

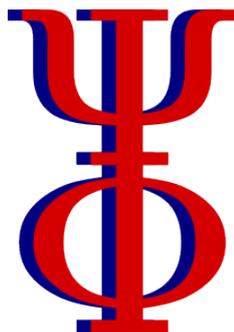
Quantics Exercises

Eryn Spinlove and Graham Worth

October 2024

Version: 1.1

QUANTICS



A suite of programs for molecular QUANTum dynamICS simulations

Worth Group

University College London

Department of Chemistry

Quantics Exercises

Eryn Spinlove and Graham Worth

October 2024

Eryn Spinlove and Graham Worth

Quantics Exercises

Handout, October 2024

Contributors: The Worth Group

University College London

Department of Chemistry

Faculty of Mathematics and Physical Sciences

20 Gordon Street

London

WC1H 0AJ

Abstract

Quantics is a general purpose program package to simulate the evolution of a molecular system by solving the time-dependent Schrödinger equation.

The code is based around the multi-configurational time-dependent Hartree (MCTDH) algorithm in various variants, including the powerful multilayer-MCTDH algorithm that has been used to propagate a wavefunction for up to 1000 degrees of freedom. MCTDH uses a contraction of basis set representations of the Hamiltonian and wavefunction, and Quantics includes a range of variable representation (DVR) grid basis sets and collocation methods.

Input is via ascii text files and for molecules with analytical potential functions no programming is required. A library of potential functions is included to treat more complicated cases, and further functions can be added as required by the user.

The code also includes the variational multi-configurational Gaussian (vMCG) method that is based on a Gaussian wavepacket expansion of the wavefunction. vMCG can be run in a “direct” manner (DD-vMCG), calculating the potential energy surfaces on-the-fly using a number of quantum chemistry programs. In addition to wavepacket propagation, Quantics can solve the time-independent Schrödinger equation for small systems and can solve the Liouville–von-Neumann equation to propagate density matrices. The Package includes auxiliary programs to help set up calculations and analyse the output.

Quantics is a community code of the UK Collaborative Computational Project for Quantum Dynamics (CCPQ) and the European E-CAM project, an e-infrastructure for software development run by the Centre Européen de Calcul Atomique et Moléculaire (CECAM). Through this it has become a framework for general dynamics codes, for example enabling an external surface hopping code to use the Quantics input and operator interfaces.

More details of the code and its capabilities are in Worth Comp. Phys.
Comm.(2020) 248, 107040.

Contents

1	Introduction	1
1.1	Starting Out	1
1.2	What's In This Handout	2
2	Exercise 1: One-Dimensional Oscillator	3
2.1	First Calculations: The Harmonic Oscillator	3
2.1.1	Operator File	3
2.1.2	Input File	4
2.1.3	Running the Calculation	4
2.1.4	Analysing the Results	5
2.1.5	Changing the system	7
2.1.6	Anharmonicity: The Morse Oscillator	8
2.2	Tully Models	9
2.2.1	Simple Avoided Crossing (Tully 1)	9
2.2.2	Dual Avoided Crossing (Tully 2)	12
2.2.3	Extended Coupling with Reflection (Tully 3)	13
3	Exercise 2: Non-adiabatic Dynamics	15
3.1	The Butatriene Cation	15
3.2	Simulating the Dynamics	16
3.2.1	Absorption Spectrum	18
3.2.2	Expectation Values	19
3.3	Stick Spectrum	20
4	Exercise 3: vMCG	21
4.1	Using a GWP basis	21
4.2	Surface Hopping	22
5	Exercise 4: DD-vMCG	25
5.1	Butatriene Cation using Molpro	25
5.2	Water Excitation using Molpro	28
6	Exercise 5: NOCI Photodissociation	31
6.1	The Absorption Spectrum	31
6.2	Computing excitation and reaction probabilities	34

Bibliography

39

Glossary

40

Introduction

” *You can't do better design with a computer, but you can speed up your work enormously.*

— **Wim Crouwel**
(Graphic designer and typographer)

1.1 Starting Out

In the following are a set of example calculations to introduce the basic ideas of quantum dynamics calculations and how they are performed using the Quantics program.

The information given is reduced as the examples progress and looking back to earlier examples should give the answer as to how something should be done. Further information is to be found in the on-line documentation included with the package, where detailed information on keywords available etc. are present.

The Quantics website can be found [here](#), or at

<https://www.chem.ucl.ac.uk/quantics/index.html>

There is a section titled **Quantics Training Workshop, Nov. 2024** with links to this manual and supporting files. You will need a copy of quantics installed on your computer. If \$QUANTICS_DIR is the root directory of the Quantics installation, the documentation is to be found at \$QUANTICS_DIR/doc/index.html

1.2 What's In This Handout

This handout contains the exercises which should help you to successfully run different types of calculations with the Quantics software package.

- **Chapter 2** Setting up and running 1D harmonic oscillator system, introduction to the analysis/plotting programs and running calculations on the Tully model systems
- **Chapter 3** Setting up and running non-adiabatic dynamics on the Butatriene cation and generating spectra from the results
- **Chapter 4** Setting up and running non-adiabatic dynamics using the vMCG (Gaussian wavepacket) method
- **Chapter 5** Setting up and running direct quantum dynamics simulations using the DD-vMCG method
- **Bibliography** Key papers for the methodology/implementation of the methods in each section.
- **Glossary** Key commands and terms with short descriptions

Exercise 1: One-Dimensional Oscillator

” *A picture is worth a thousand words. An interface is worth a thousand pictures.*

— **Ben Shneiderman**
(Professor for Computer Science)

In the first set of exercises we will look at some simple one-dimensional systems to learn the thinking process and basic input required for a Quantics calculation.

2.1 First Calculations: The Harmonic Oscillator

The Hamiltonian for a 1D harmonic oscillator is

$$H = -\frac{1}{2m} \frac{\partial^2}{\partial q^2} + \frac{1}{2} m \omega^2 q^2 \quad (2.1)$$

where m is the mass and ω the frequency. Note that we are using atomic units in which $\hbar = 1$ so it can be omitted.

2.1.1 Operator File

To specify a Hamiltonian, you need an *operator file*. The operator file “ho.op” codes the Harmonic oscillator Hamiltonian above. You will see there are 3 SECTIONS.

The OP_DEFINE-SECTION contains a title that will be printed in output.

The HAMILTONIAN-SECTION defines the operator. The line beginning “modes” defines the fact there is a single coordinate, labelled “x”. The 2 lines between the dashes (these are not required for the operator, and are included to help readability) define 2 terms that can be related to the Hamiltonian 2.2.

The PARAMETER-SECTION defines the mass and frequency. Values of a proton mass with a frequency close to 3500 cm^{-1} are given. This is an approximate C–H stretch.

2.1.2 Input File

To run a calculation you need an *input file*. Look at `ho.inp`. Again there are various SECTIONS.

The OPERATOR-SECTION defines the file containing the operator - here "ho.op".

The PRIMITIVE-BASIS-SECTION defines the “primitive basis set” - a grid of points on which the wavepacket will be propagated. Here 121 points between $x = -2 \text{ au}$ and $x = 2 \text{ au}$ are defined. ($1 \text{ au} \sim 0.5 \text{ \AA}$).

