QuAC Manual

QuAC - Quantum Atomic Collision

Yuri Alexandre Aoto

Table of Contents

1 Overview

Overview of QuAC.

A little bit of theory.

What QuAC does and

2 Compiling and running

Compiling and running explanation.

Command options.

3 Input

The input of QuAC is divided in two parts. The first contains the jobs informations and the second the potentials definitions. The first line of the input file is an job name. After, several jobs can be passed, which have the general syntax:

```
job_type [job_name]
job_keywords
...
```

job_type is the type of the job (see Chapter 5 [Jobs], page 9) and the optional argument, *job_name*, is a name for this job. it is not completely optional, since if an posterior analysis of this job will be carried it is required. The number of jobs handled by QuAC is arbitrary and the keywords that follow this first line are concerned only to this job, with the exception of particles (see Section 7.1 [Masses and Atoms], page 11) and formats (see Section 7.2 [Format], page 11) definitions. Theses special keywords apply to the subsequent job, until another keyword override it.

There are two kinds of jobs in QuAC: jobs that does some calculation and jobs that analyses this calculations. QuAC calculates just what it is asked for and do not try to be smart and analyse the results if it is not explicitly required.

The potentials definitions must be placed after all the jobs. A keyword (see Section 7.3.7 [Potential Name], page 15) links the job and the potential that it will use.

Some keywords in the job and potential definitions can be given in any unit recognised by QuAC. Run `QuAC --units` to see the available units and conversion factors. These informations is stored in an ASCII file (the command `QuAC --units` provides it) and it is possible add new units or change the conversion factor changing it.

4 Potentials

QuAC handles different types of potentials/dipole moments functions and they must be given after all of the jobs. Multiples potentials can be given and the general syntax is:

```
POTENTIAL name
[Print Lin/Log r_min r_max delta_r dist_uni ener_uni]
type lambda dist_uni [ener_uni]
...
‘parameters’
...
```

or

```
DIP_MOM name
[Print Lin/Log r_min r_max delta_r dist_uni ener_uni]
type lambda dist_uni [ener_uni]
...
‘parameters’
...
```

The argument *name* is a name for the given potential, the optional keywords and variables in the second line ask for print the potential in the ‘`.log`’ file, from *r_min* to *r_max* increasing by *delta_r*. *Lin/Log* should be ‘`Lin`’ or ‘`Log`’ and means linear or logarithm r scale and *dist_uni* and *ener_uni* are the units the potential should be printed. *type* is the type of potential and *lambda* is the quantum number for the z-component of electronic angular momentum. *dist_uni* and *ener_uni* are the units the potential parameters will be given and do not need to be the same of the ones used in ‘`Print`’ line.

Each type of potential have their specific syntax, described below, and must be placed just after these arguments.

4.1 Rigid Sphere

The rigid sphere potential, `HardSphere`, is zero valued after a given distance, the radius, and the wave function must completely vanish at distances below the radius. Only for this type of potential the unit of energy is not required.

Expression:

$u_l(r) = 0.0$ if $r \leq r_0$
$V(r) = 0.0$ if $r > r_0$

General syntax:

```
radius
```

radius $= r_0$

4.2 Soft Sphere

`SoftSphere` define a soft sphere, which is also zero valued after a given distance, the radius, but below it has a repulsive exponential behaviour.

Expression:

$V(r) = \varepsilon \left\{ \left(\frac{r_0}{r}\right)^{\alpha} - 1 \right\}$ if $r \leq r_0$

$V(r) = 0.0 \qquad\qquad \text{if } r > r_0$

General syntax:

```
radius epsilon alpha
```

radius $= r_0$, *epsilon* $= \varepsilon$, and *alpha* $= \alpha$.

4.3 Square

A square potential, keyword `Square`, is given as an step like function, being zero if the distance is larger then a given radius and constant otherwise.

Expression:

$V(r) = \varepsilon \qquad\qquad \text{if } r \leq r_0$
$V(r) = 0.0 \qquad\qquad \text{if } r > r_0$

General syntax:

```
radius epsilon
```

radius $= r_0$, *epsilon* $= \varepsilon$.

4.4 Lennard Jones

Lennard-Jones potentials can be given in multiples ways. The parameters can be given by `LenJon`, which refers to the common (12,6) potential (first expression), and by `LenJon-N-M`, which refers to a (N+M,N) type (second expression). The potential names `LSLenJon` and `LSLenJon-N-M` corresponds to a Lennard-Jones ((12,6) and (N+M,N), respectively) potentials whose parameters is obtained by least square over the given points.

