`tri_min_T` User's Guide

Curro Pérez Bernal <`francisco.perez@dfaie.uhu.es`>

Version 2.0

Contents

Chapter 1

Introduction

1.1 What's `tri_min_T`?

The `Fortran` program `tri_min_T` models bending dynamics of linear, quasilinear and bent triatomic molecules, using an algebraic model based on the U(3) Lie algebra. This program, includes fourteen adjustable parameters, including the most general one-, two-, three- and four-body Hamiltonian.

1.2 Copyright statement

Chapter 2

Installation guide

The program files can be found as a `tgz` file (`triat_U3_min_2.`*X*`.tgz`, where *X* stands for the version number). The first step is to unpack the file:

```
$ tar xzf triat_U3_min_2.0.tgz
$ ls
triat_U3_min_2.0  triat_U3_min_2.0.tgz
$ cd triat_U3_min_2.0/
triat_U3_min_2.0$ ls
bin/  doc/  examples/  README  src/
```

The directory structure is the following

1. `bin`: Executable files.
2. `src`: `Fortran` source files and compilation `Makefile`.
3. `doc`: Program documentation (this file) in several formats.
4. `examples`: Two different examples of application of the program.
    - `XCNO`: Application to the large amplitude bending mode of fulminic acid.
    - `HCP_A`: Application to the bending mode of the A excited electronic state of methinophosphide.

2.1 Program compilation

The present section details the program compilation procedure. There is also a version of the program compiled statically in the `bin` directory (`tri_min_Tstat`). If you run into trouble during the program compilation you can move forward to 'Getting Started' on page 5 using the provided static version of the program.

The program compilation in `Debian` systems should be quite smooth provided that the libraries `BLAS`, `LAPACK` and `PACKLIB` are installed.

The first two are `FORTRAN` libraries for numerical linear algebra and can be found in most Linux Distributions. In the `Debian (Lenny)` distribution they are associated with packages `libblas-dev` and `liblapack-dev` or other (`atlas` library). More information about `BLAS` and `LAPACK` can be found in the Netlib (`http://www.netlib.org`) website.

The third library is part of the CERN scientific routines, providing the `Minuit` minimization tool. This library in `Debian` can be installed with the packages `libpacklib1-dev` and `libkernlib1-dev`, though it is safer to install the `cernlib-base-dev` package.

Once the libraries are installed the compilation is quite direct.

```
triat_U3_min_2.0$ cd src
triat_U3_min_2.0/src$ ls
assign_gen.f             Makefile          so3casimirW2.f
chisqre_u3_min_gen.f     maxc.f            so3casimirW2WB2.f
cpexp.f                  minuit_u3.f       so3casimirW4.f
disdata.f                readenerg_gfort.f
fcn_sub_gen.f            scaleh_gen.f
hbldu3_gen.f             selectvl.f
triat_U3_min_2.0/src$ make clean
triat_U3_min_2.0/src$ make all
```

As can be seen in the `Makefile`, the `tri_min_T` executable file is located under the `triat_U3_min_2.0/bin` directory.

There is also a commented section in the `Makefile` (named `triat_min_stat`) used to compile the static version of the program. In case you want compile the program statically uncomment the three lines and type `make triat_min_stat`. Please, notice that in this case it is convenient to make a backup of the provided statically linked program (`tri_min_Tstat`) because you will most likely delete it.

If the program works correctly you can also, running as *root* user, install it, by default under the directory `/usr/local/bin`. The Perl launch script will be also installed in this directory. It is convenient to add to the default `$PATH` variable the `/usr/local/bin` in case it is not yet in it. This greatly facilitates the program's execution instructions in 'Program Execution' on page 7 and 'Examples' on page 8, because in this case the relative paths are not necessary. In order to install it you should do it as superuser or using `sudo`

```
triat_U3_min_2.0/src$ sudo make install
[sudo] password for curro:
triat_U3_min_2.0/src$ ls  /usr/local/bin/
minuit_U3_genH*  tri_min_T*  tri_min_Tstat*
```

Chapter 3

Getting Started

Under the directory `triat_U3_min_2.0/examples` two examples can be found, one for a linear molecule (fulminic acid fifth normal vibrational degree of freedom) and one for a bent molecule (magnesium hydroxide) are provided. We explain in detail the first case and chiefly highlight the differences that arise in the second case.

3.1 Input Files and Program Execution

We proceed now to explain the different input files and the way the program is executed. There are three necessary input files. A file with the experimental energy files to be fit, a second file with general information for the program and a third file with the Hamiltonian parameter information necessary to accomplish the `Minuit` minimization.

3.1.1 Experimental Energy File

The experimental energies have to be included in file. This file starts with a first line which is an integer equals to the number of available experimental levels. Thereafter the experimental information is given, using the following format:

```
energy  error   n_quanta  l_value
```

The fields `n_quanta` and `l_value` indicate the number of quanta of vibration and the vibrational angular momentum that characterize the state. The number of quanta could correspond to the labeling for linear or bent molecules, as indicated later in the input file.