The SPF-BASIS-SECTION defines the number of “single-particle” basis functions that will be used to describe the evolving wavepacket. For a 1D problem only 1 function is required.

The INIT_WF-SECTION defines the initial wavefunction. A Harmonic oscillator ground-state eigenfunction is specified here for mode x . The 4 parameters specify the centre of the wavepacket, its momentum, and the frequency and mass of the harmonic oscillator. Notice that the oscillator has been displaced to $x = 2.0 \text{ au}$.

Finally, the RUN-SECTION defines what will be calculated. Here we are running a propagation that will last for 20 fs with output written ever 0.5 fs. Output will be written to a directory “ho”.

2.1.3 Running the Calculation

Step 1. Set-up the directory and run the calculation

- `mkdir ho`
- `quantics ho`

It should finish in a few seconds.

Step 2. Move into the directory and list the contents

- `cd ho`
- `ls`

There will be a number of files including `psi`

2.1.4 Analysing the Results

Step 3. Plot the potential surface and wavepacket motion using the `showsys` program. This is driven by a simple menu. Type the number to activate the desired option and follow the instructions. You might have to type enter a number of times to be able to enter an option. To start type

- `showsys -gnupl4`

The option `-gnupl4` is needed to use the version of gnuplot installed on the cluster. A menu will appear. Look at item 10, the plot task. Type

- 10

A menu will appear with the possible plot tasks. Select

- 2 `plot pes`

Now type 1. This activates “plot to screen” and a plot of the potential surface will be shown. This should be a quadratic curve. Try changing the energy units using option 90 to cm^{-1} by using the keyword `cm-1`. Try changing the range over which the PES is plotted using options 30 (to change x -axis) and 70 (to change y -axis).

Step 4. Now plot the wavepacket motion. From option 10 choose

- 7 `plot reduced density`

The density is Ψ^2 , i.e. the probability density for the particle. Reduced density means that the density has been integrated over any other degrees of freedom to allow the 1D plot to be made. Type option 280 (this is not visible in the menu), select

- 2 = `movie`

and then type 1, it will provide a movie of the motion of the wavepacket.

The wavepacket will move and the time shown at the top of the plot will be incremented. The plots should be 0.5 fs apart - the value of `tpsi` in the RUN-SECTION. If this was given a different value the snapshots would be at this new value.

It is also possible to view the motion of the wavepacket as stepwise snapshots by skipping using option 280 and directly typing 1.

Step 5. Plot the trajectory of the centre of the wavepacket. The easiest way to do this is to use another program that will extract the expectation value of the coordinates stored in the file `check`. Type

- `rdcheck -g qdq 1 1`

This will produce a file `qdq_1_1.pl` for the first mode of the first (and only) state. The option `-g` meant that this file is ready to be plotted using the `gnuplot` program. Type

- `gnuplot`

A prompt `gnuplot>` appears. To plot column 1 against 2 of the data using a line plot, type

- `plot "qdq_1_1.pl" u 1:2 w l`

This is the expectation value of the coordinate, which can be interpreted as the centre of the wavepacket. It should oscillate from 2.0 to -2.0, and should agree with what you saw as the wavepacket motion.

The time-period of a harmonic oscillator is given by

$$T = \frac{2\pi}{\omega}$$

If T is in fs and ω in cm^{-1} , then using the appropriate conversions factors

$$T = \frac{33356}{\omega}$$

Consider This: For the system above what is the time period? Is that what is seen?

Column 3 contains the width of the wavepacket, defined as the standard deviation $\sqrt{\langle x^2 \rangle - \langle x \rangle^2}$. This is plotted using

- `plot "qdg_1_1.pl" u 1:3 w l`

It also oscillates, but looking at the values is nearly constant.

2.1.5 Changing the system

To run a similar, but altered, calculation move back up to the starting directory and copy the operator and input files.

- `cd ..`
- `cp ho.op ho1.op`
- `cp ho.inp ho1.inp`

Change the frequency of the oscillator in `ho1.op` to give a time period of 35 fs (change the parameter in the operator file). Run a propagation of 70 fs (2 time periods) by changing `tfinal` in the new input file. Also change the value of the `name` variable so that all the results are written to a directory `ho1`. Now make the new directory and run the new calculation

- `mkdir ho1`
- `quantics ho1`

Change directory into `ho1` and use `showsyst -gnupl4` to plot the potential surface and wavepacket motion as before.

Consider This: Is your result what you would expect?

To see the effect of different starting conditions, first try altering the starting points. To do this, change the centre of the initial wavepacket in `ho1.inp` (the first parameter in the line defining the initial wavepacket in the INIT_WF-SECTION). Also try different initial widths (change the frequency - the third parameter in the line defining the initial wavepacket in the INIT_WF-SECTION).

Points to note:

- The time period does not depend on the initial displacement.

- If the frequency defining the initial wavepacket width matches the frequency of the potential function the wavepacket remains the same width at all times.

2.1.6 Anharmonicity: The Morse Oscillator

Finally we shall change the harmonic oscillator to a Morse oscillator. To replace the harmonic potential with a Morse function. involves replacing the $\frac{1}{2}\omega q^2$ operator with a label such as V and defining the Morse operator in a LABELS-SECTION. We will base the operator on the harmonic operator file `ho.op`

- `cp ho.op mo.op`

Now edit `mo.op`.

- In the PARAMETER-SECTION add parameters for the system mass (we will use mass-scaled coordinates) and the well depth.

```
depth = 8.0, ev
mass_x = 1.0, amu
```

- In the HAMILTONIAN-SECTION replace the KE and potential operators with

```
1.0          | KE
1.0          | V
```

- The label V now needs to be defined by adding a LABELS-SECTION

```
LABELS-SECTION
V = morse[depth, omega, 0.0, 0.0, mass_x]
end-labels-section
```

- `mkdir mo`

Using the input file `mo.inp` run and look at the dynamics. Note that it looks much like the harmonic systems. Now move the initial wavepacket out, first to be centred at -0.2 and then -0.3 . Notice the effects of the anharmonicity.

2.2 Tully Models

In the next set of exercises we will look at some simple scattering problems with the Tully Models. These were set up by John C. Tully [3] to provide models to test the performance of surface hopping calculations in including non-adiabatic effects in scattering calculations.

The Hamiltonian for the Tully Models is written in a diabatic basis:

$$H = -\frac{1}{2m} \frac{\partial^2}{\partial q^2} + \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix} \quad (2.2)$$

with m is the mass and is set to 2000. Note that we are using atomic units in which $\hbar = 1$ so it can be omitted. As a comparison, a proton in atomic units has a mass of 1822.

2.2.1 Simple Avoided Crossing (Tully 1)

The operator file is `tully_scm.op` and `tully_scm.inp` the input file. The potentials are:

$$\begin{aligned} V_{11} &= A(1 - \exp(-Bx)) \quad , \quad x > 0 \\ V_{11} &= -A(1 - \exp(Bx)) \quad , \quad x < 0 \\ V_{22} &= -V_{11} \\ V_{12} = V_{21} &= C \exp(-Dx^2) \end{aligned}$$

with empirical parameters A, B, C, D .

