`triat_u3` User's Guide

Curro Perez Bernal <francisco.perez@dfaie.uhu.es>

Version 2.2.1

Contents

Chapter 1

Introduction

1.1 What's `triat_u3`?

The `Fortran` and `Perl` programs in the `triat_u3` suite perform calculations using the two-dimensional limit of the Vibron Model. This approach is used to model the bending dynamics of linear, quasilinear and bent molecules, using a model based on a bosonic U(3) Lie algebra. This suite includes several programs, in particular it includes a program to fit a fourteen parameter Hamiltonian to experimental bending energy data. This Hamiltonian is the most general one-, two-, three-, and four-body Hamiltonian operator in the model. The package, includes the optimization program `tri_min_T`, as well as other utilities to work with the two-dimensional limit of the vibron model.

1.2 Copyright statement

Chapter 2

Installation guide

The program files can be found as a `tgz` file (`triat_U3_2.X.tgz`, where *X* stands for the version number). The first step is to unpack the file:

```
$ tar xzf triat_U3_2.2.1.tgz
$ ls
triat_U3_2.2.1  triat_U3_2.2.1.tgz
$ cd triat_U3_2.2.1/
triat_U3_2.2.1$ ls
bin/  doc/  examples/  NEWS  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`: Different directories with examples of application of the programs.

- `XCNO`: Program `tri_min_T`, application to fitting the large amplitude bending mode of fulminic acid.
- `HCP_A`: Program `tri_min_T`, application to fitting the bending mode of the A excited electronic state of methinophosphide.
- `Eigenvalues`: Program `en_U3_gen_T`, input files and examples of eigenvalues calculation.
- `Eigenstates`: Program `avec_U3_gen_T`, input files and examples of eigenvalues and eigenstates calculation.
- `Observables`: Input files and examples of calculations for several observables of interest.

2.1 Program compilation

The present section details the programs compilation procedure. There is also a version of the program compiled statically in the `bin` directory (files ending with the suffix `stat`). If you run into trouble during any program compilation you can move forward to 'Getting Started' on page 7 using the provided static version of the programs.

The program compilation in `Debian` systems should be quite smooth if the libraries `BLAS` and `LAPACK` are installed.

These libraries are `FORTRAN` libraries for numerical linear algebra and can be found in most Linux Distributions. In the `Debian (Lenny or Squeeze)` distribution they are associated with packages `libblas-dev` and `liblapack-dev` or other (`atlas` library). IMHO, the best option is to install the `atlas` package. More information about `BLAS` and `LAPACK` can be found in the Netlib (`http://www.netlib.org`) website. <– Compiled with minuit subroutine

The third library (only needed for the minimization program) is part of the CERN scientific routines, providing the `Minuit` minimization tool (See 'References' on page 29). 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_2.2.1$ cd src
triat_U3_2.2.1/src$ ls
assign_gen.f            enrgy_modelH_U3_tri.o  minuit-cern.f
avalavec_gen_U3.f       fcn_sub_gen.f          minuit_u3.f
avalavec_modelH_U3.f    hbldu3_gen.f           readenerg_gfort.f
change_basis.f          hbldu3_modelH.f        scaleh_gen.f
check_phase.f           hbldu3_modelH_tri.f    selectvl.f
chisqre_u3_min_gen.f    hbldu3_modelH_tri.o    so3casimirW2.f
cpexp.f                 intrac.f90             so3casimirW2WB2.f
disdata.f               ir_intensity.f90       so3casimirW4.f
enrgy_gen_U3.f          Makefile
enrgy_modelH_U3_tri.f   maxc.f
```

As can be seen in the directory `Makefile`, if the compilation is successful, the executable files are located under the `triat_U3_2.2.1/bin` directory.

List of executable files:

1 `tri_min_T`: Optimization code. Look for the best fit to a set of experimental data.

2 `en_U3_gen_T`: Compute general Hamiltonian eigenvalues given a set of parameters.

3 `avec_U3_gen_T`: Compute general Hamiltonian eigenvalues and eigenstates given a set of parameters.

4 `en_U3_mh`: Compute simplified model Hamiltonian eigenvalues given a set of parameters.

5 `avec_U3_mh`: Compute simplified model Hamiltonian eigenvalues and eigenstates given a set of parameters.

6 `minuit_U3_genH`: Perl script to launch the minimization program. Help on this script can be obtained executing `minuit_U3_genH -h`.

7 `expected_val_n`: Perl script to compute the expectation value of the number of tau boson. Help on this script can be obtained executing `expected_val_n -h`.