3.1.2 General Input File

The general information not referring to Hamiltonian parameters is given in an input file with a `NAMELIST` structure. It provides the information needed to run the program apart from the information directly given to `Minuit` for the optimization.

We proceed to check the meaning of the variables are defined in this general input file. A template of it is as follows:

```
#
## GENERAL INPUT
#
 &INP0 BENT=.F., DTFL='expdata.dat' /
 &INP1 N2=140, LMAX=5, VMAX=5, EMINL=.F. /

#
 &INP2 IPRINT=0 /
```

Variables defined:

1 `BENT`: *logical* type. If `.T.`(`.F.`) the molecule is bent(linear) and experimental data are given accordingly.

2 `DTFL`: *character* type. Name of the file with experimental data.

3 `N2`: *integer* type. N value for the totally symmetric U(3) representation. As version 2.0 the maximum possible N value is 4000.

4 `LMAX`: *integer* type. Maximum value of the vibrational angular momentum (SO(2) quantum number) considered.

5 `VMAX`: *integer* type. Highest experimental overtone included in the fit.

6 `EMINL`: *logical* type. If this variable is `.T.` the spectrum energies are referred to the first eigenvalue for each vibrational angular momentum block. The default behavior is for `EMINL=.F.`, that implies that all the eigenvalues are referred to the energy of the ground state for vibrational angular momentum zero.

7 `IPRINT`: *integer* type. Program output verbosity. Mainly for debugging purposes. Possible values in the interval 0-5.

3.1.3 Minuit Parameter Input File

The information provided to `Minuit` for the algebraic spectroscopic parameter optimization is given in a file. The structure of this file is a three column list including the parameter labels, its initial values and initial minimization steps for the fourteen possible Hamiltonian parameters. This is followed by a statement using the `Minuit` command `fix` to indicate what parameter should be kept constant at the initial value. The file ends with the `Minuit` commands for the optimization. More information can be found in the Fortran Minuit Manual (`http://wwwasdoc.web.cern.ch/wwwasdoc/minuit/minmain.html`).

```
SET TITLE
'MINUIT MINIMIZATION. BENDING DYNAMICS'
PARAMETERS
1     'P11 '  6.129D+02         0.1D-02
2     'P21 '  9.9D+00           0.1D-02
3     'P22 '  0.30D+00          0.1D-02
4     'P23 '  -1.03D+00         0.1D-02
5     'P31 '  0.0D+00           0.0D+00
6     'P32 '  0.0D+00           0.0D+00
7     'P33 '  0.0D+00           0.0D+00
8     'P41 '  0.0D+00           0.0D+00
9     'P42 '  0.0D+00           0.0D+00
10    'P43 '  0.0D+00           0.0D+00
11    'P44 '  0.0D+00           0.0D+00
12    'P45 '  0.0D+00           0.0D+00
13    'P46 '  0.0D+00           0.0D+00
14    'P47 '  0.0D+00           0.0D+00

fix  5 7 8 9 10 11 12 13 14
#set err 1.0D-05
minimize 3000
call 3
exit
```

The parameters have labels `P`nm, where $n = 1,2,3,4$ indicates that the operator is n-th body, and m is an index to distinguish between the different operators of the same order. From the Hamiltonian building subroutine `hbldu3_gen.f`:

```
H = P11 n +
    P21 n^2 + P22 l^2 + P23 W^2 +
    P31 n^3 + P32 n·l^2 + P33 (n·W^2 + W^2·n) +
    P41 n^4 + P42 n^2·l^2 + P43 l^4 + P44 l^2·W^2 +
    P45 (n^2·W^2 + W^2·n^2) + P46 W^4 +
    P47 (W^2·Wbar^2 + Wbar^2·W^2)/2
```

where `n` is the U(2) number operator, `l` is the vibrational angular momentum, `W^2` is the second order Casimir operator associated so SO(3), and `Wbar^2` is the Casimir operator of the SObar(3) subalgebra.

3.1.4 Program Execution

The `Perl` script `minuit_U3_genH`, under the directory `triat_U3_min_2.0/bin`, is provided as a convenient interface to launch the `Fortran` program. It creates a temporary directory where the program runs and, in this way, several instances of the program can be executed simultaneously.

If the script is executed with no argument it displays a brief help message.

```
triat_U3_min_2.0/examples/XCNO$ ../../bin/minuit_U3_genH

 Usage: ../../bin/minuit_U3_genH [-vts] [-e prgfile]  \
minpar_file input_file output_file

 Options:
          -v: verbose output
          -t: do not remove temporary directory
          -s: static program version
          -e: explicit minimization program given
```

The important options are `-e` and `-t`. The first one implies that the user will provide the name (and path) of the executable program file to be launched, while the second option indicates -mainly for debugging purposes- that the temporary directory `tmpdir_XXXX` is not going to be removed once the program execution finishes. The option,`-s`, implies the use of the statically linked program `tri_min_Tstat`. Finally, the option `-v` forces a verbose output.