Step 1. Set-up the directory and run the calculation

- `mkdir tully_scm`
- `quantics tully_scm`

It should finish in a few seconds.

Move into the directory and analyse the results.

Step 2. Plot the potential surface and wavepacket motion utilising the `showsys` program using the same options as for the harmonic oscillator (Section 2.1.4). There are now 2 states and the state of interest can be selected by

- `60 = change state selection (1)`

Type `60`. In response to the question “How many states will be plotted?” select 1. Then in response to “Enter selection” enter 2. The wavepacket or potential can now be plotted for the upper state.

Type `60` again. In response to the question “How many states will be plotted?” now select 2. Then in response to “Enter selection” enter “1 2”. The wavepacket or potential can now be plotted for both states.

You should see that after 60fs the crossing region is reached and population transfers to the state 2 state where it passes on the end of the grid, while on state 1 it reflects back in the incoming channel. You will also see that at later times there is reflection from the end of the grid that causes interference with the outgoing wave.

Step 3. To remove this interference, edit the file `tully_scm.inp` to comment in

```
#alter-labels
#CAPl_x = CAP[-12.0 0.0001 3 -1]
#CAPr_x = CAP[12.0 0.0001 3]
#end-alter-labels
```

i.e. remove the “#” at the start of each line. This will add CAPs at the ends of the grid. Rerun the calculation

- `quantics -w tully_scm`

the “-w” option overwrites the previous results. If you now plot the wavepacket it will be absorbed before reaching the end and the interference will not be present.

Step 4. Analyse the branching ratio, which is percentage of the wavepacket that is transmitted through the avoided crossing (exits to RHS of the grid) divided by the percentage of the wavepacket that is reflected (exits to LHS of grid). To do this we integrate the flux going into the CAP over the propagation using the program `flux`. This program has many possible uses - type `flux -h` to see all the options.

First go into the directory with results

- `cd tully_scm`

Now run `flux` using the option `-wtt` to calculate the expectation of the flux going into the CAP at the left hand end of the x coordinate (reflection). Save the output files `wtt` and `iwtt` that contain the flux and integrated flux, respectively.

- `flux -wtt x left`
- `mv wtt wtt.left`
- `mv iwtt iwtt.left`

Do the same for the flux going into the CAP at the right hand end of the x coordinate (transmission).

- `flux -wtt x right`
- `mv wtt wtt.right`
- `mv iwtt iwtt.right`

Now plot the results using `gnuplot`. First look at the fluxes.

- `gnuplot`
- `plot "wtt.left" u 1:2 w l title "reflected"`
- `replot "wtt.right" u 1:2 w l title "transmitted"`

These should go to zero at long times to indicate the scattering has finished. Now look at the integrated flux.

- `plot "iwtt.left" u 1:2 w l title "reflected"`
- `replot "iwtt.right" u 1:2 w l title "transmitted"`

You will see that the integrated fluxes reach a plateau. The final values give the branching ratio of transmitted/reflected $\approx 0.55/0.45$.

Step 5. Repeat the calculation with different initial momenta. For each calculation copy the input file to a new name, e.g. `cp tully_scm.inp tully_scm1.inp` . Edit this new file

- change the name directory so the data is put in a new location, e.g.
name = `tully_scm1`.

- change the momenta in the `INIT_WF-SECTION` by editing the number 4.5 in the line:
`x Gauss -8.0 4.5 0.5`
- Make the new directory `mkdir tully_scm1` and run the new calculation.

If you look in the documentation, the 3 numbers in the initial wavepacket definition define the position, momentum and width, respectively.

Repeat the analysis and see if more or less wavepacket is transmitted.

2.2.2 Dual Avoided Crossing (Tully 2)

The potentials for the Dual Avoided Crossing are defined as:

$$\begin{aligned} V_{11} &= 0 \\ V_{22} &= -A \exp(-Bx^2) + E_0 \\ V_{12} = V_{21} &= C \exp(-Dx^2) \end{aligned}$$

with empirical parameters A, B, C, D, E_0 . Get the operator for this model, `tully_dac.op` and set up the input file for a calculation

- `cp tully_scm.inp tully_dac.inp`

Edit the input file to use the different operator.

- change the name directory so the data is put in a new location, e.g.
`name = tully_dac`
- change the name of the operator file e.g. `opname = tully_dac`
- Make the new directory `mkdir tully_dac` and run the new calculation.

Analysis can be carried out as for the Tully Model 1.

2.2.3 Extended Coupling with Reflection (Tully 3)

Repeat the calculations for the Tully model 3, the Extended Coupling with Reflection by copying and editing the input file `tully_scm.inp`. The potentials are:

$$\begin{aligned}V_{11} &= A \\V_{22} &= -A \\V_{12} = V_{21} &= B \exp(Cx) \quad , \quad x < 0 \\V_{12} = V_{21} &= B (2 - \exp(-Cx)) \quad , \quad x > 0\end{aligned}$$

with empirical parameters A, B, C . This operator is coded in `tully_ecwr.op`.

Exercise 2: Non-adiabatic Dynamics

” *Don't start anything new on a Friday afternoon.*

— **Graham Worth**

(Professor of Computational Chemistry)

We shall now try a multi-dimensional, multi-state calculation that uses the MCTDH algorithm. We shall also check the quality of the calculation in addition to plotting the spectrum and analysing the dynamics.

3.1 The Butatriene Cation

The butatriene (C_4H_4^+) is a classic example of a molecule with a conical intersection between the two lowest states. The Hamiltonian to first-order can be written using a vibronic coupling model as

$$\mathbf{H} = \sum_{i=1,18} \frac{\omega_i}{2} \left(\frac{\partial^2}{\partial Q_i^2} + Q_i^2 \right) + \sum_{i=8,12,14,15} \begin{pmatrix} \kappa_i^{(1)} Q_i & 0 \\ 0 & \kappa_i^{(2)} Q_i \end{pmatrix} + \begin{pmatrix} 0 & \lambda Q_5 \\ \lambda Q_5 & 0 \end{pmatrix} \quad (3.1)$$

Note that only 4 modes provide on-diagonal coupling and 1 mode off-diagonal coupling. This Hamiltonian has been coded up in the file `C4H4_linear.op` with the input file `butatriene.inp`.

In order to view the PESs, first run a test calculation using the `-t` option, in which the calculation stops after calculating the initial energy.

- `mkdir butatriene`
- `quantics -t butatriene`
- `cd butatriene`
- `showsys -gnupl4`

Option 20 is used to select the modes for the plot. Check that it says

- 20 = change coordinate section (v10a=x v6a=y)

If not, type 20 and use the option to select these modes. This option can be used to select any 1D or 2D cut through the surfaces. Use menu item 60 to plot both electronic surfaces. Now use option 10 to select what is to be plotted. We want to plot a pes. Look at the differences between the diabatic and adiabatic pictures. The conical intersection is clear in the adiabatic picture where the surfaces meet. The best plot is to set the surfaces (option 160) and the z-axis to be between 0 and 2.0 (option 70).

