

This directory contains the code `Kmatrix.1.py` used to implement Eq. (1-4) for calculating statistical reaction cross sections and branching ratios. The code computes the modulus squared of the diagonal S-matrix elements, summing over individual channels in each decay mode. The output is a table of cross sections as a function of energy in a fixed energy interval.

Also included are scripts to execute the code and to analyze the outputs. All codes are written in the Python programming language using python 2.7 together with the `numpy` numerical library. Enter on the command line `$python <name>.py`

to execute a script `<name>.py`.

There are two parameter sets governing the reaction physics. One is the T_f , the nominal sum of transmission coefficients for each type of decay mode

$$T_f = 2\pi \frac{\Gamma_f^K}{D}. \quad (1)$$

Here Γ_f^K is defined in Eq. (5) and D is the average resonance spacing. The other reaction parameter set is N_f , the effective number of channels contributing to each decay mode. Besides that there are several purely numerical parameters. The most important are N_d , the dimension of the space used to construct the GOE; and dE , the energy steps in the calculations. N_d must be large enough so that only the central portion of the GOE is used. dE must be small enough so that narrow resonances are properly included when average cross sections are extracted.

The final output of the scripts is the ratio of cross sections, averaging them over some energy interval. The energy interval is computed in segments within the scripts. The calculation is repeated an number of times to esti-

mate the variance of the computed ratio.

The scripts `example1.py` to `example4.py` illustrate the physics with some simple examples. Besides the entrance channel, the examples treat two decay modes a and b.

Example 1. $(Ta, Na) = (2, 10)$, $(Tb, Nb) = (2, 20)$, $T_e = 0.1$

This illustrates the Hauser-Feshbach limit: The equality $T_a = T_b$ implies equal probabilities for both exit channels.

Example 2. $(Ta, Na) = (1, 10)$, $(Tb, Nb) = (2, 20)$, $T_e = 0.1$

Same as above, but unequal T_f in the exit channels.

Example 3. $(Ta, Na) = (2, 10)$, $(Tb, Nb) = (2, 20)$, $T_e = 1.0$

Same as example 2, but with a strong coupling to the entrance channel.

Example 4. $(Ta, Na) = (2, 10)$, $(Tb, Nb) = (2, 1)$, $T_e = 0.1$

This example illustrates the suppression of the branching when $T_f > N_f$. The script `example4.py` gives 0.37 ± 0.04 , much different from the Hauser-Feshbach value of 1.

Example 4M. The branching ratio with the parameters of Example 4, calculated in the Hauser-Feshbach theory including the Moldauer width correction Eq.(8). It gives $\sigma_a/\sigma_b = 0.56$, almost a factor of 2 larger than Example 4.

Finally, the script `K29.py` is one of those used to compute the branching ratios presented in the article. This particular script computes the model A value and variance at $\Gamma_F^K = 420$ meV as shown in Fig. 1. It takes 20 samples of the GOE averaging the cross sections over a range of 1 keV. That interval contains about 1000 resonances for each of the two independent entrance channels. It requires about 2 hours to run on an laptop computer.