8 `intensity_IR`: Perl script to compute the squared expectation value of the infrared transition operator. Help on this script can be obtained executing `intensity_IR -h`.

There are also sections in the `Makefile` (named `triat_min_Tstat`, `energenh_Tstat`, and `avecgenh_Tstat`, etc.), not included in the `all` label, used to compile a static version of the programs. In case you want compile the program statically, type, for example, `make triat_min_Tstat`. Please, notice that in this case it is convenient to make a backup of the provided statically linked programs (`tri_min_Tstat` and `en_U3_gen_Tstat`) because you will delete them. If you need to compile all the static versions available then run `make stats`.

If the programs works correctly in your system you can also, running as *root* user, install them, by default under the directory `/usr/local/bin`. The associated Perl scripts will also be installed in this directory. The programs are installed in this directory with a suffix indicating the version installed and symbolic links to the original name are also built. It is very convenient to add to the default `$PATH` variable the `/usr/local/bin` in case it is not yet in it. This greatly facilitates the minimization program's execution instructions explained in 'Program Execution' on page 10 and 'Examples' on page 10, because in this case the explicit paths provided are not necessary. In order to install it you should do it as superuser or using `sudo`

```
triat_U3_2.2.1/src$ sudo make install
[sudo] password for sudouser:
triat_U3_2.2.1/src$ ls  /usr/local/bin/
...
avec_U3_gen_T
avec_U3_gen_T_2.2.1
avec_U3_gen_Tstat
avec_U3_gen_Tstat_2.2.1
avec_U3_mh
avec_U3_mh_2.2.1
...
```

If you want to remove these files you can also do it executing `sudo make uninstall`.

Chapter 3

Getting Started

3.1 Minimization code `tri_min_T`

Given a set of experimental bending energies and assignments, the program `tri_min_T` and the `Perl` launcher `minuit_U3_genH` optimize the Hamiltonian parameters to reproduce the experimental information. The optimization is carried out using the `minuit-CERN` code (See 'References' on page 29).

Under the directory `triat_U3_2.2.1/examples` two examples of use of this program can be found, one for a linear molecule (fulminic acid fifth normal vibrational degree of freedom) and one for a bent molecule (magnesium hydroxide). We explain in detail the first case and chiefly highlight the differences that arise in the second case.

3.1.1 Input Files and Program Execution

We proceed now to explain the different input files and the way the provided programs are 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 information on the Hamiltonian parameters needed by minuit to accomplish the minimization.

Experimental Energy File

The experimental energies have to be included in a 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.

If the experimental error is fixed to zero, then the level in question is included in the display and compared to the calculated values but the result of this comparison *is not included* in the calculation of the chi-square and standard deviation, thus making no effect in the fitting procedure.

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 2.1 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.

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) and in the `Minuit` homepage (See 'References' on page 29).

```
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 nÂ·l^2 + P33 (nÂ·W^2 + W^2Â·n) +
    P41 n^4 + P42 n^2Â·l^2 + P43 l^4 + P44 l^2Â·W^2 +
    P45 (n^2Â·W^2 + W^2Â·n^2) + P46 W^4 +
    P47 (W^2Â·Wbar^2 + Wbar^2Â·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.

Program Execution

The `Perl` script `minuit_U3_genH`, under the directory `triat_U3_2.2.1/bin`, is provided as a convenient interface to launch the `Fortran` optimization program. It creates a temporary directory where the program runs and, in this way, several instances of the program can be executed simultaneously.

In the following

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

```
triat_U3_2.2.1/examples/XCNO$ ../../bin/minuit_U3_genH

Usage: minuit_U3_genH [-vtS] minpar_file input_file output_file
       minuit_U3_genH [-vt] [-e prgfile] minpar_file input_file output_file
 Options:
          t :  do not remove temporary directory
          e :  minimization program
          S :  execute the statically linked program
          v :  verbose output
```

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. If the option `-e` is absent, the script tries to locate the program and execute it.

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 the preceding page). The second argument, *input_file*, is the general input (see 'General Input File' on page 8). The last argument is *output_file*, the name of the file with the program output. See 'Examples' on this page to check the application of the program to two cases, one corresponding to a linear molecule and the other to a bent molecule.

3.1.2 Examples

Linear Molecule Example: Fulminic Acid Large Amplitude Bending Mode

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

```
triat_U3_2.1$ cd examples/XCNO/
triat_U3_2.1/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 7. The general input file (see 'General Input File' on page 8) 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_2.2.1/examples/XCNO$ ../../bin/minuit_U3_genH -v \
> -e ~/triat_U3_2.2.1/bin/tri_min_T \
> minuit_genpar_FulminicD input_U3_minuit_FulminicD output_FulminicD

Minuit CHISQRE minimization. U(3) General Hamiltonian.
Executable program: ~/triat_U3_2.2.1/bin/tri_min_T
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_2.2.1/examples/XCNO$
triat_U3_2.2.1/examples/XCNO$ ../../bin/minuit_U3_genH -vS
minuit_genpar_FulminicD \
> input_U3_minuit_FulminicH output_FulminicH

Minuit CHISQRE minimization. U(3) General Hamiltonian.
Executable program: ../../bin/tri_min_Tstat
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.