Expression:

$$V(r) = \varepsilon \left\{ \left(\frac{r_0}{r}\right)^{12} - 2\left(\frac{r_0}{r}\right)^{6} \right\}$$

$$V(r) = \epsilon \left\{ \left(\frac{\bar{r}}{r}\right)^{N+M} - \left(\frac{\bar{r}}{r}\right)^{N} \right\}$$

General syntax:

```
radius epsilon
```

or

```
r_1 V_1
r_2 V_2
...
r_n V_n
```

In the first case, *radius* $= r_0$, the equilibrium distance, and *epsilon* $= \varepsilon$, the dissociation energy, for '`LenJon`' and '`LenJon-N-M`' potentials. Note that the parameters given are the equilibrium distance and the dissociation energy in both cases. The expression above for the general '`LenJon-N-M`' Lennard-Jones potential are in terms of the parameters ϵ and $\bar{r}$, related to r_0 and ε by:

$$\bar{r} = r_0 \left(\frac{N}{N+M}\right)^{\frac{1}{M}}$$

$$\epsilon = \varepsilon \left(\frac{N+M}{M}\right) \left(\frac{N+M}{N}\right)^{\frac{N}{M}}$$

For a '`LSLenJon`' the points are given as in the second case.

4.5 Morse

The parameters of a Morse potential can be given explicitly, by the type `Morse`, or as the ones which gives the same equilibrium distance, dissociation energy and force constant of the natural cubic splines over a given set of points, by the name `AppMorse`.

Expression:

$V(r) = \varepsilon \left\{ \left(1 - e^{-\beta(r-r_0)}\right) - 1 \right\}$

General syntax:

```
radius epsilon beta
```

or

```
r_1 V_1
r_2 V_2
...
r_n V_n
```

radius $= r_0$, *epsilon* $= \varepsilon$, *beta* $= \beta$. The second case is for '`AppMorse`' type.

4.6 Murrel Sorbie

A Murrel Sorbie type of potential is defined by the keyword `MurSorbie`.

Expression:

$V(r) = \varepsilon \left\{ 1 + \sum_{i=1}^{n} a_i \left(r - r_0\right)^i \right\} e^{-a_1(r-r_0)}$

General syntax:

```
radius epsilon
a_1
a_2
...
a_n
```

radius $= r_0$, *epsilon* $= \varepsilon$, *a_i* $= a_i$, i=1,2,...,n.

4.7 Linear Spline

`LinearSpl` asks for a potential as a linear spline over the points given. The energy after the last point is set constant and equal to the energy of this last point.

General syntax:

```
r_1 V_1
r_2 V_2
...
r_n V_n
```

4.8 Natural Cubic Splines

Cubic splines are piecewise cubic functions, with continuous first and second derivatives in the range between the first and the last points, which interpolates the given points. QuAC uses the natural cubic splines, whose second derivative in the first and last points are zero. For the `NatCubSpl` type of potentials the energy is set constant and equal to the energy of the last point for distances larger than the last given energy.

The `CubSplr6` and `CubSplr6r8` are calculated as $\frac{C_6}{r^6}$ and as $\frac{C_6}{r^6}+\frac{C_8}{r^8}$, respectively, for distances larger than the last distance given, and the coefficients are calculated in order to join this tail smoothly with the cubic spline. For the second case, where one more condition is required, the spline is calculated without the last point, whose energy is treated as being the asymptotic limit of this long range behaviour.

The coefficients of the long range tail of the potential can be supplied using the potential types `CubSplFitr6`, `CubSplFitr6r8`, and `CubSplFitr6r8r10`. For these potentials the potential after the last distance is calculated as $\frac{C_6}{r^6}$, $\frac{C_6}{r^6}+\frac{C_8}{r^8}$, and $\frac{C_6}{r^6}+\frac{C_8}{r^8}+\frac{C_{10}}{r^{10}}$, respectively.

General syntax:

```
r_1 V_1
r_2 V_2
...
r_n V_n
```

or

```
r_1 V_1
r_2 V_2
...
r_n V_n
r_inf V_inf
```

or

```
coef_r6
r_1 V_1
r_2 V_2
...
r_n V_n
```

or

```
coef_r6 coef_r8
r_1 V_1
r_2 V_2
...
r_n V_n
```

or

```
coef_r6 coef_r8 coef_r10
r_1 V_1
r_2 V_2
...
r_n V_n
```

r_i and *V_i*, *i*=1,2,...,n,inf are distance/energy the points. The first case is for `NatCubSpl` and `CubSplr6`, the second for `CubSplr6r8` and the last three for `CubSplFitr6r`, `CubSplFitr6r8`, and `CubSplFitr6r8r10`, where *coef_r6 coef_r8 coef_r10* are the coefficients C_6, C_8, and C_{10}.

