Quartic Force Field Training

Brent R. Westbrook

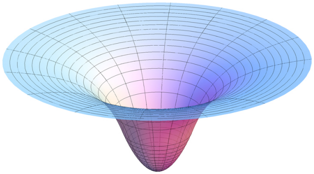
Fortenberry Patch

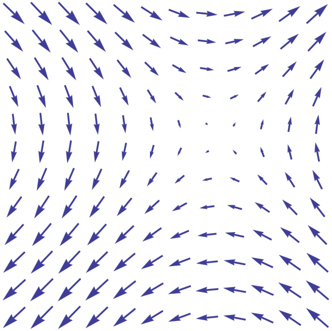
What is a quartic force field?

What is a ~~quartic~~ force field?

Definition

A vector field that describes a force acting on a particle at various positions in space


(a) A gravitational force field


(b) A vector field

What is a ~~quartic~~ force ~~field~~?

The obvious

$$F = ma$$

What is a ~~quartic~~ force ~~field~~?

The obvious

$$F = ma$$

The more useful

$$F = -\nabla V = -(\frac{\partial V}{\partial x}, \frac{\partial V}{\partial y}, \frac{\partial V}{\partial z})$$

How does this relate to us?

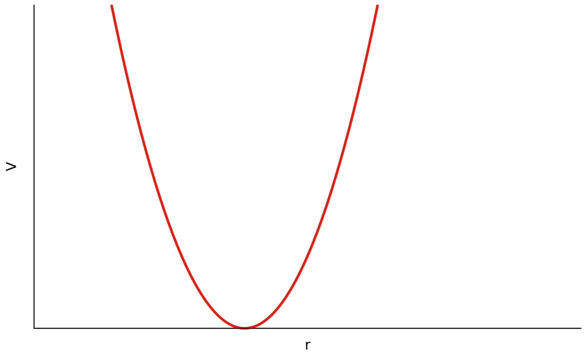
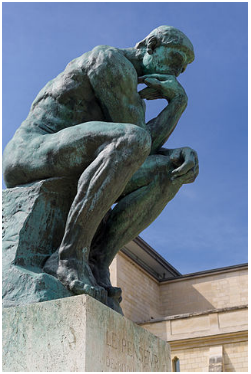


Figure: Potential vs displacement for a 1-D harmonic oscillator

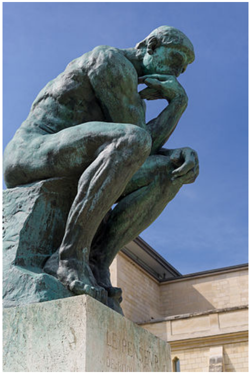
How does this relate to us?

Suppose we do not know the potential energy function for a molecule. How could we approximate it?



How does this relate to us?

Suppose we do not know the potential energy function for a molecule. How could we approximate it?



Use a Taylor series!

How does this relate to us?

Taylor series

Definition

An infinite sum of terms that are expressed in terms of the function's derivatives at a single point, *a*

$$f(x)_a \approx f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \ldots$$

Example

$$e^x_{a=0} \approx \frac{e^0}{0!} + \frac{e^0}{1!}(x-0) + \frac{e^0}{2!}(x-0)^2 + \ldots = 1 + x + \frac{x^2}{2} + \ldots$$

What is a quartic force field?

You've probably seen this in all of our papers:

Definition

A quartic force field is a fourth-order Taylor series expansion of the potential energy portion of the internuclear Watson Hamiltonian

And this in some of them:

Equation

$$V = \frac{1}{2}\sum_{ij} F_{ij}\Delta_i\Delta_j + \frac{1}{6}\sum_{ijk} F_{ijk}\Delta_i\Delta_j\Delta_k + \frac{1}{24}\sum_{ijkl} F_{ijkl}\Delta_i\Delta_j\Delta_k\Delta_l$$

Where did Taylor go?

Let's expand the potential energy for the 1-D harmonic oscillator, $V(r)$, about its equilibrium displacement r_{eq}, and let this be equal to zero.

$$V(r)_{r_{eq}} \approx \frac{1}{0!}V(r_{eq}) + \frac{1}{1!}\left(\frac{dV}{dr}\right)_{r_{eq}}(r - r_{eq}) + \frac{1}{2!}\left(\frac{d^2V}{dr^2}\right)_{r_{eq}}(r - r_{eq})^2 + \frac{1}{3!}\left(\frac{d^3V}{dr^3}\right)_{r_{eq}}(r - r_{eq})^3 + \frac{1}{4!}\left(\frac{d^4V}{dr^4}\right)_{r_{eq}}(r - r_{eq})^4$$

What do we know about the first two of these terms?

Where did Taylor go?

Let's expand the potential energy for the 1-D harmonic oscillator, $V(r)$, about its equilibrium displacement r_{eq}, and let this be equal to zero.

$$V(r)_{r_{eq}} \approx \cancel{\frac{1}{0!}V(r_{eq})} + \frac{1}{1!}\left(\frac{dV}{dr}\right)_{r_{eq}}(r-r_{eq}) + \frac{1}{2!}\left(\frac{d^2V}{dr^2}\right)_{r_{eq}}(r-r_{eq})^2 + \frac{1}{3!}\left(\frac{d^3V}{dr^3}\right)_{r_{eq}}(r-r_{eq})^3 + \frac{1}{4!}\left(\frac{d^4V}{dr^4}\right)_{r_{eq}}(r-r_{eq})^4$$

What do we know about the first two of these terms?

- The first cancels because we defined r_{eq} as our zero of potential

Where did Taylor go?

Let's expand the potential energy for the 1-D harmonic oscillator, $V(r)$, about its equilibrium displacement r_{eq}, and let this be equal to zero.

$$V(r)_{r_{eq}} \approx \cancel{\frac{1}{0!}V(r_{eq})} + \cancel{\frac{1}{1!}\left(\frac{dV}{dr}\right)_{r_{eq}}(r-r_{eq})} + \frac{1}{2!}\left(\frac{d^2V}{dr^2}\right)_{r_{eq}}(r-r_{eq})^2$$
$$+\frac{1}{3!}\left(\frac{d^3V}{dr^3}\right)_{r_{eq}}(r-r_{eq})^3 + \frac{1}{4!}\left(\frac{d^4V}{dr^4}\right)_{r_{eq}}(r-r_{eq})^4$$

What do we know about the first two of these terms?

- The first cancels because we defined r_{eq} as our zero of potential
- The second because the derivative at the minimum is zero

Where did Taylor go?

This leaves us with

$$V(r)_{r_{eq}} \approx \frac{1}{2!}\left(\frac{d^2V}{dr^2}\right)_{r_{eq}}(r-r_{eq})^2 + \frac{1}{3!}\left(\frac{d^3V}{dr^3}\right)_{r_{eq}}(r-r_{eq})^3 + \frac{1}{4!}\left(\frac{d^4V}{dr^4}\right)_{r_{eq}}(r-r_{eq})^4$$

which should be looking vaguely familiar

Where did Taylor go?

This leaves us with

$$V(r)_{r_{eq}} \approx \frac{1}{2!}\left(\frac{d^2V}{dr^2}\right)_{r_{eq}}(r-r_{eq})^2 + \frac{1}{3!}\left(\frac{d^3V}{dr^3}\right)_{r_{eq}}(r-r_{eq})^3 + \frac{1}{4!}\left(\frac{d^4V}{dr^4}\right)_{r_{eq}}(r-r_{eq})^4$$

which should be looking vaguely familiar

$$V = \frac{1}{2}\sum_{ij} F_{ij}\Delta_i\Delta_j + \frac{1}{6}\sum_{ijk} F_{ijk}\Delta_i\Delta_j\Delta_k + \frac{1}{24}\sum_{ijkl} F_{ijkl}\Delta_i\Delta_j\Delta_k\Delta_l$$

Where did Taylor go?