3.2 Simulating the Dynamics

Move back up to the starting directory and run the dynamics.

- `quantics -w butatriene`

This simulates a vertical excitation from the ground-state to the upper state. Move back into the *name* directory `butatriene` to analyse the calculation.

First, we need to check that the calculation is of a good enough quality, i.e. check for convergence of the basis sets. Start up the analysis program `rdcheck`

- `rdcheck natpop`

The natural orbitals are an ordered representation of the SPFs and for a good calculation the population of the highest natural orbital must be small, i.e. the basis function plays a small role in the wavefunction. This calculation has 2 states and 2 modes. Enter 0 for the mode and the state to be analysed, and the maximum population of the highest natural orbital is printed. The highest orbitals have negligible populations, so the basis is good.

Next check the “primitive grid” - we are using a DVR basis here so we will actually be checking whether we have enough primitive basis functions to describe the SPFs, but we loosely refer to this basis as the grid. Run the program `rdgpop`. In answer to the question “no. ==?” enter 1. This will return the population of the grid points at either end of the grid. Entering

a larger number will provide a population summed over the selected no. of points from the end. Select `dof 0`. The maximum population for the modes are then printed to screen. The population along the first mode is quite large indicating that the primitive basis set might be too small.

Now plot the diabatic state populations using the program `statepop`. In the directory with the results files, type

- `statepop -inter -gnupl4`

Select option 1 - "Plot to screen". The resulting plot shows how the system begins in one of the components of the degenerate state and oscillates rapidly between the two.

Next, plot the adiabatic populations. To do this we need to generate the populations and use `gnuplot` to plot them. Type

- `adpop`
- `gnuplot`

In `gnuplot`, type

- `plot "adp" u 1:2 w l`
- `replot "adp" u 1:3 w l`

The resulting plot shows that the system starts with some initial population in the lower adiabatic states. As time goes on, the population moves steadily to the lower state.

Finally, plot the evolving wavepacket using the analysis program `showsys`. To select the adiabatic or diabatic wavefunction use option 10. To select the wavepacket on either state use option 60. The pictures give a similar result. The wavepacket in the upper state remains compact and oscillates in the upper cone along $v6a$. The wavepacket appears in the lower state at around 30 fs when the wavepacket has reached the intersection. It then oscillates around the ground-state surface, spreading in both directions showing that the torsion, $v10a$, has been excited by the internal conversion through the intersection.

3.2.1 Absorption Spectrum

An absorption spectrum is calculated from the Fourier Transform of an autocorrelation function, i.e.

$$I(\omega) \propto \int_{-\infty}^{\infty} dt C(t) e^{i\omega t}$$

where

$$C(t) = \langle \Psi(0) | \Psi(t) \rangle$$

Assuming a *vertical excitation*, similar to that in Fig. 3.1, we can simply propagate the ground-state wavefunction placed “vertically” onto the excited state. The keyword `init_state = 2` in the `INIT_WF`-SECTION of the butatriene simulation is the command that places the wavepacket on the upper state.

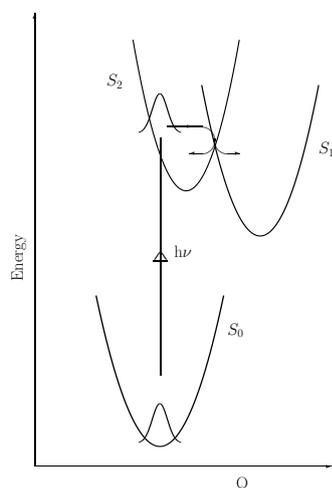


Fig. 3.1: Schematic diagram of vertical excitation from S_0 into S_2

In order to plot the autocorrelation function we need the program `rdauto`. Type

- `rdauto -inter`

A menu will appear (`-inter` is an argument that provides this). Plot the absolute value of the autocorrelation function. This is the default plot task and so this can be done by simply typing `1`. This is a periodic function with the time period of the harmonic motion that starts with a value of `1` (complete overlap) and falls to `0.8` as the wavepacket moves away.

To plot the spectrum, type

- `autospec -inter`

You will be asked to enter `E-min`, `E-max`, `unit =?` (e.g: `2.`, `5.`, `ev`), i.e. the energy bounds for the spectrum and the units. Enter `8.5 10.5 ev`. In response to `tau =?` (damping time in fs) respond with `30` and for `iexp =?` (damping order) enter `1`. Note that a larger damping time leads to narrower peaks in the spectrum) and the damping order is the exponent of the cutoff function that ensures the correlation function goes to zero at time `T`.

3.2.2 Expectation Values

It is possible to obtain the expectation value of an operator of interest during a simulation by coding the operator and using the `expect` keyword in the `RUN-SECTION`. Here we look at the situation of how to obtain the expectation value after the propagation has been run.

We want to obtain a measure of the energy flow out of mode Q_{14} in state S_2 to both S_1 and to Q_5 . For this we will calculate the expectation of the operator for the 1D oscillator

$$\mathbf{O} = \frac{\omega_{14}}{2} \left(\frac{\partial^2}{\partial Q_{14}^2} + Q_{14}^2 \right) + \kappa_i^{(2)} Q_{14} |2\rangle \langle 2| \quad (3.2)$$

To code this, add to the operator file `C4H4_linear.op` the following lines

```
HAMILTONIAN-SECTION_H0energy
modes | e1 | Q_1 | Q_2 | Q_3 | Q_4
modes | Q_5 | Q_6 | Q_7 | Q_8 | Q_9
modes | Q_10 | Q_11 | Q_12 | Q_13 | Q_14
modes | Q_15 | Q_16 | Q_17 | Q_18

omega_14 |1 S2&2 |2 KE
0.5*omega_14 |1 S2&2 |2 q^2
kappa2_14 |1 S2&2 |15 q
end-hamiltonian-section
```

This will define an operator with the label `H0energy` to distinguish it from the system Hamiltonian that has no label.

We now need to add this operator file to the `oper` file so that it can be used by `Quantics`. To do this, edit the input file:

- In the `RUN-SECTION` remove the keyword `propagation` and replace it with the keyword `genoper program`.

now run

- `quantics -w butatriene`

Change into the output directory and run the `expect` program

- `expect`

A list of possible operator will be given. Select `H0energy`. This will provide a file.

3.3 Stick Spectrum

For small systems it is possible to solve the time-independent Schrödinger equation and diagonalise the Hamiltonian to produce a stick spectrum of the eigenvalues. Copy the file `butatriene.inp` to `but_stick.inp`. Replace the `RUN-SECTION` with

```
RUN-SECTION
name = but_stick   diagonalise = 50
end-run-section
```

and run the `Quantics` calculation.

Move into the results directory. There will be a file `eigval`. Plot the stick spectrum using the analysis program `rdeigval -inter`. Compare the resulting spectrum to that from the time-dependent calculation, the same peaks should be present in both. The time-dependent spectrum could be improved by a longer propagation time and longer damping factor in the spectrum calculation to make the lines sharper.

