Exercises for “Nuclear reaction code TALYS”

D. Rochman, PSI

January 2024, n_TOF Nuclear Physics Winter School 2024

Contents

1 Exercise 1: simple cross section calculation with TALYS

In this exercise, we will calculate specific cross sections, using the nuclear reaction code TALYS. We will also compare our calculations to evaluated (recommended) values from international libraries, and to experiments.
This exercise will be performed with the teacher.

1. First download the tar file
   (before Jan 22):
   from https://www-nds.iaea.org/talys/, or from this location https://tendl.web.psi.ch/tendl_2021/talys.html;
   (after Jan 22):
   from https://www-nds.iaea.org/talys// (take for instance TALYS-1.97).
   The tar file is relatively large, and needs to be put in a dedicated directory. For instance, type the following

   ```
   mkdir talys_runs
   cd talys_runs
   ```

   Copy here the talys.tar file that you just downloaded, and then type

   ```
   tar -xvf talys.tar (or tar -xvf talys1.97.tar)
   ```

   The names of the directory are self-explanatory. The next step is to compile the code. The compilation assumes that a directory called /bin/ exists. This is the directory where all the executables are located. If not, either create one, or modify the script called "talys.setup" to copy the talys executable to another place. In the following, we will assume that the directory /bin/ exists.

   ```
   BEFORE JAN. 22
   cd talys
   ./talys.setup

   AFTER JAN. 22
   cd talys1.97
   (change the user name in the file "path_change)
   ./code_build talys
   ```

   The compilation will take a few minutes. If successful, you should find an executable called "talys" in your /bin/ directory. As mentioned, talys is a fortran code (like many codes in nuclear physics) and can be run with the following command:

   ```
   talys < input_file > ouput_file
   ```

   A number of samples exist in the directory called samples/. Running them all would take up to 2 hours. You may do that from home using the script called "verify" from the samples/ directory. Explanations for the samples can be found in the manual.

2. To get familiarized with talys, we are going to start with a simple example. You will need to create a new directory:

```
cd ..
mkdir test_nb
cd test_nb
```

In this directory, create a text file called "input_nb" and add the following lines:

```
#
# General
#
projectile n
element nb
mass 93
energy 14
```

Then run talys:

```
talys < input_nb > ouput_nb
```

Lines starting with a "#" are comments and are not read by the code. This example will perform the calculation of n+^{93}Nb with a neutron energy of 14 MeV. Open the file called "ouput_nb" and search for the section called "REACTION SUMMARY FOR E= 14.00000". Values of cross sections are presented (in mb). You can see the total cross section, the elastic cross section and reaction cross section.
Another way to present the calculated cross sections can be obtained with the addition of one line in the input file:

```
#
# General
#
projectile n
element nb
mass 93
energy 14
channels y
```

This additional line allows in fact for the calculation and output of all exclusive reaction channel cross sections. Run once again talys, but redirect the output to another file:

```
talys < input_nb > ouput_nb2
```

See the difference between the two output. What is the additional block ? It should correspond to specific cross sections for additional reaction channels.
A more convenient way to organize the calculated values is to have every single reaction cross section in a separated file. This is achieved by adding one line in the input:

```
#
# General
#
projectile n
element nb
mass 93
energy 14
channels y
filechannels y
```

Run again the input:

```
talys < input_nb > ouput_nb2
```

And see the additional files which were just created. Each file corresponds to a specific channel, and the name looks like *xsNPDTHA.tot*, where N is the neutron number of the exclusive channel, P the proton number, etc., in (a1) format.

3. It is important to understand that many models are used to perform these calculations, and these models are using a set of parameters, adjusted or not. In order to see which parameters were used, add the following line to your input:

   ```
   partable y
   ```

   This additional flag allows for all the model parameters used in a calculation on a separate file *parameters.dat*. This can be a very powerful option when one wishes to vary any nuclear model parameter in the input. The file parameters.dat has the exact input format, so it can be easily copied and pasted into any input file. This is helpful for a quick look-up of all the parameters used in a calculation. We have used this ourselves for automatic (random) TALYS-input generators for e.g. covariance data.
   Rerun talys and look at the file for parameters.
   Explanation for each parameter can be found in the manual, or discussed during the course.

4. Finally, we will repeat the above calculations but for different energies. You have seen the keyword called "energy" in the input, followed by the value of 14 (MeV). This line can be changed by "energy energy.file", where the file "energy.file" contains one column of incident energies (in MeV):

```
cat energy.file
1
5
8
9
10
11
12
13
14
15
16
17
18
19
20
22
24
```

You can change the input file with the new energy line, and rerun talys. Note that the calculation will take longer.
Look at the content of the file *xs200000.tot* with (n,2n) cross sections at various energies.
Go online in the EXFOR database and extract the measured ^{93}Nb(n,2n) cross sections. Download them in a text file and plot them with your calculations.
Do the same with the ENDF database. This last items can be done with the teacher. The final plot is as presented in Fig. 1.

