Calculating the excitation energies for the wobbling spectra of ^{135}Pr

Author: Robert Poenaru

E-mail: robert.poenaru@drd.unibuc.ro

Least-squares-fit procedure for getting the E_{RMS} and the free parameters

Details on the problem

- The model has 5 free parameters:
  - 3 inertia factors: A_1, A_2, A_3
  - the single particle potential strength V
  - the coupling angle θ
- The experimental data set contains:
  - The yrast band Y0
  - The *one-phonon* wobbling band TW1
  - The (new) *two-phonon* wobbling band TW2. Experimentally confirmed in the recent work of Sensharma et al.[1]

Structure of the code

This `C++` project is structured as follows:

- The **include** folder has the necessary header files for serving the experimental data
  - `energyFormulas.h` - Calculating the energies for each band (numerical expressions) as a function of the input parameters
  - `expData.h` - Serves the experimental data to the main algorithm. Experimental data consists in the energy states for the three triaxial bands (namely Y0,TW1 and TW2). Units are in MeV (if not, a transformation method will be provided in the header). **Algorithm**

[1] Sensharma, N., et al. "Two-phonon wobbling in 135Pr." Physics Letters B 792 (2019): 170-174.

works with EXCITATION energies. So the first state from Y0 will be subtracted.
  - `fitProcedure.h` - The methods for calculating an RMS as a function of the free parameters and also a minimization procedure to find the best RMS value.
- The **cc** directory which includes:
  - `energyFormulas.cpp`
  - `main.cpp`
  - `fitProcedure.cpp`

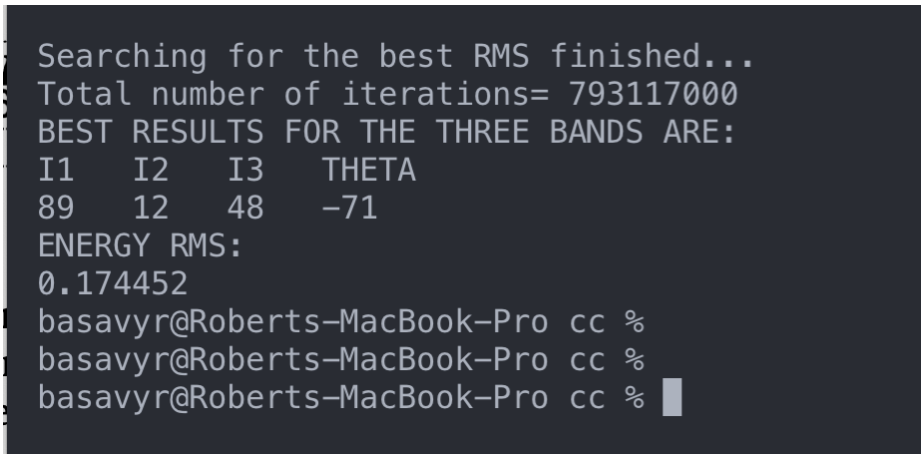
Time measurement

The main `Fit` class has an instrument for measuring the time execution of any object which is instantiated by the user through a smart point (i.e. *unique pointer* or *shared pointer*)

- `make_unique`
- `make_shared`

The class `Fit` -> private variable to store the start point (when object is created via the default constructed). The method `measureTime()` gives the total execution time of the class object on the main thread when it stops (through the default destructor).

Potential fit results



These are the fit parameters obtained within the new `C++` algorithm which includes the confirmed two-phonon wobbling band `TW2`.

Searching for the best RMS finished...
Total number of iterations= 793117000
BEST RESULTS FOR THE THREE BANDS ARE:
I1 I2 I3 THETA
89 12 48 -71

ENERGY RMS:
0.174452

Observation:

It seems that for the obtained fit parameters, the wobbling frequencies calculated for a chiral transformation (i.e. rotated system with an angle π), the quantity under the square root in the expression of ω^I becomes negative, which results in a **complex value > For these parameters, small spins might provide non-physical solutions!**

Potential fix for the chiral frequencies

The wobbling frequency calculated by the current algorithm has only one value which depends on the coupling angle θ.
However, this wobbling frequency has a *chiral* partner, defined by the geometrical rotation with the angle π around the quantization axis. So, having the normal mode wobbling frequency ω_θ will result in the chiral partner

$$\omega^{\text{chiral}} = \omega_{\theta+\pi}$$

In our results, the corresponding chiral partner of $\theta = -71^o$ would be $\theta^{\text{chiral}} = 109^o$. > Solution

Find the set of parameters $\mathbf{X}$ fot which the tuple $(\omega, \omega^{\text{chiral}})$ are real and positive numbers, so that the parameters will provide physical solutions to the problem;

New method for returning the wobbling frequency

Define a tuple `omegaTuple` which returns a `struct` object `omegas`, that is of the type `omegas.omega` and `omegas.omegaChiral`

```
struct omegaTuple
{
  double omega;
  double omegaChiral;
};
//initialize the tuple for storing the two values of the frequencies.
auto omegas= new omegaTuple();
```

introduce the object `omegas` in the fit condition so that both struct members are REAL and positive numbers