Bent Molecule Example: Methinophosphide A Excited Electronic State Bending Mode

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

```
triat_U3_2.2.1$ cd examples/HCP_A/
triat_U3_2.2.1/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_2.2.1/examples/HCP_A$ ../../bin/minuit_U3_genH -v minuit_genpar_HCP_
> input_U3_minuit_HCP_A output_HCP_A

Minuit CHISQRE minimization. U(3) General Hamiltonian.
Executable program: ../../../bin/tri_min_T
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` in order that the use can compare the results.

3.2 Eigenvalue calculation codes

There are three possible codes for the calculation of the Hamiltonian eigenvalues. The first one, `en_U3_gen_T` is valid for the general, fourteen-parameter, Hamiltonian, while the other two, `en_U3_mh` and `en_U3_mh_trid`, are valid for a simplified model Hamiltonian, with only two parameters: the pairing and the number operator.

3.2.1 Eigenvalue calculation code `en_U3_gen_T`

Under the directory `triat_U3_2.2.1/examples/Eigenvalues` one example of use of this program can be found.

Input Files and Program Execution

The input file for this program is rather simple. An example input file with all parameters equal to zero called `input_file` is provided. A possible input file is

```
10 # N      INPUT FILE
0   # l
2   # iopts
0.0d0   # P11
0.0d0 # P21
0.0d0   # P22
-10.0d0   # P23
0.0d0   # P31
0.0d0   # P32
0.0d0   # P33
0.0d0   # P41
0.0d0   # P42
0.0d0   # P43
0.0d0   # P44
0.0d0   # P45
0.0d0   # P46
0.0d0   # P47
```

The parameter `N` is the value of the totally symmetric U(3) representation used to model the system. As version 2.1 the maximum possible `N` value is 6000. The vibrational angular momentum is given by the `l` value.

The parameter `iopts` controls the output of the eigenvalues. Possible values are

- `iopts` = 0 the program displays only the ground state energy.
- `iopts` = 1 the program displays all energies.
- `iopts` = 2 the program displays all excitation energies (with g.s. (l=0) energy = 0)

For debugging purposes, in the source file `enrgy_gen_U3.f` there is a variable fixed at zero, called `IPRINT`. Nonzero positive values increase the level of verbosity of the program.

This `Fortran` program is directly launched, providing the described input file as the standard input. If we execute the program with the input file `input_file_doc` given as an example the result is

```
triat_U3_2.2.1/examples/Eigenvalues$ ../../bin/en_U3_gen_T < input_file_doc
   0.0000000000000000
   380.00000000000011
   680.00000000000023
   900.00000000000023
   1040.0000000000002
   1100.0000000000002
```

Examples

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

```
triat_U3_2.2.1/examples/Eigenvalues$ ls
input_file  input_file_doc  input_file_HCP
```

The obtention of the eigenvalues that correspond to the parameters obtained in the fit to the HCP A state are, for zero vibrational angular momentum

```
triat_U3_2.2.1/examples/Eigenvalues$ ../../bin/en_U3_gen_T < input_file_HCP
   0.0000000000000000
   566.85562373229914
   1028.5555120308779
   1484.2632344381639
   2046.9932672140753
   2689.7360152596357
   3382.0393187789541
   4103.8525588498451
   4838.4084093582642
   5569.7433116564898
   6281.1719539978449
   6953.5491948507433
   7562.3703503418556
   8078.6701738697175
   8529.0156142688957
```

3.2.2 Eigenvalue calculation code `en_U3_mh`

Under the directory `triat_U3_2.2.1/examples/Eigenvalues` one example of use of this program can be found. The `LAPACK` subroutine used for diagonalization is `DSYEV`.

Input Files and Program Execution

The input file for this program is rather simple. An example input file with all parameters equal to zero is provided (file `input_file_mh`). A possible input file is