The script is invoked with three arguments. The first, *minpar_file* is the file with the parameter input for `Minuit` (see 'Minuit Parameter Input File' on page 6). The second argument, *input_file*, is the general input (see 'General Input File' on page 5). The last argument is *output_file*, the name of the file with the program output. See 'Examples' on the current page to check the application of the program to two cases, one corresponding to a linear molecule and the other to a bent molecule.

3.2 Examples

3.2.1 A Linear Molecule Example: Fulminic Acid Large Amplitude Bending

The files necessary to run this example are located under the directory `examples/XCNO`

```
triat_U3_min_2.0$ cd examples/XCNO/
triat_U3_min_2.0/examples/XCNO$ ls
exp_FulminicD_Feb03.dat   minuit_genpar_FulminicD
exp_FulminicH_Feb03.dat   minuit_genpar_FulminicH
input_U3_minuit_FulminicD output_FulminicD.orig
input_U3_minuit_FulminicH output_FulminicH.orig
```

The files `exp_FulminicH_Feb03.dat` and `exp_FulminicD_Feb03.dat` contain the available experimental information for the large amplitude bending vibration of fulminic acid and its deuterated isotopologue, and they conform to the standard described in 'Experimental Energy File' on page 5. The general input file (see 'General Input File' on page 5) in this case is called `input_U3_minuit_FulminicH` (and `input_U3_minuit_FulminicD` in the deuterated molecule case).

The information for the algebraic spectroscopic parameter optimization is provided in files `minuit_par_FulminicH` and `minuit_par_FulminicD` for HCNO and DCNO, respectively.

The program is executed invoking the `Perl` script with the already mentioned three arguments.

The verbose execution of the program for the fulminic acid case for both species is launched with the following commands:

```
triat_U3_min_2.0/examples/XCNO$ ../../bin/minuit_U3_genH -v \
> -e ~/triat_U3_min_2.0/bin/tri_min_T \
> minuit_genpar_FulminicD input_U3_minuit_FulminicD output_FulminicD

Minuit CHISQRE minimization. U(3) General Hamiltonian.
Making temporary directory ... Done.
Experimental energy file name is: exp_FulminicD_Feb03.dat
Copying files to temporary directory ... Done.
Running optimization program ... Done.
Copying output file from temporary directory ... Done.
Removing temporary directory ... Done.
triat_U3_min_2.0/examples/XCNO$
triat_U3_min_2.0/examples/XCNO$ ../../bin/minuit_U3_genH -vs
minuit_genpar_FulminicD \
> input_U3_minuit_FulminicH output_FulminicH

Minuit CHISQRE minimization. U(3) General Hamiltonian.
Making temporary directory ... Done.
Experimental energy file name is: exp_FulminicH_Feb03.dat
Copying files to temporary directory ... Done.
Running optimization program ... Done.
Copying output file from temporary directory ... Done.
Removing temporary directory ... Done.
```

In the HCNO case the static version of the program is employed. The final part of the output in these two cases, once convergence is reached, is included in files `output_Fulminic_HCNO.orig` and `output_Fulminic_DCNO.orig`.

Notice that the quality of the fit is given in terms of the standard deviation and not the rms. To compute the rms of the fit it is necessary to multiply the standard deviation by the square root of the total number of experimental data and divide it between the square root of the total number of experimental data minus the number of free parameters.

3.2.2 A Bent Molecule Example: Methinophosphide A Excited Electronic State Bending

The files necessary to run this example are located under the directory `HCP_A`.

```
triat_U3_min_2.0$ cd examples/HCP_A/
triat_U3_min_2.0/examples/HCP_A$ ls
exp_HCP_A_bending.dat  minuit_genpar_HCP_A
input_U3_minuit_HCP_A  output_HCP_A_orig
```

The program is executed in the same way than in the linear case

```
triat_U3_min_2.0/examples/HCP_A$ ../../bin/minuit_U3_genH -v
minuit_genpar_HCP_A \
> input_U3_minuit_HCP_A output_HCP_A

Minuit CHISQRE minimization. U(3) General Hamiltonian.
Making temporary directory ... Done.
Experimental energy file name is: exp_HCP_A_bending.dat
Copying files to temporary directory ... Done.
Running optimization program ... Done.
Copying output file from temporary directory ... Done.
Removing temporary directory ... Done.
```

Last part of the output is saved on file `output_HCP_A_orig`.

Chapter 4

References

1 J. Chem. Phys. *104* (1996) 6956.

2 Chem. Phys. Lett. *365* (2002) 57-68.

3 Chem. Phys. Lett. *375* (2003) 309-320.

4 Phys. Rev. *A77* (2008) 032115.