4.9 Direct Potential

Calculations can be carried using a direct potential, where the energy is get directly from the output of an electronic structure calculation.

General syntax:

```
elec_struc_command
get_ener_command
rm_command
keyword format_ener format_dist
r_inf zero_e_ret lower_bound upper_bound eq_dist force_const
elec_struc_input
...
```

Here, *elec_struc_command* is the command to run the electronic structure calculation, *get_ener_command* is a command to print, in the standard output, the energy (that will be read by QuAC using the format *format_ener*), and *rm_command* is a command to delete the files from the electronic structure calculation. *keyword* is a word in the input file of the electronic structure that will be replaced by the distance to be calculated, using the fortran format *format_dist*. In the following line some properties of the potential that are automatically computed by QuAC for other potentials must be given, since the electronic structure programs usually have efficient procedures to calculate them. *r_inf* is the distance at which the particles can be considered isolated (this distance will be used to calculate the dissociation energy), *zero_e_ret* is the zero energy turning point, *lower_bound* and *upper_bound* are distances to be considered lower and upper bound of the potential, that is, the potential is assumed to be infinity below *lower_bound* and almost constant after *upper_bound* (but is also calculated directly after this point). *eq_dist* and *force_const* are the equilibrium distance and the force constant and should be negative if the potential is repulsive.

The rest of the definition until a the end of the potential definition (the end of the input file or a line starting with ‘`POTENTIAL`’ or ‘`DIP_MOM`’) is the input file to be used in the electronic structure calculation. It depends, of course, of the program and every instance of the keyword *keyword* in it will be replaced by the distance.

5 Jobs

QuAC handle an arbitrary number of jobs and they are executed sequentially. Here is a general discussion of what each job does.

5.1 Phase Shifts

The first type of job implemented in QuAC is a phase shift calculation and this is for what QuAC was first designed (QuAC stands for **Qu**antum **A**tomic **C**ollision). In this jobs the radial Schrdinger equation is integrated for positive energy values (relative to the dissociation limit) and the phase shift is obtained by the analysis of the behaviour of the wave function in the long non interacting range of the potential, $\Psi_l(r) = \sin(\delta_l)j_l(kr) + \cos(\delta_l)n_l(kr)$, where δ_l is the phase shift, j_l and n_l are the spherical Bessel functions of first and second kind, and $k^2 = \frac{2mE}{\hbar^2}$.

QuAC stores the phase shifts for posterior analysis.

5.2 Scattering Length

The scattering length is calculated by the asymptotic analysis of the zero energy and zero angular momentum wave function at large distances. It behaves as $\Psi_0(r) = Ar + B$, with $a_{scatt} = -\frac{B}{A}$.

The scattering length can also be computed by the integration of the log-derivative equation, as described by Meshkov (REF). This is done using '`MESHKOV`' as method of scattering length (see Section 7.3.16 [Scattering Length Method], page 17).

5.3 Bound States

The vibrational bound states can be obtained by the second order algorithm of Cooley and Lowdin (REF). They can be calculated in three ways. QuAC can calculate exactly the ones required, by the keywords `VibLevel` or `VibLevelMax` (see Section 7.3.19 [Vibrational Levels], page 18), use the energies given as starting points for the algorithm or calculate all the vibrational levels, if neither the energies nor the vibrational levels were given. QuAC automatically stores the energy, vibrational level, number of iterations and wave function for posterior analysis.

5.4 Results Analysis

The calculations carried by Phase Shifts or Bound States calculations can be analysed in this job. The analysis implemented are the calculation of the total, partial and differential cross sections, for Phase Shifts results, and vibrational transition intensities, Dunhan parameters and vibrationally averaged equilibrium distances for Bound States results. Only one analysis is carried by each Results Analysis job and must be asked by the keyword `AnalysisType` (see Section 7.4.1 [Analysis Type], page 21).

6 Output

QuAC generates two output files. One, with an extension '`.out`', print the input, most import information, like parameters of calculations, results and masses. The other, with an extension '`.log`', print secondary informations, like potential information, the wave function (if requested), phase shifts (if requested), etc.