```
10 # N      INPUT FILE
0   # l
2   # iopts
1.0d0   # scale
0.5d0 # xi
```

The parameter `N` is the value of the totally symmetric U(3) representation used to model the system. As version 2.2.1 the maximum possible `N` value is 6000. The vibrational angular momentum is given by the `l` value.

The parameter `iopts` controls the output of the eigenvalues. Possible values are

- `iopts` = 0 the program displays only the ground state energy.
- `iopts` = 1 the program displays all energies.
- `iopts` = 2 the program displays all excitation energies (with g.s. (l=0) energy = 0)

For debugging purposes, in the source file `enrgy_modelH_U3_tri.f` there is a variable fixed at zero, called `IPRINT`. Nonzero positive values increase the program verbosity.

Examples

The recommended way of computing eigenvalues is making use of the `Perl` script `alg_U3_energy` described below in 'Model and General Hamiltonian Eigenvalues' on page 24. This `Fortran` program can also be directly launched, providing the described input file as the standard input. If we execute the program with the provided input file `input_file_mh_doc`, located in the directory `examples/Eigenvalues`, the result is

```
triat_U3_2.2.1/examples/Eigenvalues$ ../../bin/en_U3_mh < input_file_mh_doc
   0.0000000000000000
   2.0025471824580259
   3.4348964743602903
   4.7423306229038324
   6.5330201878030874
   8.7380263091129393
```

3.2.3 Eigenvalue calculation code `en_U3_mh_trid`

Under the directory `triat_U3_2.2.1/examples/Eigenvalues` one example of use of this program can be found.

Input Files and Program Execution

As in the previous case, the input file for this program is quite simple. An example input file with all parameters equal to zero is provided (file `input_file_mh`). A possible input file is

```
10 # N      INPUT FILE
0   # l
2   # iopts
1.0d0   # scale
0.5d0 # xi
```

Again, the parameter `N` is the value of the totally symmetric U(3) representation used to model the system. The present program uses for the matrix diagonalization a `LAPACK` subroutine that takes advantage of the fact that the matrix is tridiagonal and symmetric (subroutine `DSTEVX`). The computing time is larger compared to the `en_U3_mh` program, but it is more stable numerically and it permits to reach larger N values. It also allows for the calculation of a limited subset of eigenvalues. As version 2.2.1 the maximum possible `N` value in this case is 40000. The vibrational angular momentum is given by the `l` value.

The parameter `iopts` controls the output of the eigenvalues. Possible values are

- `iopts` = 0 the program displays only the ground state energy.
- `iopts` = 1 the program displays all energies.
- `iopts` = 2 the program displays all excitation energies (with g.s. (l=0) energy = 0)
- `iopts` > 2 the program calculates and displays the first `iopts` excitation energies (with g.s. (l=0) energy = 0)

For debugging purposes, in the source file `enrgy_modelH_U3_tri.f` there is a variable fixed at zero, called `IPRINT`. Nonzero positive values increase the program verbosity.

Examples

This `Fortran` program can be directly launched, providing the described input file as the standard input. It is recommended to use the `Perl` script `alg_U3_energy`, decribed in 'Model and General Hamiltonian Eigenvalues' on page 24, to compute system eigenvalues. If we execute the program with the provided input file `input_file_mh_doc`, located in the directory `examples/Eigenvalues`, the result is

```
triat_U3_2.2.1/examples/Eigenvalues$ ../../bin/en_U3_mh_trid < input_file_mh_
   0.0000000000000000
   2.0025471824580254
   3.4348964743602899
   4.7423306229038316
   6.5330201878030865
   8.7380263091129429
```

3.3 Eigenvalues and eigenstates calculation code `avec_U3_gen_T`

Under the directory `triat_U3_2.2.1/examples/Eigenstates` examples of use of this program can be found.

3.3.1 Input Files and Program Execution

The input file for this program is the same that for the previous program `en_U3_gen_T`. An example input file with all parameters equal to zero is provided

```
10 # N      INPUT FILE
0   # l
2   # iopts
0.0d0   # P11
0.0d0 # P21
0.0d0   # P22
0.0d0   # P23
0.0d0   # P31
0.0d0   # P32
0.0d0   # P33
0.0d0   # P41
0.0d0   # P42
0.0d0   # P43
0.0d0   # P44
0.0d0   # P45
0.0d0   # P46
0.0d0   # P47
```

See parameter explanation in 'Eigenvalue calculation code `en_U3_gen_T`' on page 13.

In the source file `avalavec_gen_U3.f` a variable named `TOL`, whose default value is `TOL = 1.0D-05` that makes equal to zero the square of the components whose absolute value is less than `TOL`.

For debugging purposes, in the source file `avec_gen_U3.f` there is a variable fixed at zero, called `IPRINT`. Nonzero positive values increase the level of verbosity of the program.

This `Fortran` program is directly launched, providing the described input file as the standard input. Each energy is followed by the corresponding eigenstate expressed in the U(2) and SO(3) basis. The first column is the component, and the second the component squared, followed by the basis state quantum labels. If we execute the program with the input file found above the result is