Let's zoom in on the first term. Looking familiar?

Comparison

$$V(r)_{r_{eq}} \approx \frac{1}{2!}\left(\frac{d^2V}{dr^2}\right)_{r_{eq}}(r - r_{eq})^2$$

$$V = \frac{1}{2}\sum_{ij} F_{ij}\Delta_i\Delta_j$$

Where did Taylor go?

Let's zoom in on the first term. Looking familiar?

Comparison

$$V(r)_{r_{eq}} \approx \frac{1}{2!}\left(\frac{d^2V}{dr^2}\right)_{r_{eq}}(r - r_{eq})^2$$

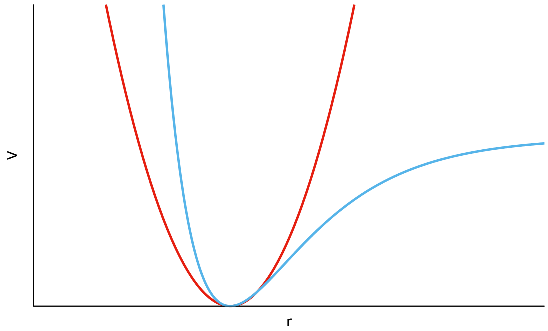
$$V = \frac{1}{2}\sum_{ij} F_{ij}\Delta_i\Delta_j$$

$$V = \frac{1}{2}kx^2$$

The first term is just the harmonic oscillator potential energy, accounting for the fact a second derivative can be two steps in one direction (like x^2), or one step in different directions (like $\Delta_i\Delta_j$)

What is a quartic ~~force field~~?

Quartic just refers to the fourth-degree part of the Taylor series



The quadratic or harmonic terms give us the red curve, while the third- and fourth-order terms treat the anharmonicity, giving something like the blue curve.

How do we make one?

Let’s go back to our full quartic equation

$$V = \frac{1}{2}\sum_{ij} F_{ij}\Delta_i\Delta_j + \frac{1}{6}\sum_{ijk} F_{ijk}\Delta_i\Delta_j\Delta_k + \frac{1}{24}\sum_{ijkl} F_{ijkl}\Delta_i\Delta_j\Delta_k\Delta_l$$

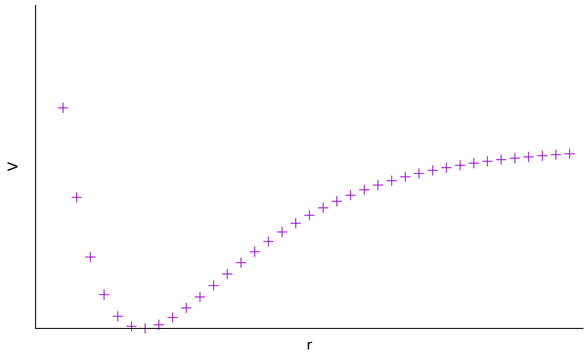
Of the terms here, we can “easily” get:

- V $\rightarrow$ comes out of Molpro
- Δ $\rightarrow$ can be anything we want, typically 0.005 Å or radians

By the way, why the summations?

How do we make one?

If we were to plot a 1-D case of taking steps (Δ) and computing the energy with Molpro, we would get something like this



But probably not this smooth

How do we make one?

To go from the scatterplot to a potential function, we have to

- Fit a function to our points, linear regression

How do we make one?

To go from the scatterplot to a potential function, we have to

- Fit a function to our points, linear regression
- Use that function to calculate the force constants $(F_{i,j,k,l})$

Recall

$$F = -\nabla V$$

How do we make one?

To go from the scatterplot to a potential function, we have to

- Fit a function to our points, linear regression
- Use that function to calculate the force constants ($F_{i,j,k,l}$)

Recall

$$F = -\nabla V$$

- Convert the force constants into frequencies

Harmonic oscillator again

$$\omega = \sqrt{\frac{k}{m}}$$

How do we make one?

"Fortunately" we have ancient programs that handle these steps for us

- Anpass - fits the function to our points and generates the force constants
- Spectro - determines the frequencies from the force constants
- Intder - the glue that holds all of the steps together

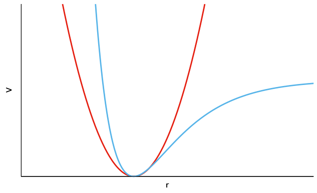
THIS IS HOW WE WOULD GET TO WORK IF WE BELIEVE THE SAYING
"IF IT AIN'T BROKE DON'T FIX IT"
imgflip.com

Getting started

Geometry optimization

Looking back at these, we see we need to start off at a good minimum

$$V(r)_{r_{eq}} \approx \cancel{\frac{1}{0!}V(r_{eq})} + \cancel{\frac{1}{1!}\left(\frac{dV}{dr}\right)_{r_{eq}}(r - r_{eq})} + \frac{1}{2!}\left(\frac{d^2V}{dr^2}\right)_{r_{eq}}(r - r_{eq})^2 + \frac{1}{3!}\left(\frac{d^3V}{dr^3}\right)_{r_{eq}}(r - r_{eq})^3 + \frac{1}{4!}\left(\frac{d^4V}{dr^4}\right)_{r_{eq}}(r - r_{eq})^4$$

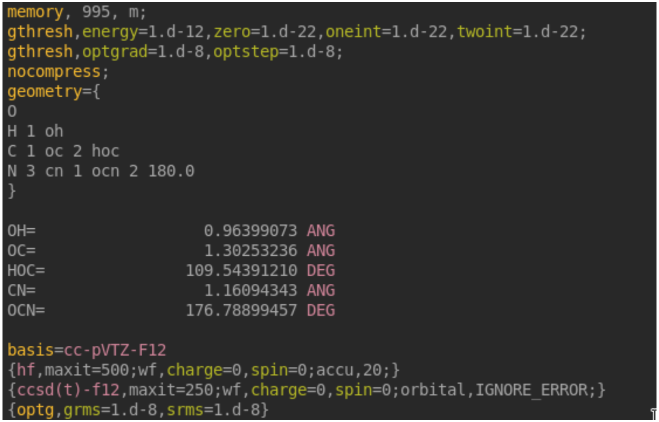


Cancelling the second term, in particular, requires that we are at the minimum energy structure

Geometry optimization

Assuming Molpro

CCSD(T)-F12/cc-pVTZ-F12 example, also shortened to F12-TZ (with my syntax highlighting "plugin")



```
memory, 995, m;
gthresh,energy=1.d-12,zero=1.d-22,oneint=1.d-22,twoint=1.d-22;
gthresh,optgrad=1.d-8,optstep=1.d-8;
nocompress;
geometry={
O
H 1 oh
C 1 oc 2 hoc
N 3 cn 1 ocn 2 180.0
}

OH=                  0.96399073 ANG
OC=                  1.30253236 ANG
HOC=               109.54391210 DEG
CN=                  1.16094343 ANG
OCN=               176.78899457 DEG

basis=cc-pVTZ-F12
{hf,maxit=500;wf,charge=0,spin=0;accu,20;}
{ccsd(t)-f12,maxit=250;wf,charge=0,spin=0;orbital,IGNORE_ERROR;}
{optg,grms=1.d-8,srms=1.d-8}
```