7 Keywords

The behaviour of the calculations carried by QuAC can be modified by several keywords, some optional and some mandatory for some calculations. The keywords for masses (see Section 7.1 [Masses and Atoms], page 11) and format (see Section 7.2 [Format], page 11) specification are the only that are given outside the jobs and are applied to all calculation, until another format or mass specification is reached, which overwrite the previous definitions.

All the other keywords apply for the concerned job. They have one of the general syntaxes:

```
keyword [arguments ...]
[arguments]
[...]
[arguments]
```

Whether some or all of the arguments are required depends on '`keyword`'.

7.1 Masses and Atoms

The keywords `MASSES`, `RED_MASS` and `ATOMS` can be used to define the masses of the particles. The first two are used to give directly the numerical value of the masses, along with the unit, and the last to give the atomic symbols and use the corresponding masses from the atomic data library of QuAC. A list of the atoms and the syntax of atomic symbols available can be viewed running `QuAC --atoms`.

General syntax:

```
MASSES mass_1 mass_2 unit
```

or

```
RED_MASS reduced_mass unit
```

or

```
ATOMS atom_1 atom_2
```

mass_1 and *mass_2* are the masses of the particles, *reduced_mass* is the reduced mass of the two particle system, and *atom_1* and *atom_2* are atomic symbols. *unit* is an unit of mass.

Incompatibilities:

If a differential cross section in the laboratory system of reference will be calculated, the mass of both particles should be given.

Default values:

ANALYSE DEFAULT VALUES. WHEN MASS DEFINITION CAN BE SKIPPED?

7.2 Format

The format for output print can be specified with the keyword `FORMAT`.

General syntax:

```
FORMAT
quantity format
quantity format
...
quantity format
```

quantity can be one of the following: '`energy`', '`energy_cm-1`', '`cross_sec`', '`angle`', '`ang_momentum`', '`distance`', '`wave_function`', or '`potential`'. *format* is any valid fortran format (without the enclosing parentheses) for '`REAL`' or '`INTEGER`' kind of variable, depending, of course, of *quantity*. All the formats can be restored to default values using '`default`' as *quantity*, case in which no *format* need to be passed.

Incompatibilities:

There are no incompatibilities for this keyword.

Default values:

The default formats are: '`ES15.8`' for '`energy`', '`ES15.8`' for '`cross_section`', '`F0.10`' for '`angle`', '`I0`' for '`ang_momentum`', '`F0.10`' for '`distance`', '`ES15.8`' for '`wave_function`', '`ES17.10`' for '`potential`', and '`F0.5`' for '`energy_cm1`'

7.3 Calculation Keywords

Calculation keywords.

7.3.1 Angular Momentum

The specification of the angular momentum for `PHASE_SHIFTS` and for `BOUND_STATES` calculations can be done giving the individuals values or the maximum value and must be given just once.

General syntax:

```
l
l_1
l_2
...
l_n
```

or

```
lmax l_max
```

l_1, *l_2*, *l_n*, *l_max* must be integers and the later case is equivalent to give all values from zero to the maximum values specified.

Incompatibilities:

The value of angular momentum should not be given for a `SCATT_LENGTH` calculation, since it does only zero angular momentum integration.

Default values:

There is no default values for `PHASE_SHIFTS` calculation and it always must be given. For `BOUND_STATES`, only the zero rotational levels are calculated by default.

7.3.2 Energy and Velocity

The energy for the calculation can be given individually, in a range or in an exponential range. It can be given as energy or as velocity and the *unit* must be consistent with this choice.

General syntax:

```
Energy unit
energy_1
energy_2
...
energy_n
```

or

```
InterEnergy e_min e_max delta_e unit
```

or

```
ExpInterEnergy exp_min exp_max delta_exp unit
```

In the first case the values of *energy_1*, *energy_2*, and *energy_n* are the energies required. In the second, the values of energy are between *e_min* (including) and *e_max*, increasing by *delta_e*. In the later case the range is given in the exponent (base 10). Analogously, replacing '`Energy`' by '`Velocity`', the values can be given as the collision velocity.

Incompatibilities:

It is not allowed pass velocity For a `BOUND_STATES` calculation.

Default values:

For `PHASE_SHIFTS` there are no default values and the energy is mandatory. For `SCATT_LENGTH` the integration is carried at zero energy by default. The behaviour of a `BOUND_STATES` calculation with no energy given explicitly depends on the vibrational levels required. (See Section 5.3 [Bound States], page 9.)