Exercise 3: vMCG

” *Users do not care about what is inside the box, as long as the box does what they need done.*

— **Jef Raskin**

about Human Computer Interfaces

The vMCG method replaces the single-particle functions of MCTDH with Gaussian wavepackets (GWP). In the following exercises we will use this basis type on analytical potential surfaces to see the extra analysis available.

4.1 Using a GWP basis

Take the butatriene input file (butatriene.inp) and create a new input file, butatriene_gwp.inp changing the `name` directory to `butatriene_gwp` for the new output. Now change the primitive basis to a GWP basis. I.e in the `PRIMITIVE-BASIS-SECTION` replace on all the lines

```
H0      10      0.0      1.0      1.0
```

with

```
GWP
```

Also change the 1-dimensional SPFs to be multi-dimensional GWPs by replacing

```
SPF-BASIS-SECTION
multi-set
    Q_5      = 8,8
    Q_14     = 8,8
end-spf-basis-section
```

to

```
SPF-BASIS-SECTION
single-set
    Q_14, Q_5    = 8
end-spf-basis-section
```

The keyword `single-set` means that we do not use a separate set of functions of functions for each state.

Run this calculation.

Plot the state populations using

- `statepop -inter -i butatriene_gwp`

and compare it to the state populations obtained from the MCTDH calculation

- `statepop -inter -i butatriene`

The GWP result has significant differences due to the small basis set size. Rerun the calculation with more GWPs by editing the line `Q_14, Q_5 = 8`, increasing the basis set until convergence is reached. I.e. the state populations match that of the MCTDH calculation.

To plot the trajectories at the centres of the wavepackets, use the analysis program `gwptraj`

- `gwptraj -i butatriene_gwp -gnupl4`

4.2 Surface Hopping

Quantics is coupled to a surface hopping code [1] that uses Tully fewest switches. To demonstrate how it works we will use a simple 2-dimensional model of a conical intersection from Ferretti [2].

Take the input file `ferretti.inp` and the associated operator `ferretti.op`. This is for a numerically exact simulation starting on the upper surface.

Run this calculation. For analysis, plot the surfaces using `showsys` and then the diabatic state populations using `statepop`. The population transfer is at around 40 fs and almost complete.

Calculate and plot the adiabatic populations:

- `cd ferretti`
- `adpop`

and then use `gnuplot`

- `plot "adp" u 1:3 w l`
- `replot "adp" u 1:4 w l`

You will see that the transfer is minimal. If you look at the wavepacket (use `showsys`) you will see that it all stays on the upper adiabatic surface and the crossing is purely diabatic. Note that we are plotting the populations for states 2 and 3 as the model includes the ground state (state 1).

Run the calculation again, but placing the initial wavepacket further away from the intersection. To do this edit the `INIT_WF-SECTION`, replacing

- `x HO 2.0 0 0.001 20000`

with

- `x HO 4.0 0 0.001 20000`

The conical intersection is at $x = 3.5$ so this moves the wavepacket away. As a result, the wavepacket has a higher momentum as it reaches the crossing. Plot the diabatic and adiabatic populations. You will see that now the diabatic transfer is less, but the adiabatic transfer is significant and permanent.

Now take the input file `ferretti_tsh.inp`. This is for a surface hopping calculation with 10 trajectories. It is a complicated input as we need to transform the coordinates to Cartesian coordinates (the `tshtrans-section` and provide the parameters for the surface hopping code. These are in the `SH_ZAGREB-SECTION` (the code is known as the “Zagreb Surface Hopping program”). Here parameters such as the time step and random number

generator seed are defined. Other than that, the input is similar to that for an MCTDH calculation, with the same initial wavepacket definition.

The code samples this initial wavepacket using a Wigner distribution and runs each trajectory in turn, storing them in a directory `ferretti_tsh/zagreb_trj`. When they are all completed the code constructs the files needed for analysis. To plot the state populations, use

- `statepop -inter -gnupl4`

as before. This time, adiabatic populations are plotted. They compare quite well to the results from the MCTDH simulation starting at $x = 0$. For better convergence, add some more trajectories by editing the keyword `swarm = 10` in the `RUN_SECTION`. Rerunning the calculation will add trajectories to the swarm.

Exercise 4: DD-vMCG

” *Sometimes it is best to make a tea or coffee and come back to it.*

— **Graham Worth**

(Professor of Computational Chemistry)

In this exercise we will look at how to set up and run a DD-vMCG calculation in which the potential energy surfaces are generated on-the-fly using an external quantum chemistry code. The ideas behind the calculations are the same as those used in the vMCG and MCTDH calculations of the previous exercises. Calculations are run using normal mode coordinates as used in the vibronic coupling model.

5.1 Butatriene Cation using Molpro

Get the input file `but_dd.inp` and the directory `but_dddata`. Before running a calculation, you need to

- Obtain an optimised ground-state state and frequencies. These define the coordinates of the system.
- Decide on the level of theory for the excited states.
- Know how to write the input for an excited state calculation, including gradients, non-adiabatic couplings and Hessian.

In this calculation of the butatriene cation we will use the Molpro program, with 2 cation states calculated with a CAS(5,6) calculation.

In `but_dddata` there is a file `but_fc_cas_freq.out` which is a CAS(6,6) calculation of the neutral ground-state frequencies. Note the keyword `freq=hpmodes` which is needed to output the frequencies at higher precision than the standard output.

There is also a file `template.dat` which is a template for the quantum chemistry calculations. It is like a Molpro input file, but certain keywords which are not always needed, such as that to calculate the Hessian, are bracketed with markup-like keywords. The space where the geometry and molecular orbitals are written is defined with `<geometry>` and `<mo-coefficients>`.

Finally, there is a checkpoint file `start.chk` containing the molecular orbitals for the CAS calculation at the Frank-Condon point.

The Quantics input file `but_dd.inp` is different from the MCTDH style of input. In the `RUN_SECTION` the keywords `direct = nmodes` specifies a direct dynamics calculation in normal modes. `ngwp = 5` species a basis of 5 GWPs.

The `INITIAL-GEOMETRY-SECTION` replaces the `PRIMITIVE-BASIS-SECTION` and `INIT_WF-SECTION` sections. It specifies the atoms in the molecule, along with the Cartesian coordinates of the initial (Frank-Condon) geometry. It also specifies the labels and frequencies of the normal modes. Other information such as the number of electronic states (`nstates=2`) and the initial state (`init_state = 2`)

Note that many of the normal mods have the keyword `freeze` at the end of the line. This modes are frozen and only the 5AU and 14AG modes are included in the dynamics. This 2D model is the same as that of the MCTDH exercises.