Project management (optimizing you)

What I do

Once that's running, you have a little while to plan your attack

Molecule	Opt	Freq	Pts	Freqs	Int	Notes
H_2O	x	R	R		x	Warning in opt.out
H_2S	R				x	
S_2O	R				x	

- Opt - Molpro geometry optimization
- Freq - Molpro harmonic frequencies
- Pts - QFF single-point energies
- Freqs - QFF anharmonic frequencies
- Int - intensities, usually done in Gaussian

Project management

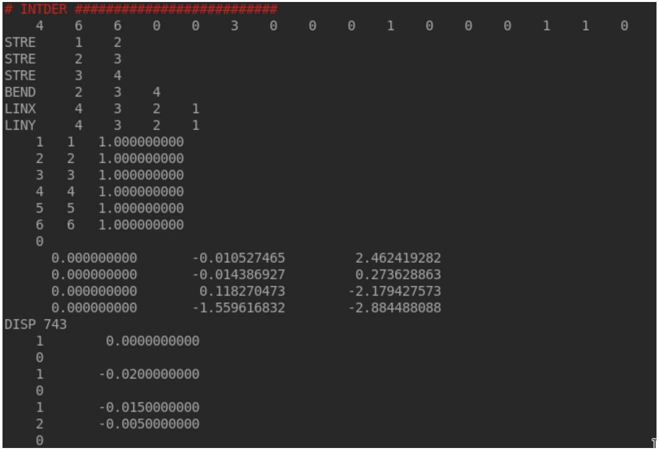
Dependencies/ordering

$$\begin{tikzcd}
 & \text{Freq} & & \text{Int} \\
\text{Opt} \arrow[ur, "1"] \arrow[dr, "2"'] & & & \\
 & \text{Pts} \arrow[rr, "3"] & & \text{Freqs}
\end{tikzcd}$$

1 can be done with a script called freqgen.sh, and we'll cover the others now

Generating points

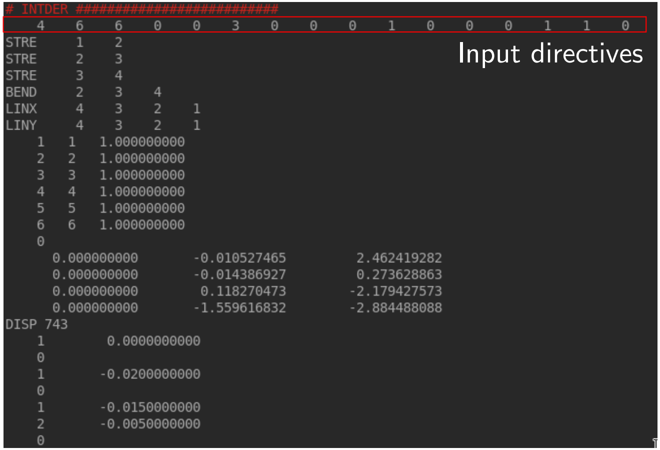
intder.in - anatomy



```
# INTDER ########################
    4    6    6    0    0    3    0    0    0    1    0    0    0    1    1    0
STRE      1    2
STRE      2    3
STRE      3    4
BEND      2    3    4
LINX      4    3    2    1
LINY      4    3    2    1
    1   1   1.000000000
    2   2   1.000000000
    3   3   1.000000000
    4   4   1.000000000
    5   5   1.000000000
    6   6   1.000000000
    0
      0.000000000       -0.010527465        2.462419282
      0.000000000       -0.014386927        0.273628863
      0.000000000        0.118270473       -2.179427573
      0.000000000       -1.559616832       -2.884488088
DISP 743
    1        0.0000000000
    0
    1       -0.0200000000
    0
    1       -0.0150000000
    2       -0.0050000000
    0
```

Generating points

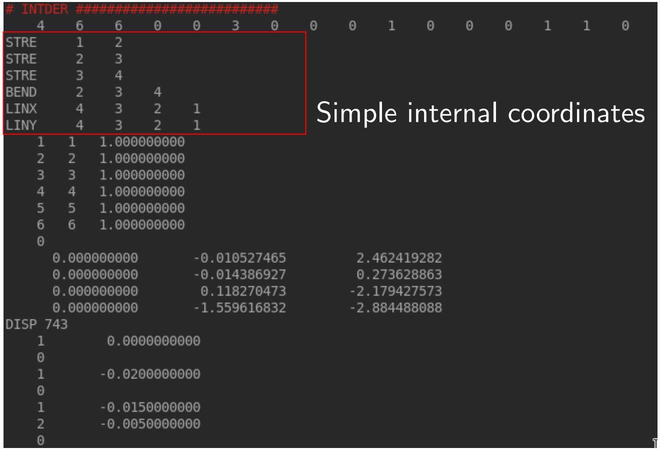
intder.in - anatomy



```
# INTDER ########################
    4    6    6    0    0    3    0    0    0    1    0    0    0    1    1    0
STRE     1    2
STRE     2    3
STRE     3    4
BEND     2    3    4
LINX     4    3    2    1
LINY     4    3    2    1
    1   1   1.000000000
    2   2   1.000000000
    3   3   1.000000000
    4   4   1.000000000
    5   5   1.000000000
    6   6   1.000000000
    0
      0.000000000        -0.010527465         2.462419282
      0.000000000        -0.014386927         0.273628863
      0.000000000         0.118270473        -2.179427573
      0.000000000        -1.559616832        -2.884488088
DISP 743
    1        0.0000000000
    0
    1       -0.0200000000
    0
    1       -0.0150000000
    2       -0.0050000000
    0
```

Generating points

intder.in - anatomy

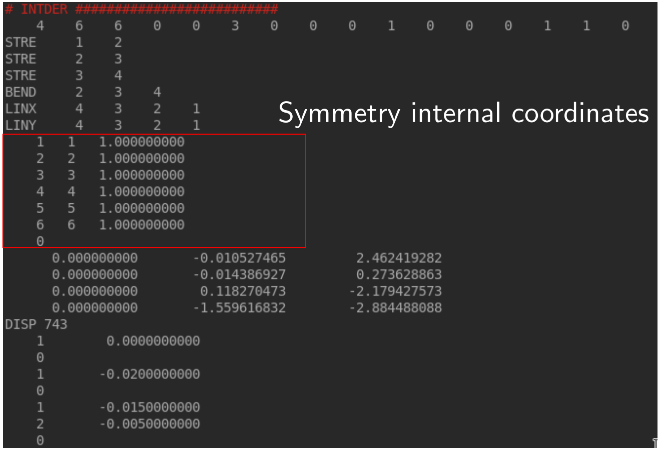


```
# INTDER #########################
    4    6    6    0    0    3    0    0    0    1    0    0    0    1    1    0
STRE      1    2
STRE      2    3
STRE      3    4
BEND      2    3    4
LINX      4    3    2    1
LINY      4    3    2    1
    1    1   1.000000000
    2    2   1.000000000
    3    3   1.000000000
    4    4   1.000000000
    5    5   1.000000000
    6    6   1.000000000
    0
      0.000000000       -0.010527465        2.462419282
      0.000000000       -0.014386927        0.273628863
      0.000000000        0.118270473       -2.179427573
      0.000000000       -1.559616832       -2.884488088
DISP 743
    1         0.0000000000
    0
    1        -0.0200000000
    0
    1        -0.0150000000
    2        -0.0050000000
    0
```