7.3.3 Integration Method

The numerical integration can be performed by fourth-order Runge-Kutta, Adams-Bashforth-Moulton predictor-corrector, Numerov, or Johnson log-derivative method. It is specified by the keyword `IntMeth`

General syntax:

```
IntMeth int_method
```

Here `int_method` is the integration method and can be *RK4*, *ABM*, *NUMEROV*, or *JOHNSON*.

Incompatibilities:

The Johnson log-derivative method is implemented only for `SCATT_LENGTH` calculations by the Meshkov procedure, for which is required. (See Section 5.2 [Scattering Length], page 9.)

Default values:

The default integration method is the Numerov, with the exception of the Meshkov procedure for scattering length calculation, which requires the Johnson log-derivative method.

7.3.4 Integration Step

The integration step of the numerical integration can be changed using the `IntStep` and `FinalIntStep` keywords. The first gives the step size of the integration through the interaction region. This step can be iteratively doubled in the long range region of the potential until the value specified by the second keyword. The units of the integration steps specified by these keywords are the same of the distance unit used in the potential definition.

General syntax:

`IntStep` *step*

or

`FinalIntStep` *step*

step is the step size in the unit used in the potential of the current job.

Incompatibilities:

The `IntStep` keyword should not be given for a scattering length by the Meshkov procedure. Use `AddSteps` instead, to give the size of the integration mesh, see Section 7.3.12 [Additional Steps], page 16. `FinalIntStep` can be given only for `PHASE_SHIFTS` and `SCATT_LENGTH` calculations. Must be greater or equal the integration step.

Default values:

The default integration step is $\rho = kr = 0.005$.(See Chapter 1 [Overview], page 1.)

7.3.5 Print

The wave function can be printed in the '`.log`' file using the keyword `Print`. Since most calculations involves several integration, take care because huge files may be generated even for medium size calculations!

General syntax:

`Print` *print_interval*

The wave function will be printed at each *print_interval* steps.

Incompatibilities:

Currently is not possible print the final vibrational wave function if `SaveGlobalPot` (see Section 7.3.15 [Save Global Potential], page 17) is given.

Default values:

By default the wave function is not printed.

7.3.6 Normalisation

The normalisation of the wave function can be controlled using the keyword `WFScale`. It can be used to avoid overflow or underflow for some problematic cases or to print wave functions scaled as your desire.

General syntax:

`WFScale` *scale*

The initial condition of the numerical integration will be multiplied by the *scale* argument.

Incompatibilities:

There are no incompatibilities for this keyword.

Default values:

The default value of *scale* is 1.

7.3.7 Potential Name

The keyword `PotName` should be used to define the potential that will be used in the job. It should be the name of one of the potentials given in the input. (See Chapter 4 [Potentials], page 4.)

General syntax:

```
PotName potential_name
```

potential_name is the name of the potential to be used in this job.

Incompatibilities:

There are no incompatibilities for this keyword.

Default values:

If just one potential was defined in the input, this is the default potential and this keyword is not required. if more than one potential was given, there is no potential default and it should be passed.

7.3.8 Initial Condition

The initial condition of the free particle (the value of the function in the first point is not zero, See Chapter 1 [Overview], page 1,) can be used, by the keyword `FPIniCond`.

General syntax:

```
FPIniCond
```

Incompatibilities:

There are no incompatibilities for this keyword. CHANGE THE BOUND—-> INWARD INTEGRATION WILL FAIL!!

Default values:

The initial condition of the free particle is used by default just for calculations on '`FREE_PARTICLE`' or '`Square`' potentials. In the other cases the derivative is the one of the free particle and the value of the function is set to zero at the initial integration point.

7.3.9 Save Steps

The steps saved in the end of the numerical integration for posterior analysis can be printed in the '`.log`' file using the `SaveSteps` keyword.

General syntax:

```
SaveSteps
```

Incompatibilities:

There are no incompatibilities for this keyword.

Default values:

The points are not printed in the '`.log`' file by default.

7.3.10 Phase Shift Method

The phase shift calculation method can be changed using the keyword `PhaseShiftMeth`. *Nowadays only one phase shift method is implemented!*

General syntax:

`PhaseShiftMeth` *`method`*

The only value for *method* implemented is '`REGRESSION`'.

Incompatibilities:

It is valid just for a `PHASE_SHIFTS` calculation.

Default values:

The default phase shift calculation method is '`REGRESSION`'.