```
triat_U3_2.2.1/examples/Eigenstates$ ../../bin/avec_U3_gen_T \
> < input_file_doc
```

```
 N =    10 , l =   0 , DIM =      6
    Energy =           0.0000
        Component U(2)          |  n ,  l  >        Component SO(3)         |
      0.36932E-01  0.13640E-02 | 10 ,   0 >      0.53532E-17   0.0000      |
     -0.26115      0.68198E-01 |  8 ,   0 >      0.84737E-17   0.0000      |
      0.60309      0.36372     |  6 ,   0 >      0.76209E-16   0.0000      |
     -0.66066      0.43647     |  4 ,   0 >      0.75201E-16   0.0000      |
      0.35314      0.12471     |  2 ,   0 >     -0.65821E-15   0.0000      |
     -0.74448E-01  0.55424E-02 |  0 ,   0 >      -1.0000       1.0000      |

    Energy =           380.00
        Component U(2)          |  n ,  l  >        Component SO(3)         |
      0.11964      0.14313E-01 | 10 ,   0 >      0.30352E-15   0.0000      |
     -0.52450      0.27510     |  8 ,   0 >     -0.10342E-15   0.0000      |
      0.46888      0.21985     |  6 ,   0 >     -0.13730E-15   0.0000      |
      0.29962      0.89772E-01 |  4 ,   0 >     -0.25502E-15   0.0000      |
     -0.59486      0.35385     |  2 ,   0 >      -1.0000       1.0000      |
      0.21705      0.47111E-01 |  0 ,   0 >      0.32605E-15   0.0000      |

          ..........................................................

    Energy =           1100.0
        Component U(2)          |  n ,  l  >        Component SO(3)         |
      0.60779      0.36941     | 10 ,   0 >      -1.0000       1.0000      |
      0.42977      0.18470     |  8 ,   0 >     -0.12420E-15   0.0000      |
      0.37219      0.13853     |  6 ,   0 >      0.15755E-15   0.0000      |
      0.33976      0.11544     |  4 ,   0 >      0.35379E-16   0.0000      |
      0.31782      0.10101     |  2 ,   0 >      0.32559E-15   0.0000      |
      0.30151      0.90909E-01 |  0 ,   0 >      0.55052E-16   0.0000      |

triat_U3_2.2.1/examples/Eigenstates$
```

3.3.2 Examples

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

```
triat_U3_2.2.1/examples/Eigenstates$ ls
avec_HCP_A.dat input_file_doc  input_file_HCP
```

The obtention of the eigenvalues and eigenstates that correspond to the parameters obtained in the fit to the HCP A state are, for zero vibrational angular momentum, included in the file `avec_HCP_A.dat`. You can reproduce this calculation as follows

```
triat_U3_2.2.1/examples/Eigenstates$ ../../bin/avec_U3_gen_T \
> < input_file_HCP > avec_HCP_A_new.dat
```

3.3.3 Eigenvector calculation code `avec_U3_mh`

Under the directory `triat_U3_2.2.1/examples/Eigenvectors` one example of use of this program can be found. The `LAPACK` subroutine used for diagonalization is `DSYEV`. This program can display and, if requested, save in a file name

Input Files and Program Execution

The input file for this program is rather simple. An example input file with all parameters equal to zero is provided (file `input_file_mh`). A possible input file is

```
10 # N      INPUT FILE
0   # l
2   # iopts
1.0d0   # scale
0.5d0 # xi
.FALSE.        # lsave
```

The parameters `N`, `l`, and `iopts` play the same role than in the eigenvalue calculation program `avec_U3_mh` (see 'Input Files and Program Execution' on page 15). The logical parameter *lsave* controls the saving of the program output. If `.TRUE.` the output is saved in file `u3_model_Ham_results.dat`.

Examples

The file `input_file_mh_doc`, located in the directory `examples/Eigenstates`, can be used to test the program the result is

```
triat_U3_2.2.1/examples/Eigenstates$ ../../bin/avec_U3_mh < input_file_mh_doc
           6
   2.0544928335232679
 -9.89109288407004003E-003          10           0
  0.10702247936336659                8           0
 -0.38507229495936940                6           0
  0.67160435417509567                4           0
 -0.58731580369455527                2           0
  0.21017949717056550                0           0
    .
    .
```

```
   .
   10.792519142636207
 -0.95238689984753599              10          0
 -0.28724503833496851               8          0
 -9.71809090966967437E-002          6          0
 -3.06134933799240318E-002          4          0
 -8.12286486006358333E-003          2          0
 -1.47815676570819368E-003          0          0
```

3.4 Calculation of observables of interest

We now proceed to describe a set of `Perl` scripts provided to facilitate the calculation of different observables of interest. The sample and input files provided for these scripts are located in the directory `triat_U3_2.2.1/examples/Observables`.

3.4.1 Expectation value of the number of tau bosons operator

The `Perl` script `expected_val_n` computes the expectation value of the tau boson number operator with different options. A basic help can be obtained running the script with the `-h` option.