Simple internal coordinates

Generating points

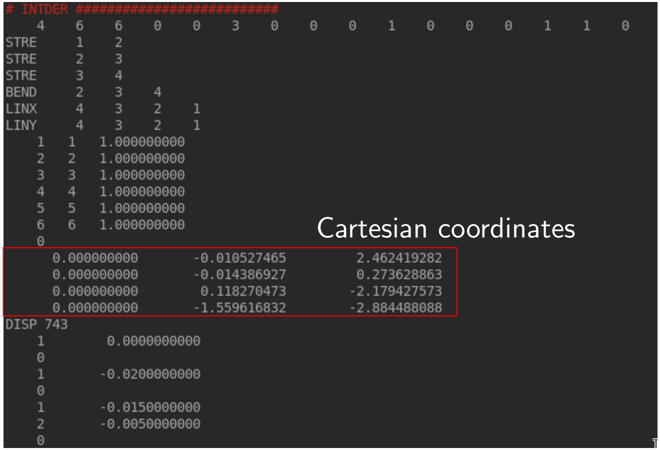
intder.in - anatomy



```
# INTDER ##########################
    4    6    6    0    0    3    0    0    0    1    0    0    0    1    1    0
STRE      1    2
STRE      2    3
STRE      3    4
BEND      2    3    4
LINX      4    3    2    1
LINY      4    3    2    1
    1   1   1.000000000
    2   2   1.000000000
    3   3   1.000000000
    4   4   1.000000000
    5   5   1.000000000
    6   6   1.000000000
    0
      0.000000000       -0.010527465        2.462419282
      0.000000000       -0.014386927        0.273628863
      0.000000000        0.118270473       -2.179427573
      0.000000000       -1.559616832       -2.884488088
DISP 743
    1         0.0000000000
    0
    1        -0.0200000000
    0
    1        -0.0150000000
    2        -0.0050000000
    0
```

Generating points

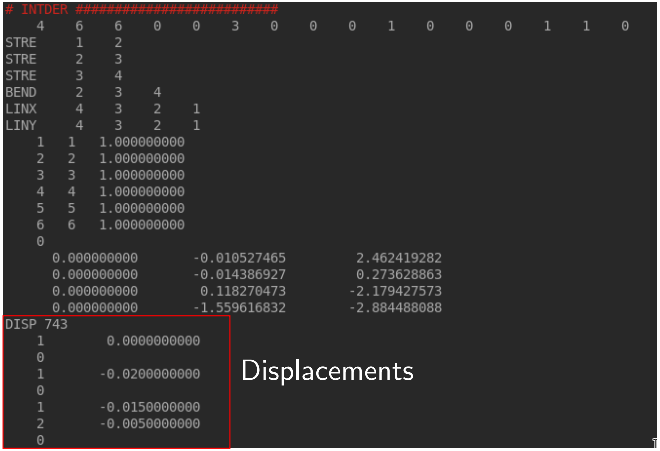
intder.in - anatomy



```
# INTDER #########################
    4    6    6    0    0    3    0    0    0    1    0    0    0    1    1    0
STRE      1    2
STRE      2    3
STRE      3    4
BEND      2    3    4
LINX      4    3    2    1
LINY      4    3    2    1
    1   1   1.000000000
    2   2   1.000000000
    3   3   1.000000000
    4   4   1.000000000
    5   5   1.000000000
    6   6   1.000000000
    0
      0.000000000        -0.010527465         2.462419282
      0.000000000        -0.014386927         0.273628863
      0.000000000         0.118270473        -2.179427573
      0.000000000        -1.559616832        -2.884488088
DISP 743
    1        0.0000000000
    0
    1       -0.0200000000
    0
    1       -0.0150000000
    2       -0.0050000000
    0
```

Generating points

intder.in - anatomy

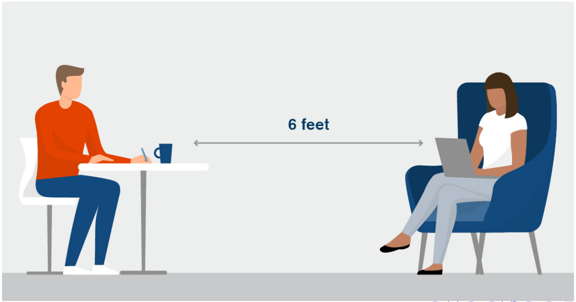


```
# INTDER ########################
    4    6    6    0    0    3    0    0    0    1    0    0    0    1    1    0
STRE      1    2
STRE      2    3
STRE      3    4
BEND      2    3    4
LINX      4    3    2    1
LINY      4    3    2    1
    1    1   1.000000000
    2    2   1.000000000
    3    3   1.000000000
    4    4   1.000000000
    5    5   1.000000000
    6    6   1.000000000
    0
      0.000000000       -0.010527465         2.462419282
      0.000000000       -0.014386927         0.273628863
      0.000000000        0.118270473        -2.179427573
      0.000000000       -1.559616832        -2.884488088
DISP 743
    1         0.0000000000
    0
    1        -0.0200000000
    0
    1        -0.0150000000
    2        -0.0050000000
    0
```

Generating points

intder.in - general tips

- Spacing matters! Everywhere
- Don't use tabs
- Don't have trailing whitespace
- Copy a similar line from a known file to compare
- Spacing matters



Generating points

intder.in - general tips

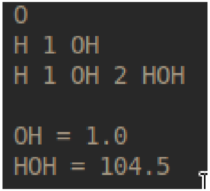
That looks confusing. What do I actually change?

- Only the Cartesians for now (I think)
- Directives are in the manual if you really need them
- Internal coordinates are our next topic

Internal coordinates

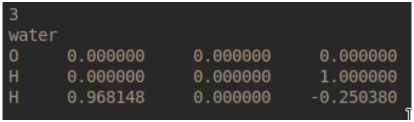
Definition

Coordinates with an internal reference, in contrast to Cartesian coordinates, for example, which reference an external set of axes



```
O
H 1 OH
H 1 OH 2 HOH

OH = 1.0
HOH = 104.5
```

Figure: Z-matrix: (hopefully) familiar internal coordinates



```
3
water
O      0.000000      0.000000      0.000000
H      0.000000      0.000000      1.000000
H      0.968148      0.000000     -0.250380
```