7.3.11 R min and R max

The initial and final points of the numerical integration are changed using the `Rmin` and `Rmax` keywords, respectively. Like the integration step, they are given in the unit used in the potential definition of the current job.

General syntax:

`Rmin` *`r_min`*

or

`Rmax` *`r_max`*

Incompatibilities:

For a `BOUND_STATES` calculation they can be passed only in a fixed mesh calculation. (See Section 5.3 [Bound States], page 9.). The value of *r_min* can not be lower than the lower bound of the corresponding potential. (See Chapter 4 [Potentials], page 4.)

Default values:

The default value of *r_min* and *r_max* are the lower and the upper bound of the potential, respectively. If the lower bound is zero it will depend of the potential.(See Chapter 4 [Potentials], page 4.)

7.3.12 Additional Steps

After the integration up to *r_max* and the eventual increasing of the step size, the integration is carried out some more points to be saved for further analysis. The number of steps of this extra integration is given by the keyword `AddSteps`. For `SCATT_LENGTH` calculation by the Meshkov procedure it consists of the number of steps in the mesh.

General syntax:

`AddSteps` *`steps`*

steps is the number of the extra steps.

Incompatibilities:

Has meaning only for `PHASE_SHIFTS` and `SCATT_LENGTH` calculations.

Default values:

The default value for the additional steps is 1000.

7.3.13 Save Steps at each

The interval at each steps are saved in the additional integration can be set using the `SaveAtEach` keyword.

General syntax:

`SaveAtEach` *`interval`*

interval is the interval at which each step are saved.

Incompatibilities:

Do not have meaning for `SaveAtEach` for `BOUND_STATES` or `SCATT_LENGTH` by Meshkov procedure calculations.

Default values:

Points are saved at every step, i.e., the default value of *interval* is 1.

7.3.14 Print Phase Shifts

To print the phase shifts in the '`.log`' file use the keyword `PrintPhaseShift`

General syntax:

`PrintPhaseShift`

Incompatibilities:

Have meaning only for `PHASE_SHIFTS` calculations.

Default values:

The phase shifts are not printed by default.

7.3.15 Save Global Potential

In order to save computing time, the potential at every point needed in a calculation can be calculated just once and stored. This is useful specially for computing demanding potentials, like '`Direct`' potential.

General syntax:

`SaveGlobalPot`

Incompatibilities:

Not implemented for `SCATT_LENGTH` calculation, since only one calculation is carried. For a `BOUND_STATES` calculation require a fixed mesh points integration. (See Section 7.3.20 [Vibrational Fixed Mesh], page 19.)

Default values:

Do not save the potential by default.

7.3.16 Scattering Length Method

One of the two methods available to calculate the scattering length can be specified by the keyword `ScattLenMeth`.(See Section 5.2 [Scattering Length], page 9.)

General syntax:

`ScattLenMeth` *`method`*

method can be '`DIRECT`' or '`MESHKOV`'.

Incompatibilities:

Have meaning only for `SCATT_LENGTH` calculation.

Default values:

The default method is the '`DIRECT`'.

7.3.17 Meshkov Parameters

The Meshkov parameters, $\bar{r}$ and β (See Section 5.2 [Scattering Length], page 9.), are required for a Meshkov procedure for scattering length calculation. It can be given with the keyword `MeshkovParam`.

General syntax:

```
MeshkovParam r_bar beta
```

The *r_bar* and *beta* arguments are the Meshkov parameters and *r_bar* must be given in the units used in the potential of the current job.

Incompatibilities:

Have meaning only for scattering length calculations by the Meshkov procedure.

Default values:

Do not have default values. Must be given in a Meshkov calculation.

7.3.18 Richardson Extrapolation

A Meshkov calculation with the Richardson extrapolation to zero step size can be achieved using the keyword `RichardsonExtr`.

General syntax:

```
RichardsonExtr
```

Incompatibilities:

Have meaning only for scattering length calculations by the Meshkov procedure.

Default values:

By default the Richardson extrapolation is not done.

7.3.19 Vibrational Levels

The required vibrational levels for a `BOUND_STATES` calculation can be specified in two ways, similar to the specification of the angular momentum. (See Section 7.3.1 [Angular Momentum], page 12.)

General syntax:

```
VibLevel
v_1
v_2
...
v_n
```

or

```
VibLevelMax v_max
```

Here, *v_1*, *v_2*, *v_n*, and *v_max* are integers and correspond to the vibrational levels required. The second case is equivalent to ask for the levels 0, 1, ..., v_max by the first.

