear Engineering

Dr. Olga Palazhchenko
opalazhc@unb.ca

Assignment #1

Question 4 (Includes Coding Component) / 10

In this question, you'll be calculating the binding energies of the following seven isotopes using the SEMF equation (liquid drop model) and comparing these to their actual binding energies.

(a) Fill out the table below for each of the nuclides **(2 marks)**

Mark by deduction, -0.25 marks per incorrect individual entry, until 0/2 reached

Isotope	Number of Protons (Z)	Number of Neutrons (N)
^{1}H	**1**	**0**
^{3}He	**2**	**1**
^{4}He	**2**	**2**
^{6}Li	**3**	**3**
^{12}C	**6**	**6**
^{14}C	**6**	**8**
^{40}S	**16**	**24**
^{40}Ca	**20**	**20**
^{240}Pu	**94**	**146**

Coding Portion **/ 4**

(b) Using Python, write a code that calculates the binding energies of each of the nuclides from the list above. Automate as much of this calculation as possible.

Option #1: Traditional List Approach

One approach is to make lists with the input parameters (e.g., a list for the atomic mass numbers of all the isotopes in part(a) and another list with all the atomic numbers). An empty list can then be used to append the BE for each isotope as you loop through all of the isotopes, performing the SEMF calculation for each nucleus. E.g., using the zip function to loop through both Z and A lists, performing the BE calculation inside the loop, may be useful.

Option #2: periodictable Package Approach

If you feel like you have a good handle on using lists in Python and would like to try something a little more advanced in addition to this, try the following

Under Assignment 1 on D2L is an optional pdf guide to installing external packages, using the example of the package periodictable as an example ("Package Import Instructions"). The guide includes some useful examples of applying the periodictable package that may be handy here. You will still rely on lists to complete your BE calculations for now; however, the periodictable package will remove the need to manually input isotope properties such as Z, A, etc.

Note: there are no extra/bonus marks associated with selecting this option, and the content presented here will be covered by everyone in the next CE. This is only for those who are itching to do a little more now 😊.

There is no one correct approach to writing a functional code (using either option), as long as your code carries out the necessary calculations and is well-documented and clear. In writing your code, try to avoid repeatedly writing out the same commands. **(4 marks)**

Coding Portion **/ 2**

(c) Now that you've calculated the binding energies from the SEMF in part (ii), we want to compare the values to the actual binding energies of these isotopes.

Using Python, make a list of the actual (eg., from the Chart of the Nuclides) BEs for all the isotopes from part (a). Calculate the percent error between the true BEs and your calculations from the SEMF in part (b).

Add a print statement(s) to the end of your code that shows each isotope, its binding energy (calculated and actual), and the percent error. Make sure all binding energies are rounded to 3 decimal places, e.g., using Pu-240 as an example,

```
Isotope BE/A  BE/A (LDM) % error
      MeV      MeV
240-Pu   7.556035    7.541    0.199
```

(2 marks)

Final print statement should look something like this (see full coded up solutions by pulling from sample-code-solutions repository from with Eclipse):

```
Isotope BE/A  BE/A (LDM) % error
        MeV      MeV
1-H   0    -26.51    -100.0
3-He   2.572681    0.072    3473.168
4-He   7.073915    7.014    0.854
6-Li   5.332331    3.865    37.965
12-C   7.680144    7.441    3.214
14-C   7.520319    7.418    1.379
40-S   8.329325    8.29    0.474
40-Ca    8.551304    8.489    0.734
240-Pu    7.556035    7.541    0.199
```

(d) Are there any isotopes that have a greater difference between the calculated and actual binding energies than you would expect (especially when compared to similar isotopes)? Use your understanding of the liquid drop model and the nuclear shell model to try and explain these differences. It may be useful to consider each of the terms in the SEMF individually when justifying sources of error. **(2 marks)**

- H-1 has a zero actual binding energy as there is only 1 proton and no other nucleons inside this nucleus (meaning, it makes sense that there is no BE). The SEMF isn’t designed to predict BE for only 1 nucleon **(1 mark)**

Some other BE discrepancies to highlight (1 mark for noting H-1 discrepancy, then 0.5 per any of the following, up to 2 total):

- He-3 has a high % error, likely due to having a low number of nucleons overall. The SEMF is better at higher nucleon numbers overall. The BE is very underpredicted by the SEMF, which either overpredicts the surface term or the pairing term (the negative terms).

- This is a similar idea for Ca-40, where N= 20 and Z = 20, which are both magic numbers. SEMF will try to predict this, though it seems to underpredict. It does a better job with the nearby S-40, which doesn’t have dual magic #s of Z and N.

- Li-6 has a high percent error – this has a very underpredicted BE, also due to having a low number of nucleons or overcompensation from either of the neg. terms.

(1 mark for any two from the above list, or anything else that makes sense)