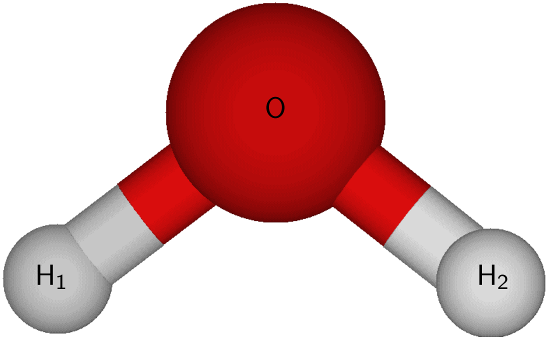
Figure: Cartesian coordinates

Internal coordinates

Simple internals

Definition

A single internal coordinate, like a bond stretch, bend, or torsion

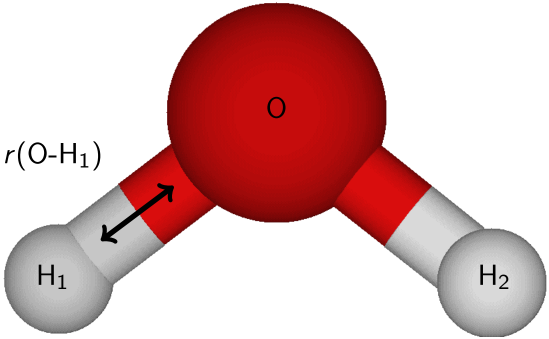


Internal coordinates

Simple internals

Definition

A single internal coordinate, like a bond stretch, bend, or torsion

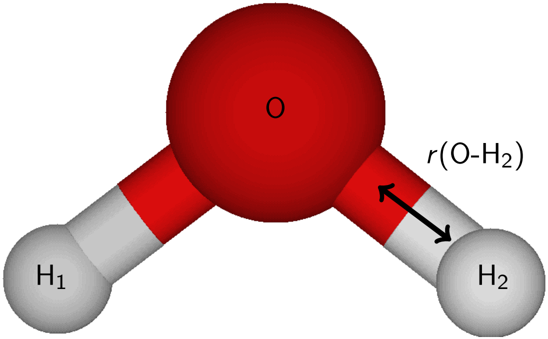


Internal coordinates

Simple internals

Definition

A single internal coordinate, like a bond stretch, bend, or torsion

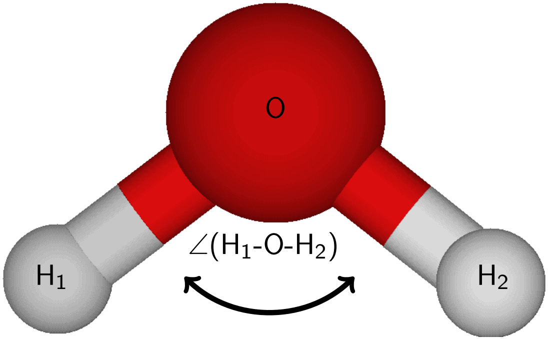


Internal coordinates

Simple internals

Definition

A single internal coordinate, like a bond stretch, bend, or torsion

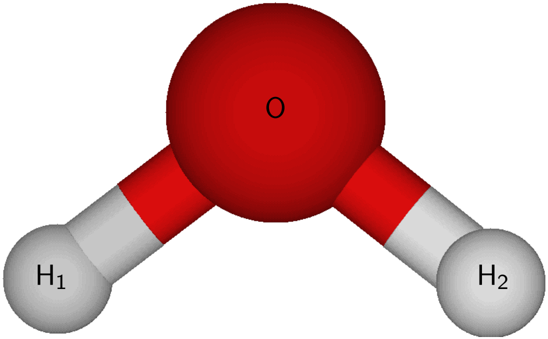


Internal coordinates

Symmetry internals

Definition

Linear combinations of simple internal coordinates



Internal coordinates

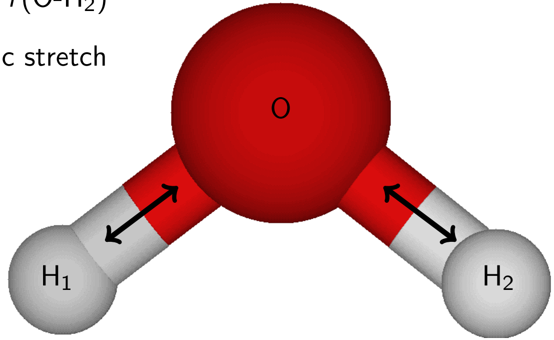
Symmetry internals

Definition

Linear combinations of simple internal coordinates

$r(\text{O-H}_1)+r(\text{O-H}_2)$

Symmetric stretch



Internal coordinates

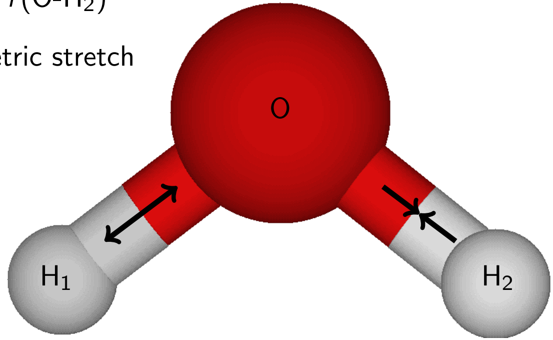
Symmetry internals

Definition

Linear combinations of simple internal coordinates

$r(\text{O-H}_1) - r(\text{O-H}_2)$

Anti-symmetric stretch



Internal coordinates

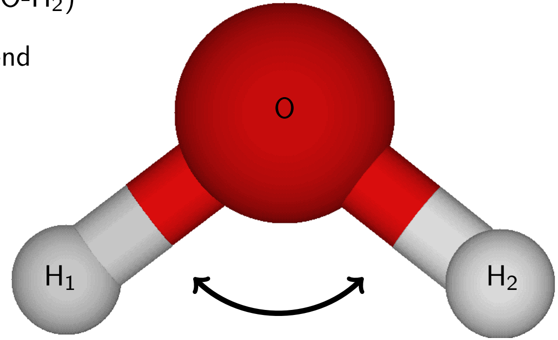
Symmetry internals

Definition

Linear combinations of simple internal coordinates

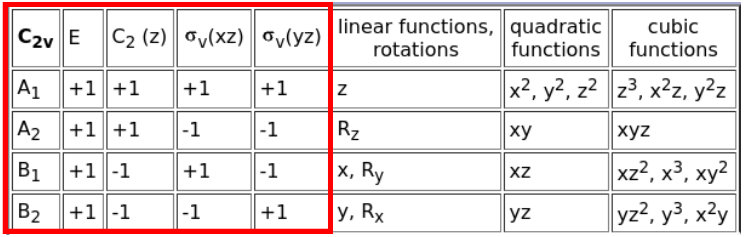
$\angle(H_1\text{-}O\text{-}H_2)$

Bend



Symmetry

We can characterize a molecule, and its vibrational modes, based on symmetry. Water has C_{2v} symmetry, which is a pretty common one in our research.



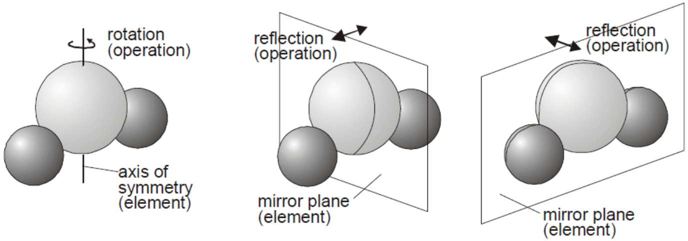
C_{2v}	E	C_2 (z)	$\sigma_v(xz)$	$\sigma_v(yz)$	linear functions, rotations	quadratic functions	cubic functions
A_1	+1	+1	+1	+1	z	x^2, y^2, z^2	z^3, x^2z, y^2z
A_2	+1	+1	-1	-1	R_z	xy	xyz
B_1	+1	-1	+1	-1	x, R_y	xz	xz^2, x^3, xy^2
B_2	+1	-1	-1	+1	y, R_x	yz	yz^2, y^3, x^2y

Figure: Symmetry or character information is found in character tables

Symmetry

Symmetry operations

- Identity - E
- Rotation - C_n, at left
- Reflection - σ, at right
- Inversion - i, not shown
- Improper rotation - S_n, not shown
  - $S_1 = \sigma$
  - $S_2 = i$

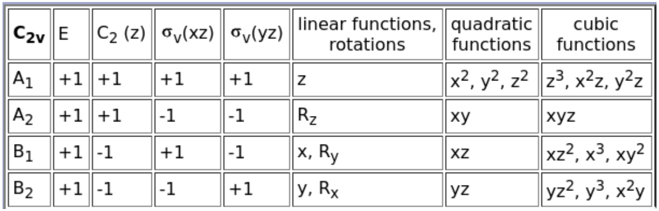


Symmetry

- Usually rotations and reflections are enough to assign a symmetry, or when the symmetry is C_{2v}, that's all there is anyway.
- How do we do it?