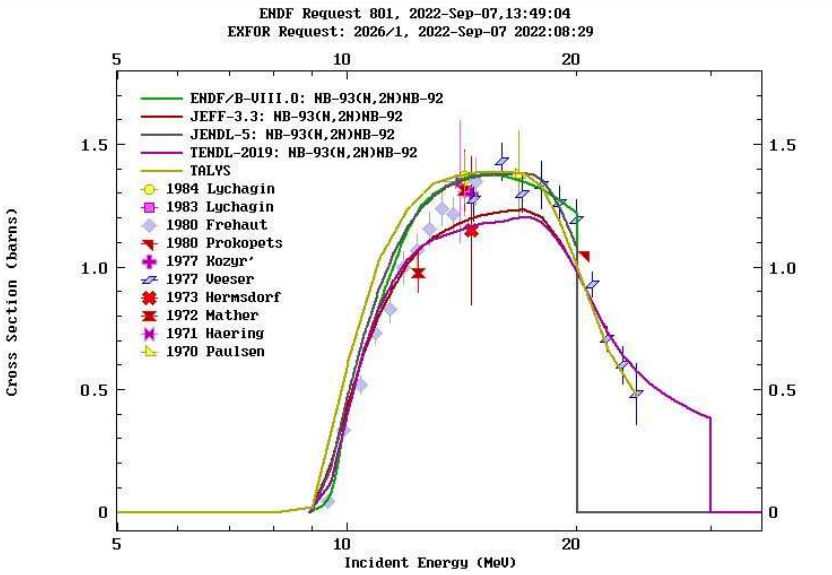

Figure 1: Example of the ^{93}Nb(n,2n) cross section.

2 Exercise 2: Modified cross section calculation with TALYS

We will continue with the previous example and the previous TALYS input file. As mentioned, the file *parameters.dat* contains the list of input parameters used by TALYS. These parameters concerns for instance the level densities of the compound nucleus, as well as of many residual nuclei. Changing some of these parameters will modify some cross sections, and let other ones unchanged.

1. One of the most important parameter is the parameter g_ν, being the single-particle neutron level density parameter in 1/MeV for the compound nucleus. You can modify it by 10 % by adding one line in the TALYS input file for the parameter "gnadjust"

   ```
   gnadjust          41  94   1.10000
   ```

   First copy the previous result for the (n,2n) cross section and then rerun talys with the updated input file

   ```
   cp cp xs200000.tot xs200000.tot.org
   talys < input_nb > ouput_nb2
   ```

   Compare both files xs200000.tot and xs200000.tot.org. Estimate the change of the (n,2n) cross section. Is the effect linear ?
   Plot both calculated (n,2n) cross section and compare with experimental data.

2. Another possibility is to change the type of model used in the calculation. In the previous example, the same model was used, but with different parameters. This time, one can change the level density model. TALYS contains 6 different level density models:

   ```
   ldmodel 1: Constant Temperature + Fermi gas model (CTM)
   ldmodel 2: Back-shifted Fermi gas Model (BFM)
   ldmodel 3: Generalised Superfluid Model (GSM)
   ldmodel 4: Skyrme-Hartree-Fock-Bogoluybov level densities from numerical tables
   ldmodel 5: Gogny-Hartree-Fock-Bogoluybov level densities from numerical tables
   ldmodel 6: Temperature-dependent Gogny-Hartree-Fock-Bogoluybov level densities from
   numerical tables
   ```

   The default is the model number 1. You can see the effect of changing level densities by commenting the line with "gnadjust" (add a "#" in front of the line) and add for instance

   ```
   ldmodel 2
   ```

   Rerun TALYS and compare the (n,2n) cross section with the previous one. Try all six models and plot all six cross sections, together with experimental data.

3. If time allows, one can apply random variations to model parameters. Coming back to the question 1., we have seen that the parameter called "gnadjust" has a strong influence on the (n,2n) cross section. You can build a little script, where you change the same parameter *n* times, each time with a different value (for instance within 5 %). Perform 100 of these changes (by script, not by hand), and plot the calculated cross sections. Calculate the average values, as well as the uncertainty, and an energy-energy correlation matrix.

4. Similar to number 3., apply random variations to all parameters from the file *parameters.dat*. Same questions (plot, average, uncertainties and correlation).