```
triat_U3_2.2.1/examples/Observables$ ../../bin/expected_val_n -h

 Perl script to calculate the normalized expected value of
the U(2) Casimir n for a given eigenstate u (g.s. -> u = 1) and a xi
value interval, for all the eigenstates given a xi value, or for the
eigenstates of a general Hamiltonian.

 The expected value is normalized by the number of bosons N (<n>/N).

 by Currix TM

Usage: expected_val_n [-vhS] [-e progpath] -N Nval lval scale xi
       expected_val_n [-vhS] [-e progpath] -E Nval lval scale xi
       expected_val_n [-vhS] [-e progpath] Nval lval scale u ximin ximax xist
       expected_val_n [-vhS] [-e progpath] [-n] -g inputfile

  Options:
    N : expected value of n as a function of v (0, 1, 2, ... ) for all states
    E : expected value of n as a function of energy
        normalized by N for all states
    S : execute statically linked Fortran program
    v : verbose output
```

```
h : help message
g : general Hamiltonian (supply name of the parameter file)
n : normalize energies by N in the general Hamiltonian case or
    or v by N if -N
e : provide the executable triat files
```

The script allows the user to compute the expected value of the number operator for all the eigenstates for a given control parameter xi value [1] For example, to compute the expected value of the number operator in the eigenstates of the zero angular momentum, N = 10, and xi = 0.4 Hamiltonian

```
triat_U3_2.2.1/examples/Observables$ ../../bin/expected_val_n -E \
10 0 1.0 0.4
0.227411444586968 0.282433724484853
0.375148937971884 0.209292031824226
0.489249830006017 0.275091429272299
0.63853518398931 0.506402309333926
0.831820286396651 0.740247061857603
1.06005653927139 0.986533443227093
```

If the dependence with the number of quanta, instead of the energy, is required, the `-E` option is replaced by `-N`.

The expected value for a particular eigenstate and evaluated in an interval of control parameter values can also be computed[2]. For example, to compute the expected value of n for the ground state of the zero angular momentum, N = 10, and 0.2 < xi < 0.4 Hamiltonian with a step of 0.05 in the control parameter

```
triat_U3_2.2.1/examples/Observables$ ../../bin/expected_val_n \
 10 0 1.0 1 0.2 0.4 0.05
0.2 0.0395832133992901
0.25 0.094015233850396
0.3 0.168639783188264
0.35 0.233906937667085
0.4 0.282433724484853
```

Finally, the last option is to compute the expectation value for the eigenstates of a given general Hamiltonian. To do so, a file with the parameter values of the Hamiltonian has to be provided. In the examples directory you can find the file `input_file_HCP`, with the Hamiltonian parameters of the HCP molecule. To compute the expectation value of n for these eigenstates

```
triat_U3_2.2.1/examples/Observables$ ../../bin/expected_val_n \
```

[1] This makes reference to the model Hamiltonian H = scale[(1-xi) n + xi/(N-1) P].
[2] This makes reference to the model Hamiltonian H = scale[(1-xi) n + xi/(N-1) P].

```
 -ng input_file_HCP
0 0.252375640144704
19.5468965517241 0.214629419159648
35.4689655172414 0.175676105843901
51.1827586206897 0.223634301458906
70.5862068965517 0.293699978181552
92.748275862069 0.354259363279492
116.620689655172 0.411790067302586
141.513793103448 0.468494422286793
166.841379310345 0.525142319743586
192.058620689655 0.582222581646207
216.593103448276 0.640168981577777
239.775862068966 0.699641599554915
260.772413793103 0.761652148046633
278.575862068966 0.817777007616914
294.103448275862 0.820195218254436
```

These results can be checked against the contents of the file `expected_n_HCP.out`.

3.4.2 Expectation value of the IR transition operator

The `Perl` script `intensity_IR` computes the squared expectation value of the infrared transition operator with different options[3]. A basic help can be obtained running the script with the `-h` option.