The final new section is the `DIRDYN-SECTION`. This has the information required to run the direct dynamics interface. The main keywords that need to be set are:

- `data = but_dddata` defines the directory with the DD files and where the database will be stored.
- `transfile = but_fc_cas_freq.log` defines the file with frequencies
- `qcprogram = molpro` defines that Molpro will be used
- `method = cas` defines that CAS will be the electronic structure method used.
- `ener0 = -152.9326464722` defines the energy of the ground-state equilibrium geometry (in Hartree)
- `dd_diab=global` defines the diabatisation method (propagation diabatisation)

- `dbmin = 0.20` defines how far the molecule moves before a new QC calculation is performed.
- `db = rdwr` A read-write database is used. I.e. new calculations are run and added to the DB when required.
- `nbasis = 44` number of Molecular orbitals
- `subcmd = /user/gworth/bin/run_molpro15` script used to run the Gaussian calculation

Run the calculation

- `mkdir but_dd`
- `quantics but_dd`

Much of the analysis follows that used for the vMCG and MCTDH calculations. The diabatic state populations can be plotted using `statepop` and the potential using `showsys`. To obtain the adiabatic populations it is not possible to do the full transformation as this requires multi-dimensional integrals. Instead, we use a simple “saddle-point” approximation that transforms the basis only at the centre of the GWPs. In the results directory use

- `adpop -saddle0`

and then plot the results with `gnuplot`.

The centres of the GWPS can be plotted using `gwptraj`. In addition, we can now plot the molecular structure along the trajectory using the program

- `ddtraj`

In response to the question `Enter: Trajectory No. (Max: 5)` enter a number between 1 and 5. A window is opened using the `Jmol` program. Select

- `Tools → Animate → Once`

to watch the movie. It is possible to watch the average structure etc.

Finally, the database can be examined to see the structure sampled. In the directory `but_dddata`, there is a file `database.sql`. This is an sqlite file containing all the structures, energies, gradients, etc. from the quantum

chemistry calculations. This data forms the potential surfaces used in the dynamics. Type

- `checkdb`

This will report the information in the database and how many structures. To view the structures type

- `rddddb`
- `jmol plot.db`

where the second line starts a jmol window to plot the structures extracted from the database using the `rddddb` program.

The results so far were obtained with a small basis (5 GWPs) and the DB only contains 9 points. To improve the quality of the data, edit the input file and increase the number of GWPs to 20. Now rerun the calculation using

- `quantics -w but_dd`

The `-w` options means that the output in the `but_dd` directory will be overwritten, but the data already collected in the `but_dddata` database will be added to.

At the end of this calculation re-run the analysis.

5.2 Water Excitation using Molpro

The second example is a calculation of the photo-dissociation of water using Molpro. Take the input file `h2o-4s-dd.inp` and the directory for the database `h2o-4s-dd_data`

As before, the DB directory contains a file with the ground-state frequencies `water-opt-nosym.out` along with a `template.dat` file. This time the template is that for a Molpro input file for a SA4-CAS(6,8)/6-31G* calculation. The file `init-orbital.orb` contains the CASSCF molecular orbitals at the Franck-Condon point.

In the Quantics input file `h2o-4s-dd.inp` the sections

```
HAMILTONIAN-SECTION_system
modes | 1A1 | 2A1 | 3B2
1.0 |2 capa
end-hamiltonian-section
```

```
LABELS-SECTION
capa=CAP[10.0,0.01,3,1]
end-labels-section
```

define a CAP along the 2A1 vibration, which is the anti-symmetric O-H stretch.

Run the calculation and analyse the results.

Exercise 5: NOCl Photodissociation

6.1 The Absorption Spectrum

The photodissociation of NOCl is a simple photo-chemical reaction. After excitation from the ground to the first excited state, $S_0 \rightarrow S_1$, the chlorine atom dissociates on a femto-second time-scale. This results in a broad band for the absorption spectrum.

The calculation consists of two stages. First, the ground state wavefunction is generated by energy relaxation of an initial guess wavefunction on the ground state surface, S_0 . The second stage then places this wavepacket on the excited state surface, S_1 , leading to photodissociation.

Copy the files `nocl0.inp`, `nocl0.op`, `nocl1.inp` and `nocl1.op` to your directory, and create there the directories `nocl0` and `nocl1`.

To find the ground-state eigenfunction, we first perform a relaxation calculation. Type

- `quantics nocl0`

To perform the photo-dissociation calculation, the wavepacket from the `nocl0` calculation is read in and projected on the S_1 surface. Type

- `quantics nocl1`

This calculation takes a few minutes.

The calculation can now be analysed. Move to the directory `nocl1` which contains all the data files from the propagation. To watch the system dissociating, start the `showd1d` program

- `showd1d -inter`

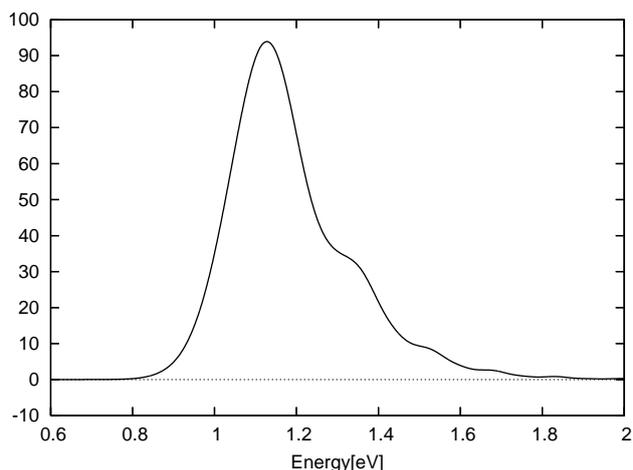


Fig. 6.1: The absorption spectrum for the NOCl molecule on excitation to the S_1 state.

and follow the menu options to plot. First type 1. This gives a 3D plot of the wavepacket as a function of time. Next type 10 to change the plot task and choose 2 to plot time slices. Enter 1 to plot these. You can now make a movie by typing 280 and selecting 1 or 2. With option 1 you need to press return to go through the frames.

To plot the spectrum, type

- `autospec -inter`

You now need to enter the required range for the spectrum, a damping time and a damping order. For the range enter 0.6 2.0 ev. For the damping time 0, for the damping order 1. To plot the spectrum type 1. The result is shown in Fig. 6.1. Note that the spectrum shown is the Fourier-transform of the autocorrelation function times the energy. Hence it is assumed that the ground state energy is at zero, such that energy equals excitation energy. If this is not the case, use option 802 to shift the energy scale.

Save the spectrum to a gnuplot file using option 4 = save plot to a gnuplot file (use after 1 or 2). Give the name `spectrum1.pl`. This file contains the data and commands to plot using `gnuplot spectrum1.pl`

- `gnuplot spectrum1.pl`

The program **showsys** is a powerful tool for plotting 1D and in particular 2D views on wavepackets and potentials. Type