Symmetry

- Usually rotations and reflections are enough to assign a symmetry, or when the symmetry is C_{2v}, that's all there is anyway.
- How do we do it? Back to the character table

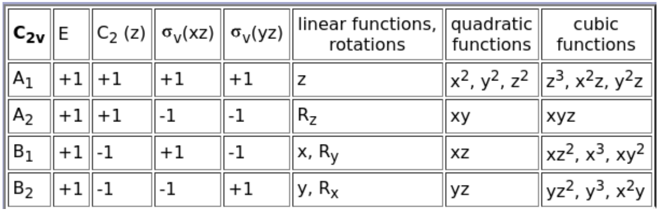


C_{2v}	E	C_2 (z)	$\sigma_v(xz)$	$\sigma_v(yz)$	linear functions, rotations	quadratic functions	cubic functions
A_1	+1	+1	+1	+1	z	x^2, y^2, z^2	z^3, x^2z, y^2z
A_2	+1	+1	-1	-1	R_z	xy	xyz
B_1	+1	-1	+1	-1	x, R_y	xz	xz^2, x^3, xy^2
B_2	+1	-1	-1	+1	y, R_x	yz	yz^2, y^3, x^2y

Symmetry

Determining symmetry

- Perform a symmetry operation
  - If the molecule looks the same, assign a 1
  - If the molecule looks the "opposite", assign a -1
- Match your ± 1 values to rows of the character table

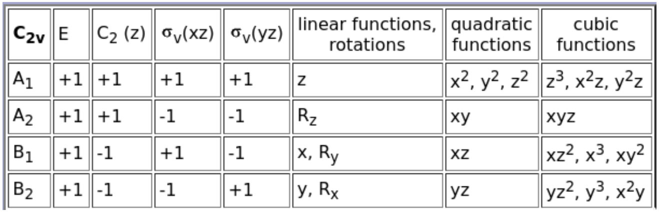


C_{2v}	E	C_2 (z)	σ_v(xz)	σ_v(yz)	linear functions, rotations	quadratic functions	cubic functions
A_1	+1	+1	+1	+1	z	x^2, y^2, z^2	z^3, x^2z, y^2z
A_2	+1	+1	-1	-1	R_z	xy	xyz
B_1	+1	-1	+1	-1	x, R_y	xz	xz^2, x^3, xy^2
B_2	+1	-1	-1	+1	y, R_x	yz	yz^2, y^3, x^2y

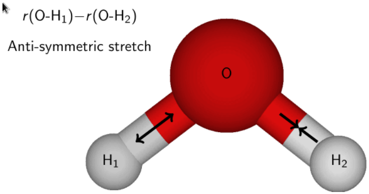
Symmetry

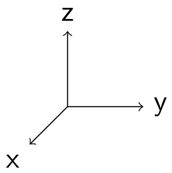
Determining symmetry

What is the symmetry of the anti-symmetric stretch in water? Let the main axis of rotation be the z-axis



$\mathbf{C_{2v}}$	E	C_2 (z)	$\sigma_v(xz)$	$\sigma_v(yz)$	linear functions, rotations	quadratic functions	cubic functions
A_1	+1	+1	+1	+1	z	x^2, y^2, z^2	z^3, x^2z, y^2z
A_2	+1	+1	-1	-1	R_z	xy	xyz
B_1	+1	-1	+1	-1	x, R_y	xz	xz^2, x^3, xy^2
B_2	+1	-1	-1	+1	y, R_x	yz	yz^2, y^3, x^2y





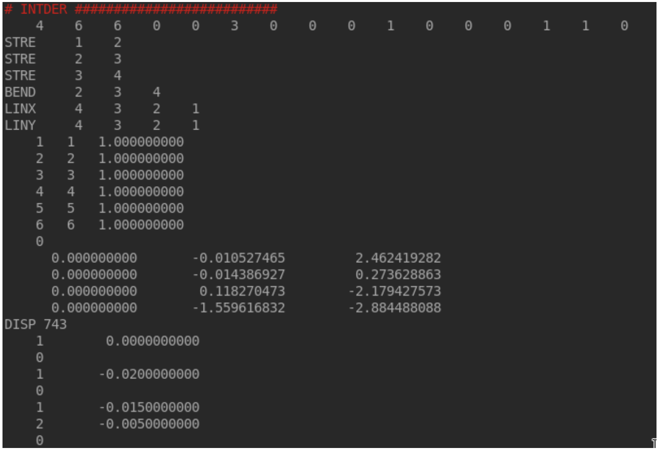
Symmetry

Why do we need it?

- Limits the number of displacements we have to calculate
- Part of what determines the IR activity of a mode
- It goes in your computational details

Generating points

intder.in - water

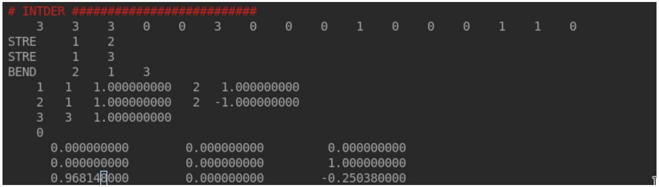


```
# INTDER ##########################
    4    6    6    0    0    3    0    0    0    1    0    0    0    1    1    0
STRE     1    2
STRE     2    3
STRE     3    4
BEND     2    3    4
LINX     4    3    2    1
LINY     4    3    2    1
    1   1   1.000000000
    2   2   1.000000000
    3   3   1.000000000
    4   4   1.000000000
    5   5   1.000000000
    6   6   1.000000000
    0
      0.000000000       -0.010527465        2.462419282
      0.000000000       -0.014386927        0.273628863
      0.000000000        0.118270473       -2.179427573
      0.000000000       -1.559616832       -2.884488088
DISP 743
    1        0.0000000000
    0
    1       -0.0200000000
    0
    1       -0.0150000000
    2       -0.0050000000
    0
```

Figure: Not water

Generating points

intder.in - water



```
# INTDER ##########################
    3    3    3    0    0    3    0    0    0    1    0    0    0    1    1    0
STRE      1    2
STRE      1    3
BEND      2    1    3
    1   1   1.000000000   2   1.000000000
    2   1   1.000000000   2  -1.000000000
    3   3   1.000000000
    0
      0.000000000        0.000000000         0.000000000
      0.000000000        0.000000000         1.000000000
      0.968148000        0.000000000        -0.250380000
```

Figure: Water

Generating points

Procedure

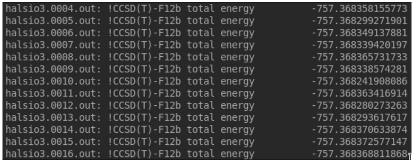
- Copy (yank) optimized geometry (in Bohr) from the Molpro log file
- Paste it into the intder.in file and make sure alignment is perfect
- Run $ intder < intder.in > intder.out
  or something similar
- Use pts-gen.py to generate the single-point energy calculations from intder output
  - $ pts-gen.py h2o O H H
    in our water example
- $ cd inp && sh submit
  to submit the calculations

Gathering output

- Grab all the energies from output files
  - $ grep “CCSD(T)-F12b” *.out > energy.dat
    or use check.sh
- Check that the number of energies is the same as the number of displacements
  - handy command for this is $ wc -l filename
    “word count -lines”
  - $ comp.py h2o
    to print missing ones to terminal
  - $ comp.py h2o | sh
    to submit them right away
- Remember to rerun grep or check.sh after these run!