```
triat_U3_2.2.1/examples/Observables$ ../../bin/intensity_IR -h

Perl script to calculate the infrared intensity for a dipole transition
between eigenstates {l_1}_{u_1} and {l_2}_{u_2} of the U(3) model Hamiltonian
in a given xi or N value interval or for a given xi value.

Note that {l}_{u} stands for u-th eigenvector with l angular momentum (g.s. u

The computed matrix element is

|<{l_2}_{u_2}|T_+|{l_1}_{u_1}>|^2 + |<{l_2}_{u_2}|T_-|{l_1}_{u_1}>|^2

with the selection rule |l_1-l_2| = 1.


     by Currix TM

Usage: intensity_IR [-Sv] Nval l1 u1 l2 u2 scale xi
```

[3]The transition operator is defined as T = D+/sqrt(N) + D-/sqrt(N).

```
        intensity_IR [-Sv] -s Nval l1 u1 l2 u2 scale ximin ximax xistep

        intensity_IR [-Sv] -n Nmin Nmax Nstep l1 u1 l2 u2 scale xi

        intensity_IR [-h]

    Options:
      S : execute statically linked Fortran program
      v : verbose output
      h : help message
      s : consider a series of control parameter values
      n : consider a series of  N values
```

The script allows the user to compute the squared expectation value of the transition operator for a given pair of eigenstates of the model Hamiltonian for a given control parameter xi value [4] For example, to compute the transition intensity between the ground state (zero angular momentum) and the first eigenstate with angular momentum equal to one, in the N = 10, and xi = 0.4 Hamiltonian we will execute

```
triat_U3_2.2.1/examples/Observables$  ../../bin/intensity_IR \
> 10 0 1 1 1 1.0 0.4

 0.4       9.6007861982152359788
```

The expectation value for a particular pair of eigenstates and evaluated in an interval of control parameter values can also be computed using the option `-s`. For example, to compute the transition intensity between the ground state (zero angular momentum) and the second eigenstate with angular momentum l=1, with N = 10, and 0.2 < xi < 0.4 Hamiltonian with a step of 0.05 in the control parameter

```
triat_U3_2.2.1/examples/Observables$ ../../bin/intensity_IR \
> -s 10 0 1 1 2 1.0 0.2 0.5 0.05

 0.2      1.0319595159027931 7806E-0002
 0.25      4.8898564513207522 3243E-0002
 0.3      0.10416957597335169859
 0.35      0.11597972613072936657
 0.4      9.4674596168859432 1492E-0002
 0.45      6.8927045903571804 2020E-0002
 0.5      4.8175816528624174 3362E-0002
```

Finally, the last option is to compute the expectation value for two eigenstates varying the system's size, for different N values, using the option `-n`. For example, to compute the transition

[4] This makes reference to the model Hamiltonian H = scale[(1-xi) n + xi/(N-1) P].

intensity between the ground state (zero angular momentum) and the second eigenstate with angular momentum l=1 in the critical value of the control parameter (xi = 0.2) with N taking values between 10 and 110 with a step of 20 bosons

```
triat_U3_2.2.1/examples/Observables$ ../../bin/intensity_IR \
> -n 10 110 20 0 1 1 2 1.0 0.2
 10      1.0319595159027931 7806E-0002
 30      1.7664923047851473 3430E-0002
 50      2.1148556224653089 7512E-0002
 70      2.3535350685174235 9296E-0002
 90      2.5390079234936954 9193E-0002
 110      2.6926637346404148 0800E-0002
```

3.4.3 Model and General Hamiltonian Eigenvalues

The `Perl` script `alg_U3_energy` computes the Hamiltonian eigenvalues for the model and the general algebraic Hamiltonian. [5]. A basic help can be obtained running the script with the `-h` option.

```
triat_U3_2.2.1/examples/Observables$ ../../bin/alg_U3_energy -h

Perl script to calculate the either the ground state energy or
the full spectrum of a U(3) model or general Hamiltonian.

In the model Hamiltonian case, the calculation can be limited to only a subse
of the eigenvalues (first nstates), and a program taking advantage of the
tridiagonal character of the matrix is provided (and default for N > 5000).

The output can be adapted to the drawing of Birge-Sponer plots.

     by Currix TM

Usage: alg_U3_energy [-STvbB] Nval lval scale xi
       alg_U3_energy [-STvbB] [-I nstates] Nval lval scale xi
       alg_U3_energy [-STvG] Nval lval scale xi
       alg_U3_energy [-SvbB] [-n] -g inputfile
       alg_U3_energy -h


  Options:
    b : Birge Sponer plot output.
    B : Birge Sponer plot output with energy dependence.
    v : verbose output.
```

[5] This makes reference to the model Hamiltonian H = scale[(1-xi) n + xi/(N-1) P].