Incompatibilities:

Have meaning only for a `BOUND_STATES` calculation.

Default values:

If not passed in a vibrational calculation, QuAC attempt to calculate every vibrational level.

7.3.20 Vibrational Fixed Mesh

In order to run a calculation with the potential calculated just once (See Section 7.3.15 [Save Global Potential], page 17,) the integration mesh must be fixed and this should be asked by the `VibFixedMesh` keyword. *QuAC do not tries to be smart and save a global potential just because you ask for a fixed mesh calculation or vice-versa! It must be done manually!*

General syntax:

```
VibFixedMesh
```

Incompatibilities:

Have meaning only for a `BOUND_STATES` calculation.

Default values:

It do not run a fixed mesh calculations by default.

7.3.21 Print Vibration Iterations

The information of each iteration for a `BOUND_STATES` calculations can be printed if requested by the `PrintVibIterations` keyword.

General syntax:

```
PrintVibIterations
```

Incompatibilities:

Have meaning only for a `BOUND_STATES` calculation.

Default values:

The iterations are not printed by default.

7.3.22 Print WF Vibration Iterations

If the keyword `PrintVibWFIterations` is given, the wave function will be printed in every iteration for a `BOUND_STATES` calculation. The keyword `Print` (See Section 7.3.5 [Print], page 14,) can be used to define the interval that the wave function will be printed.

General syntax:

```
PrintVibWFIterations
```

Incompatibilities:

Have meaning only for a `BOUND_STATES` calculation.

Default values:

The vibrational wave function for each iterations are not printed by default.

7.3.23 Maximum Iteration

The maximum number of iterations in a vibrational calculation can be specified using the keywords `Vib1stOrderMaxIter` and `VibMaxIter`. The first is the maximum number of iterations of the first order halving algorithm, used to approach to the desired eigenvalue, and the second is the maximum total number of iterations.

General syntax:

```
Vib1stOrderMaxIter maxit_1
```

or

```
VibMaxIter maxit_tot
```

The values *maxit_1* and *maxit_tot* are the maximum number of iterations, of the first order and total algorithm, respectively.

Incompatibilities:

Have meaning only for a `BOUND_STATES` calculation.

Default values:

The default value for *maxit_1* is 20 and for *maxit_tot* is 50.

7.3.24 Vibrational Threshold

The threshold of the vibrational calculation is specified with the keyword `VibThreshold`.

General syntax:

```
VibThreshold threshold
```

threshold is the threshold in cm^{-1} , if the energy was not given for this job or in the same used in the energy specification.

Incompatibilities:

Have meaning only for a `BOUND_STATES` calculation.

Default values:

The default value is $1.0 \cdot 10^{-5}$ cm^{-1} .

7.3.25 Vibrational R ini shift

The first point in the outward and inward integration can be specified by keyword `VibRiniShift`. Its arguments correspond to how much away from the classical turning points the integration should start.

General syntax:

```
VibRiniShift shift_int shift_ext
```

The outward integration will begin at '`ICTP`' - *shift_int*, where '`ICTP`' is the internal classical turning point, in the unit used in the potential definition. CHECK!!!! Analogously for *shift_ext*

Incompatibilities:

incompatibilities

Default values:

default

7.4 Analysis Keywords

Analysis keywords.

7.4.1 Analysis Type

The analysis type to be performed by the job should be given by the `AnalysisType` keyword.

General syntax:

```
AnalysisType type
```

Here, *type* is the analysis type and can be one of the followings: '`PartialCrossSec`', '`TotalCrossSec`', '`DifferentialCrossSec`', '`VibrationalIntensity`', '`DunhanExpansion`', and '`EquilibriumDistance`'. (See Chapter 1 [Overview], page 1.)

Incompatibilities:

The analysis type must be consistent with the job to be analysed. Hence '`PartialCrossSec`', '`TotalCrossSec`', and '`DifferentialCrossSec`' can be asked only for `PHASE_SHIFTS` calculations and '`VibrationalIntensity`', '`DunhanExpansion`', and '`EquilibriumDistance`' only for `BOUND_STATES` calculations.

Default values:

There are no default values for this keyword and it must be passed for every `RES_ANALYSIS` job.

7.4.2 Angles

The angles for an differential cross section analysis must be passed by the `Angle` or `InterAngle` keywords.