- `showsys`

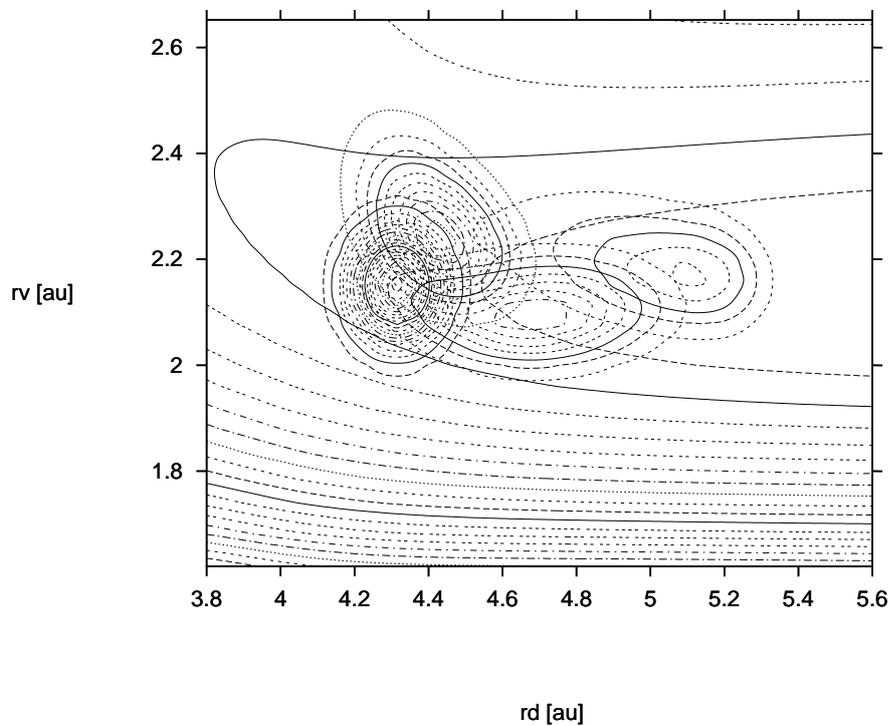


Fig. 6.2: Overlay plot, wavepacket on potential. The Wavepacket density is shown for the times $t = 0, 10, 20, 30$ fs. The density is obtained by integrating $|\Psi|^2$ over all angles, whereas the potential contour lines are obtained by fixing the angle to 2.1 rad.

A menu appears which allows various options to be set. Go to menu point 10 (type 10), and change the plot task to 2 = *plot pes* (type 2). Next input a 1 three times and a 2D cut through the surface (with theta fixed to 1.545 radians) will pop up. Now use menu point 20 = *change coordinate section*, i.e. chose another cut. If one specifies only 1 coordinate with an x and the others with numbers, a 1D plot will appear. After you have played around enough, go back to menu point 20 and input

```
rd x
rv y
theta 2.1
```

Then use menu point 5. You will be asked for a file name. Chose any convenient name, e. g. xyz . The plot data is then written to the file xyz for later use.

Next we want to inspect the wavefunction. Go to menu point 10 and chose 5 = *plot reduced density*. The density, i.e $|\Psi|^2$ integrated over all coordinates, except those specified by x and y (that is integrated over all angles in the present case), will be shown. Input a 1 three times and you will see the initial density. Pressing RETURN will display the density propagated by one time step, and so on. After you have returned to the menu, chose point

400 = Overlay plots and then 410 = File for overlay and enter the file name (xyz). Turn off the key using option 240. After inputting 1's you will now see an overlay plot, i. e. the wavepacket on top of the contour lines of the potential.

With menu points 240 and 245 one may switch off the legend (or keys) and the title. Menu point 285 allows to take larger time steps and with point 280 one may switch to different plot forms, e. g. to plot all time slices at once. Such a plot is shown in Fig. 6.2.

Inspect the ASCII files of the output directory, in particular output, log, and timing. The file input contains a copy of the input file, the options, and the operator file. Thus, it tells you exactly what you have been doing. Since an NOCI run is so fast, NOCI is ideally suited for testing. Just play around with it! You may e. g. change the numbers of single particle functions or alter the integrator accuracies.

6.2 Computing excitation and reaction probabilities

The analysis of scattering processes, i. e. the computation of excitation and reaction probabilities, is performed by evaluating the quantum flux going into a particular channel. The quantum flux is determined by the interaction of the time-dependent wavepacket with a CAP. The program **flux** performs the necessary analysis. The psi file is read and the energy resolved flux is computed and written to the flux file. Additionally this flux is divided by the energy distribution of the initial wavepacket to obtain the transition or reaction probabilities. The latter step requires in general that the **quantics** program has computed the energy distribution of the initial wavepacket and has written it to the enerd file. Besides the flux file, **flux** creates the files flux.log, gtau and wtt. When the file gtau is present the flux program skips the time consuming evaluation of the integrals $\langle \Psi(t)|W|\Psi(t + \tau) \rangle$ but reads the function $g(\tau)$ from the gtau file. (See the MCTDH review, section 8.6.3, Eq.(199), for more details). This allows to re-do the Fourier integrals for another energy interval very quickly. The option -w enforces the re-calculation of $g(\tau)$.

The file wtt contains the expectation values $W_{tt} = \langle \Psi(t)|W|\Psi(t) \rangle$. This information is useful for checking that the absorption process has finished.

The shell script **plwtt** visualises the function W_{tt} , while **plflux** and **plflux -r** visualise the flux and the transition probability, respectively.

The program **flux** cannot only determine the total energy resolved flux going into a particular arrangement channel (i. e. going into a particular CAP) but can also determine the flux which is projected onto final quantum states or which is weighted by an operator. This is probably best demonstrated by an example. We also want to plot the flux as a function of time so want to calculate the expectation value of the flux operator in the dissociation channel.

To analyse the result, move into the directory `nocl1`. First plot the flux as a function of time. This is contained in the file `expectation`. Start up `gnuplot` and type

- `plot "expectation" u 1:2 w l`

This plots the norm which is close to 0 at the end of the propagation.

- `plot "expectation" u 1:3 w l`

plots the flux. There is a set of peaks, dominated by a first, fast component and a second, slower component.

Next evaluate the total flux as a function of energy and save it in the file `flux.0`

- `flux -w 0.61 2.0 ev rd`
- `mv flux flux.0`

To display it, type

- `plflux -f flux.0`

As you will notice, the plot looks very similar to the absorption spectrum calculated above (and shown in figure 6.1). It is not identical, though, as the definition of an absorption spectrum contains a factor ω , the energy of the absorbed photon. This multiplication is omitted when **autospec** is run with the `-FT` option. Type

- `autospec -FT 0.61 2.0 ev 0 1`

which writes the spectrum to the file `spectrum.pl`. Now the spectra are identical as one observes when typing

- `plflux -G -f flux.0 -d spectrum.pl`

This shows the flux and the Fourier transform of the autocorrelation function on top of each other.

Next we modify the flux by letting the operator of the vibrational energy act on it.

- `flux -w -O vib -u 200. 0.61 2.0 ev rd`
- `mv flux flux.op`

To view this

- `plflux -G -f flux.0 -d flux.op`

and you will see the total flux in comparison with the vibrational energy weighted flux. It is now clear that the structures in the spectrum are due to vibrational excitation of the NO fragment. By the way, via the option `-u 200` the modified flux was multiplied by the factor 200. This was done to make it comparable with the total flux. Usually the option `-u` is followed by an energy keyword, e.g. `-u ev`, to transform the weighted flux from a.u. to a desired energy unit.

One may divide the weighted flux by the total flux to observe the vibrational energy content by using the option `-ed flux.0`

- `flux -ed flux.0 -O vib -u ev 0.61 2.0 ev rd`
- `mv flux flux.op_r`

To visualise the quotient `weighted-flux/total-flux`, which is related to the reaction probability, type

- `plflux -G -r -f flux.op_r`

The structures below 0.8 eV are numerical noise because one divides a very small number by another very small number. Obviously, the higher the energy of the absorbed photon, the larger is the vibrational energy of the NO fragment and the higher the chance of dissociation.