```
I : compute the energies for the first nstates (nstates > 2)
T : use the program version with tridiagonal matrix diagonalization
G : compute only the ground state energy
g : general Hamiltonian (supply name of the parameter file).
S : execute the statically linked program.
n : normalize energies or quantum number by N in the general Hamiltonian
h : help message.
```

For example, to compute the energy spectrum for the first three excited states with angular momentum `l = 2`, `N = 1000`, and scale and control parameter equal to 1.0 and 0.555, respectively.

```
triat_U3_2.2.1/examples/Observables$ ../../bin/alg_U3_energy -I 4 1000 2 1.0
 0       200.08946405931044
 1       202.26535936605771
 2       204.43652907169133
 3       206.60297075985744
```

To compute the ground state energy for `N = 10000`, `l = 0`, and scale and control parameter equal to 1.0 and 0.75, respectively.

```
triat_U3_2.2.1/examples/Observables$ ../../bin/alg_U3_energy -G 10000 0 1.0 0
 0       1197.8541544435745
```

Finally, to compute the full energy spectrum for `l = 2`, `N = 1000`, and scale and control parameter equal to 1.0 and 0.555

```
triat_U3_2.2.1/examples/Observables$ ../../bin/alg_U3_energy -I 1 1000 2 1.0
 0       200.08946405931044
 1       202.26535936605774
 2       204.43652907169133
 3       206.60297075985744
 4       208.76468198691873
 5       210.92166028156083
 ....


 495       992.09574571198141
 496       994.26349601727406
 497       996.43459207588285
 498       998.60902996360642
 499       1000.7868057766277
```

When the general Hamiltonian spectrum is computed it is necessary to provide the program with an input file, as the provided example input file `input_file_HCP`

```
triat_U3_2.2.1/examples/Observables$ ../../bin/alg_U3_energy -g input_file_HC
 0       0.0000000000000000
 1       566.85562373229914
 2       1028.5555120308779
 3       1484.2632344381639
 4       2046.9932672140753
 5       2689.7360152596357
 6       3382.0393187789541
 7       4103.8525588498442
 8       4838.4084093582642
 9       5569.7433116564898
 10       6281.1719539978449
 11       6953.5491948507442
 12       7562.3703503418556
 13       8078.6701738697193
 14       8529.0156142688975
```

3.4.4 Excitation Energy Diagrams

The `Perl` script `exc_energ_diag_U3_mh` computes the excitation energies necessary to plot correlation energy diagrams for given control parameter values with different options[6]. A basic help can be obtained running the script with the `-h` option.

```
triat_U3_2.2.1/examples/Observables$ ../../bin/exc_energ_diag_U3_mh -h
 Perl script to calculate the u first eigenvalues (g.s. -> u = 1) of
the algebraic U(3) model Hamiltonian in a given xi value interval.
If  u = 0 all anergies are included.

 The energy values are normalized by the number of bosons N (E/N).

 by Currix TM

Usage: exc_energ_diag_U3_mh [-vhS] [-e progpath] Nval lval scale u ximin xima


  Options:
    S : execute statically linked Fortran program
    v : verbose output
    h : help message
    e : provide the executable triat files
```

For example, to compute the excitation energy for the first three excited states with zero angular momentum, with N = 10, and in the interval of xi starting at 0.1 and ending at 0.4 with a step of 0.05 we execute

[6] This makes reference to the model Hamiltonian H = scale[(1-xi) n + xi/(N-1) P].

```
triat_U3_2.2.1/examples/Observables$ ../../bin/exc_energ_diag_U3_mh \
10 0 1.0 4 0.1 0.4 0.05

0.1 0.0969591084744126 0.24303900731891 0.410527294642692 0.595860630601058
0.15 0.141605046653722 0.261843236909951 0.416273614122525 0.596349388138778
0.2 0.180891112017846 0.279990992516115 0.423496605540945 0.599080754985964
0.25 0.210586829109494 0.300331396846074 0.433272340537003 0.604387056209837
0.3 0.226957621450917 0.325171141950419 0.446759291603703 0.612551514295007
0.35 0.2312738839248 0.351753480722973 0.465174292657181 0.623837735817851
0.4 0.227411444586968 0.375148937971884 0.489249830006017 0.63853518398931
```

Chapter 4

References

1 Minuit minimization package (http://www.cern.ch/minuit): Comput. Phys. Commun. *10* (1975) 343-367.

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

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

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

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