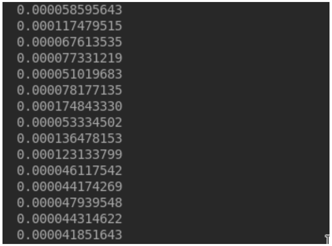
Preparing output

To minimize numerical noise, we want to work with relative energies. That is, we need to subtract the minimum energy from all of the other energies.



```
halsio3.0004.out: !CCSD(T)-F12b total energy          -757.368358155773
halsio3.0005.out: !CCSD(T)-F12b total energy          -757.368299271901
halsio3.0006.out: !CCSD(T)-F12b total energy          -757.368349137881
halsio3.0007.out: !CCSD(T)-F12b total energy          -757.368339420197
halsio3.0008.out: !CCSD(T)-F12b total energy          -757.368365731733
halsio3.0009.out: !CCSD(T)-F12b total energy          -757.368338574281
halsio3.0010.out: !CCSD(T)-F12b total energy          -757.368241908086
halsio3.0011.out: !CCSD(T)-F12b total energy          -757.368363416914
halsio3.0012.out: !CCSD(T)-F12b total energy          -757.368280273263
halsio3.0013.out: !CCSD(T)-F12b total energy          -757.368293617617
halsio3.0014.out: !CCSD(T)-F12b total energy          -757.368370633874
halsio3.0015.out: !CCSD(T)-F12b total energy          -757.368372577147
halsio3.0016.out: !CCSD(T)-F12b total energy          -757.368368811868
```

Figure: Want to go from this...



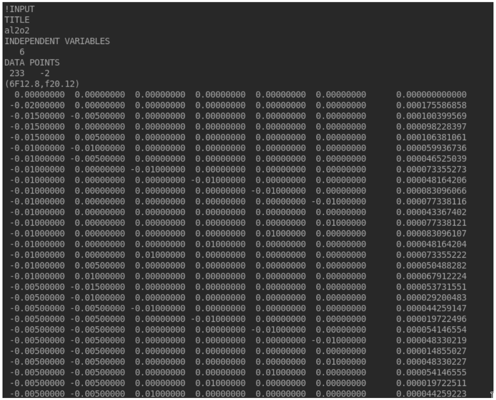
```
0.000058595643
0.000117479515
0.000067613535
0.000077331219
0.000051019683
0.000078177135
0.000174843330
0.000053334502
0.000136478153
0.000123133799
0.000046117542
0.000044174269
0.000047939548
0.000044314622
0.000041851643
```

Figure: to this.

Use awk to do this, see relmaker.sh

Anpass

With the relative energies in hand, we are ready to fit them to a function

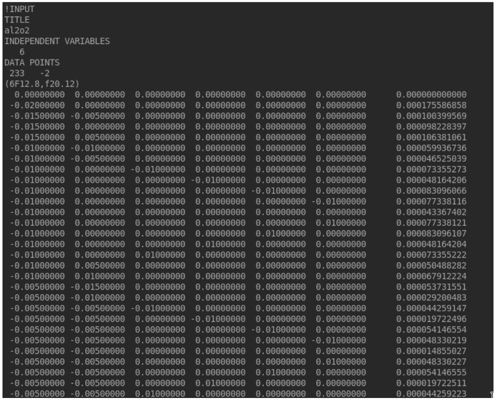


```
!INPUT
TITLE
al2o2
INDEPENDENT VARIABLES
   6
DATA POINTS
 233   -2
(6F12.8,f20.12)
  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000000000000
 -0.02000000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000175586858
 -0.01500000 -0.00500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000100399569
 -0.01500000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000098228397
 -0.01500000  0.00500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000106381061
 -0.01000000 -0.01000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000059936736
 -0.01000000 -0.00500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000046525039
 -0.01000000  0.00000000 -0.01000000  0.00000000  0.00000000  0.00000000      0.000073355273
 -0.01000000  0.00000000  0.00000000 -0.01000000  0.00000000  0.00000000      0.000048164206
 -0.01000000  0.00000000  0.00000000  0.00000000 -0.01000000  0.00000000      0.000083096066
 -0.01000000  0.00000000  0.00000000  0.00000000  0.00000000 -0.01000000      0.000077338116
 -0.01000000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000043367402
 -0.01000000  0.00000000  0.00000000  0.00000000  0.00000000  0.01000000      0.000077338121
 -0.01000000  0.00000000  0.00000000  0.00000000  0.01000000  0.00000000      0.000083096107
 -0.01000000  0.00000000  0.00000000  0.01000000  0.00000000  0.00000000      0.000048164204
 -0.01000000  0.00000000  0.01000000  0.00000000  0.00000000  0.00000000      0.000073355222
 -0.01000000  0.00500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000050488282
 -0.01000000  0.01000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000067912224
 -0.00500000 -0.01500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000053731551
 -0.00500000 -0.01000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000029200483
 -0.00500000 -0.00500000 -0.01000000  0.00000000  0.00000000  0.00000000      0.000044259147
 -0.00500000 -0.00500000  0.00000000 -0.01000000  0.00000000  0.00000000      0.000019722496
 -0.00500000 -0.00500000  0.00000000  0.00000000 -0.01000000  0.00000000      0.000054146554
 -0.00500000 -0.00500000  0.00000000  0.00000000  0.00000000 -0.01000000      0.000048330219
 -0.00500000 -0.00500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000014855027
 -0.00500000 -0.00500000  0.00000000  0.00000000  0.00000000  0.01000000      0.000048330227
 -0.00500000 -0.00500000  0.00000000  0.00000000  0.01000000  0.00000000      0.000054146555
 -0.00500000 -0.00500000  0.00000000  0.01000000  0.00000000  0.00000000      0.000019722511
 -0.00500000 -0.00500000  0.01000000  0.00000000  0.00000000  0.00000000      0.000044259223
```

Figure: Example anpass input file

Anpass

Formatting is again very important!