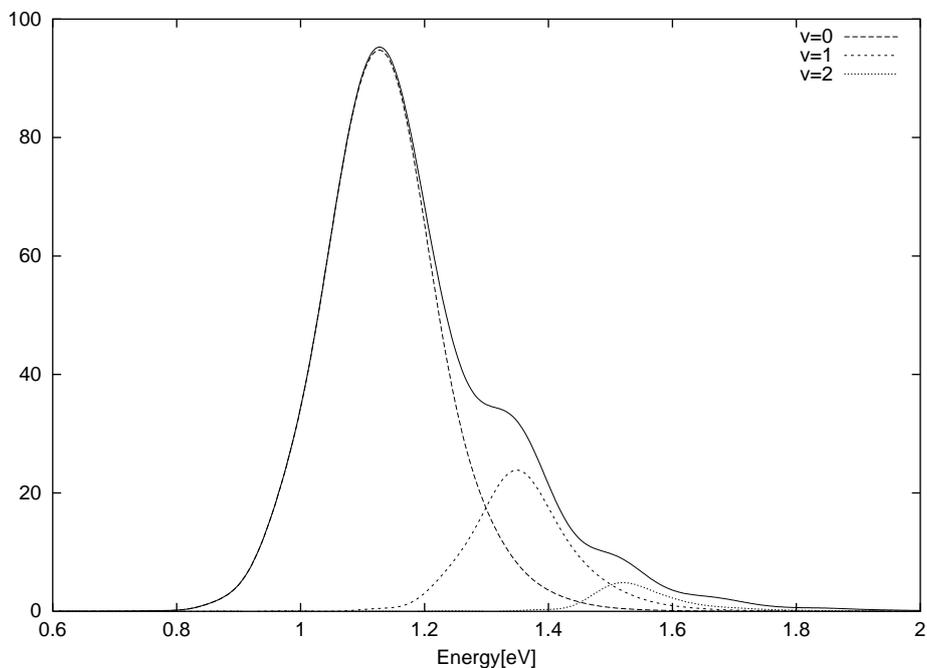


Fig. 6.3: The total (full line) and projected flux of dissociating NOCl. The projection is on the vibrational states of the NO fragment, $v=0$, $v=1$, and $v=2$, respectively.

Finally we demonstrate the use of projectors. To project onto the lowest 3 vibrational eigenfunctions of the NO Hamiltonian, type

- `flux -w -P 2 eigenf vib 1 % 0.61 2.0 ev rd`
- `mv flux flux.1`
- `flux -w -P 2 eigenf vib 2 % 0.61 2.0 ev rd`
- `mv flux flux.2`
- `flux -w -P 2 eigenf vib 3 % 0.61 2.0 ev rd`
- `mv flux flux.3`

To display the results type

- `plflux -G -f flux.0 -d flux.1 -e flux.2`
- `plflux -G -a 1.0 -y 25. -f flux.0 -d flux.2 -e flux.3`

The first plot shows the flux projected onto the vibrational ground state and the first excited state, respectively, in comparison with the total flux. The second plot is similar, but shows the flux projected onto the first and second excited state, respectively. The Fig. 6.3 displays the total and projected flux, similarly to the plots generated above.

Bibliography

- [1] J. Coonjobeeharry, K. E. Spinlove, C. Sanz Sanz, et al. „Mixed-quantum-classical or fully-quantized dynamics? A unified code to compare methods“. In: *Phil. Trans. Roy. Soc. A* 380 (2022), p. 20200386 (cit. on p. 22).
- [2] A. Ferretti, G. Granucci, A. Lami, M. Persico, and G. Villani. „Quantum mechanical and semiclassical dynamics at a conical intersection“. In: *J. Chem. Phys.* 104 (1996), pp. 5517–5527 (cit. on p. 22).
- [3] J. C. Tully. „Molecular dynamics with electronic transitions“. In: *J. Chem. Phys* 93 (1990), pp. 1061–1071 (cit. on p. 9).

Glossary

CAP

Complex absorbing potential used to absorb the wavepacket. 10

DD-vMCG

The direct dynamics version of variational multiconfigurational Gaussian method whereby potential energy surfaces are calculated on-the-fly (as required) utilising external Quantum Chemistry software. 2

DVR

Discrete variable representation. 16

GWP

Gaussian wavepacket. 21

HAMILTONIAN-SECTION

In an *operator file* this is the section where one defines the operator. 3

INIT_WF-SECTION

In an *input file* this is the section where the initial wavefunction is defined. 4, 7

input file

This is the input file for the calculations. Here we link to an operator file (where appropriate), the basis set, basis functions, the initial wavefunction and the run parameters of the calculation are set. 4

MCTDH

Multi-Configuration Time Dependant Hartree (method). 21

OP_DEFINE-SECTION

In an *operator file* this the section that defines the file - including the title. 3

operator file

This is where the operators for a calculation are defined. Here the operator file is defined with sections for the Hamiltonian and parameters. 3

OPERATOR-SECTION

In an *input file* this is the section that defines the *operator file* being used. 4

PARAMETER-SECTION

In an *operator file* this is the section where one defines the parameters of the operator. 4

PRIMITIVE-BASIS-SECTION

In an *input file* this is the section where the “primitive basis set”, or the grid, is defined. 4, 21

adpop

Results/Analysis: Calculates the diabatic and adiabatic populations of the electronic states. The calculation of one and two dimensional adiabatic densities is also possible.. 17

analysis

Results/Analysis: A driver for calling the appropriate analysis programs. 20

autospec

Results/Analysis: Computation of a spectrum by Fourier-transforming the autocorrelation function. 19

gwptraj

Results/Analysis: Plots the centres of GWP basis functions in vMCG (and G-MCTDH) calculations.. 22

-inter

Results/Analysis: “inter” is an abbreviation of “interactive” and hence is the command that calls for the interactive menu for results/analysis programs. 17, 18

rdauto

Results/Analysis: Reads the auto steps from file "auto" and enables plotting of the autocorrelation function.. 18

rdcheck

Results/Analysis: For plotting the trajectory of the centre of the wavepacket, **UPDATE THIS**. 6, 16

showsys

Results/Analysis: For plotting potential energy surfaces, with a numerical selection based interface. 5, 15, 17

statepop

Results/Analysis: Outputs the diabatic state populations.. 22

RUN-SECTION

In the *input file* this is the section which defines the calculation such as the propagation time.. 4

SPF

“Single-particle” basis function. 16, 21

SPF-BASIS-SECTION

In an *input file* this is the section where the number of “single-particle” basis functions will be defined. 4

Tully Models

A set of three model 1-dimensional potential energy surfaces set up by John C. Tully to benchmark scattering calculations including non-adiabatic effects (Tully JCP (90) **93**: 1061). 9

vMCG

The variational multiconfigurational Gaussian method where the MCTDH SPFs are replaced with variational Gaussians. 2, 21