General syntax:

```
Angle
angle_1
angle_2
...
angle_n
```

or

```
InterAngle a_min a_max delta_a
```

The values *angle_1*, *angle_2*, and *angle_n*, for the first case, are the angles required, in degree. In the second the angles are from *a_min* (including) to *a_max*, increasing by *delta_a*.

Incompatibilities:

Have meaning only for a '`DifferentialCrossSec`' analysis type.

Default values:

There are no default values for angles and it must be given for every differential cross section calculation.

7.4.3 Job Analysed

The job or the jobs to be analysed must be given with the keywords `Job` and `Jobs`.

General syntax:

```
Job job_name
```

or

```
Jobs job_name_1 job_name_2
```

job_name, *job_name_1*, and *job_name_2* are the names of the jobs. The first can be given with spaces and the complete name are properly recognised, but for the second keyword the job names, *job_name_1* or *job_name_2* can not have spaces. To give two jobs when both or one of them have spaces use the keyword *Job* twice.

Incompatibilities:

The job required should be consistent with the analysis type. (See Section 7.4.1 [Analysis Type], page 21.) The only analysis type which analyse two jobs is '`VibrationalIntensity`'. You can analyse only jobs calculated before the analysis.

Default values:

There are no default values for these keywords. The job (or jobs) must be passed for every analysis.

7.4.4 Electric Dipole Function

The electric dipole moment function for an intensity analysis can be given with the keyword `ElecDipoleFunction`.

General syntax:

```
ElecDipoleFunction dip_name
```

dip_name is the name of the dipole moment function to be used in this analysis.

Incompatibilities:

Have meaning only for a `VibrationalIntensity` analysis. It is incompatible with the keyword `FrankCondon`.

Default values:

There are no default value for this keyword. !!!CHANGE TO RECOGNISE IF ONLY ONE WAS GIVEN!!

7.4.5 Frank Condon

The vibrational intensity with the Frank Condon approximation can be carried using the keyword `FrankCondon`.

General syntax:

```
FrankCondon ave_dip ave_dip_unit
```

Here the *ave_dip* and *ave_dip_unit* arguments are the averaged electric transition dipole moment and its unit, respectively.

Incompatibilities:

Have meaning only for '`VibrationalIntensity`' analysis performed with two vibrational jobs. It is incompatible with the keyword `ElecDipoleFunction`.

Default values:

By default the Frank Condon approximation is not assumed.

7.4.6 Electronic Transition

The energy difference between the botton of the two wells in a vibronic transition. job1%ener-job2%ener

General syntax:

`ElecTransEnergy` *`energy unit`*

Incompatibilities:

Have meaning only for ...

Default values:

xaasdsad

7.4.7 CM System of Reference

Calculate the differential cross section in the centre of mass (CM) system of reference can be done with the keyword `CenterOfMassSys`.

General syntax:

`CenterOfMassSys`

Incompatibilities:

Have meaning only for a '`DifferentialCrossSec`' analysis.

Default values:

By default the differential cross section is calculated in the laboratory system of reference.

7.4.8 Dunhan Parameters

The Dunhan parameters that will be computed in an '`DunhanExpansion`' analysis can be defined with the keywords `DunhanNumberOfParam`, `DunhanPureVibrational`, or `DunhanPureRotational`.

General syntax:

`DunhanNumberOfParam` *`n_vib n_rot n_vib_rot`*

`DunhanPureVibrational` *`n_vib`*

`DunhanPureRotational` *`n_rot`*

The arguments *n_vib*, *n_rot* and *n_vib_rot* stands for the number of parameters vibrational, rotational and vibrational-rotational coupling in the Dunhan expansion. (See Chapter 1 [Overview], page 1.)

Incompatibilities:

Have meaning only for a '`DunhanExpansion`' analysis.

Default values:

There are no default value for the number of Dunhan parameters. One (and only one) of these keywords must be given.

7.4.9 Cross Section Unit

The unit to print the cross section can be given with the `CrossSecUni` keyword. *This keyword can also be used in an* `SCATT_LENGTH` *calculation, since the corresponding zero energy cross section is automatically computed.*

General syntax:

`CrossSecUni` *`unit`*

The argument *unit* is an unit of distance.

Incompatibilities:

Have meaning only for '`PartialCrossSec`', '`TotalCrossSec`', '`DifferentialCrossSec`' analysis and `SCATT_LENGTH` calculation.

Default values:

By default the cross section are printed in atomic units, a_0^2 .

8 Code

What you need now to change the source code.

Keyword Index

Concept Index