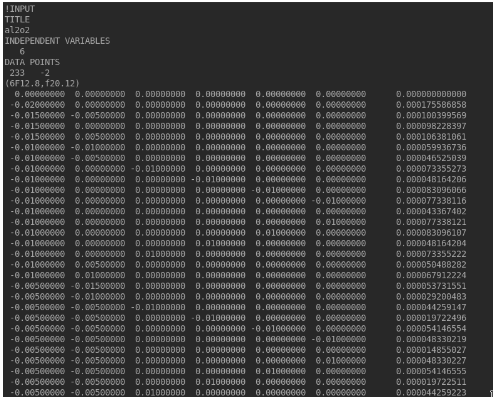


```
!INPUT
TITLE
al2o2
INDEPENDENT VARIABLES
   6
DATA POINTS
 233   -2
(6F12.8,f20.12)
  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000000000000
 -0.02000000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000175586858
 -0.01500000 -0.00500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000100399569
 -0.01500000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000098228397
 -0.01500000  0.00500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000106381061
 -0.01000000 -0.01000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000059936736
 -0.01000000 -0.00500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000046525039
 -0.01000000  0.00000000 -0.01000000  0.00000000  0.00000000  0.00000000      0.000073355273
 -0.01000000  0.00000000  0.00000000 -0.01000000  0.00000000  0.00000000      0.000048164206
 -0.01000000  0.00000000  0.00000000  0.00000000 -0.01000000  0.00000000      0.000083096066
 -0.01000000  0.00000000  0.00000000  0.00000000  0.00000000 -0.01000000      0.000077338116
 -0.01000000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000043367402
 -0.01000000  0.00000000  0.00000000  0.00000000  0.00000000  0.01000000      0.000077338121
 -0.01000000  0.00000000  0.00000000  0.00000000  0.01000000  0.00000000      0.000083096107
 -0.01000000  0.00000000  0.00000000  0.01000000  0.00000000  0.00000000      0.000048164204
 -0.01000000  0.00000000  0.01000000  0.00000000  0.00000000  0.00000000      0.000073355222
 -0.01000000  0.00500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000050488282
 -0.01000000  0.01000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000067912224
 -0.00500000 -0.01500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000053731551
 -0.00500000 -0.01000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000029200483
 -0.00500000 -0.00500000 -0.01000000  0.00000000  0.00000000  0.00000000      0.000044259147
 -0.00500000 -0.00500000  0.00000000 -0.01000000  0.00000000  0.00000000      0.000019722496
 -0.00500000 -0.00500000  0.00000000  0.00000000 -0.01000000  0.00000000      0.000054146554
 -0.00500000 -0.00500000  0.00000000  0.00000000  0.00000000 -0.01000000      0.000048330219
 -0.00500000 -0.00500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000014855027
 -0.00500000 -0.00500000  0.00000000  0.00000000  0.00000000  0.01000000      0.000048330227
 -0.00500000 -0.00500000  0.00000000  0.00000000  0.01000000  0.00000000      0.000054146555
 -0.00500000 -0.00500000  0.00000000  0.01000000  0.00000000  0.00000000      0.000019722511
 -0.00500000 -0.00500000  0.01000000  0.00000000  0.00000000  0.00000000      0.000044259223
```

Figure: Example anpass input file

Anpass

The first six columns in this case are the displacements. The final column is the relative energies, which you need to replace with your data

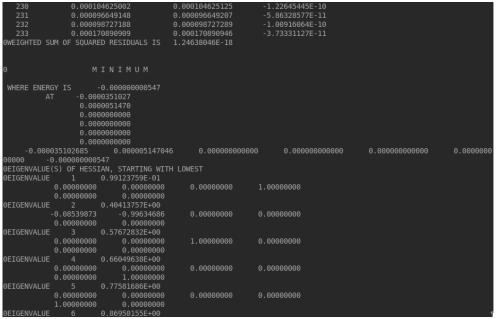


```
!INPUT
TITLE
al2o2
INDEPENDENT VARIABLES
   6
DATA POINTS
 233   -2
(6F12.8,f20.12)
  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000000000000
 -0.02000000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000175586858
 -0.01500000 -0.00500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000100399569
 -0.01500000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000098228397
 -0.01500000  0.00500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000106381061
 -0.01000000 -0.01000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000059936736
 -0.01000000 -0.00500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000046525039
 -0.01000000  0.00000000 -0.01000000  0.00000000  0.00000000  0.00000000      0.000073355273
 -0.01000000  0.00000000  0.00000000 -0.01000000  0.00000000  0.00000000      0.000048164206
 -0.01000000  0.00000000  0.00000000  0.00000000 -0.01000000  0.00000000      0.000083096066
 -0.01000000  0.00000000  0.00000000  0.00000000  0.00000000 -0.01000000      0.000077338116
 -0.01000000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000043367402
 -0.01000000  0.00000000  0.00000000  0.00000000  0.00000000  0.01000000      0.000077338121
 -0.01000000  0.00000000  0.00000000  0.00000000  0.01000000  0.00000000      0.000083096107
 -0.01000000  0.00000000  0.00000000  0.01000000  0.00000000  0.00000000      0.000048164204
 -0.01000000  0.00000000  0.01000000  0.00000000  0.00000000  0.00000000      0.000073355222
 -0.01000000  0.00500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000050488282
 -0.01000000  0.01000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000067912224
 -0.00500000 -0.01500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000053731551
 -0.00500000 -0.01000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000029200483
 -0.00500000 -0.00500000 -0.01000000  0.00000000  0.00000000  0.00000000      0.000044259147
 -0.00500000 -0.00500000  0.00000000 -0.01000000  0.00000000  0.00000000      0.000019722496
 -0.00500000 -0.00500000  0.00000000  0.00000000 -0.01000000  0.00000000      0.000054146554
 -0.00500000 -0.00500000  0.00000000  0.00000000  0.00000000 -0.01000000      0.000048330219
 -0.00500000 -0.00500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000014855027
 -0.00500000 -0.00500000  0.00000000  0.00000000  0.00000000  0.01000000      0.000048330227
 -0.00500000 -0.00500000  0.00000000  0.00000000  0.01000000  0.00000000      0.000054146555
 -0.00500000 -0.00500000  0.00000000  0.01000000  0.00000000  0.00000000      0.000019722511
 -0.00500000 -0.00500000  0.01000000  0.00000000  0.00000000  0.00000000      0.000044259223
```

Anpass

Once the input file is set up:

- Run anpass with \$ anpass < anpass1.in > anpass1.out
  - Notice the 1 here, this is the first of two Anpass runs
- Go to the bottom of the first output file and make sure the sum of squared residuals is less than about 10^{-16}

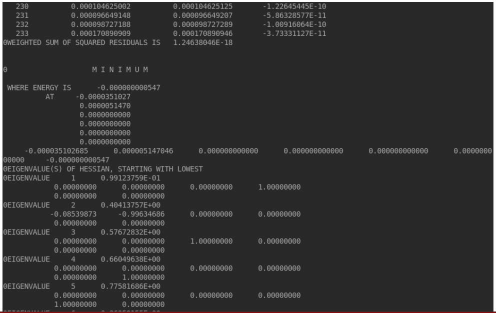


```
  230         0.000104625002         0.000104625125       -1.22645445E-10
  231         0.000096649148         0.000096649207       -5.86328577E-11
  232         0.000098727188         0.000098727289       -1.00916064E-10
  233         0.000170890909         0.000170890946       -3.73331127E-11
0WEIGHTED SUM OF SQUARED RESIDUALS IS   1.24638046E-18


0                    M I N I M U M

 WHERE ENERGY IS      -0.000000000547
         AT     -0.0000351027
                 0.0000051470
                 0.0000000000
                 0.0000000000
                 0.0000000000
                 0.0000000000
     -0.000035102685      0.000005147046      0.000000000000      0.000000000000      0.000000000000      0.0000000
00000     -0.000000000547
0EIGENVALUE(S) OF HESSIAN, STARTING WITH LOWEST
0EIGENVALUE     1      0.99123759E-01
            0.00000000      0.00000000      0.00000000      1.00000000
            0.00000000      0.00000000
0EIGENVALUE     2      0.40413757E+00
           -0.08539873     -0.99634686      0.00000000      0.00000000
            0.00000000      0.00000000
0EIGENVALUE     3      0.57672832E+00
            0.00000000      0.00000000      1.00000000      0.00000000
            0.00000000      0.00000000
0EIGENVALUE     4      0.66049638E+00
            0.00000000      0.00000000      0.00000000      0.00000000
            0.00000000      1.00000000
0EIGENVALUE     5      0.77581686E+00
            0.00000000      0.00000000      0.00000000      0.00000000
            1.00000000      0.00000000
0EIGENVALUE     6      0.86950155E+00
```

Anpass

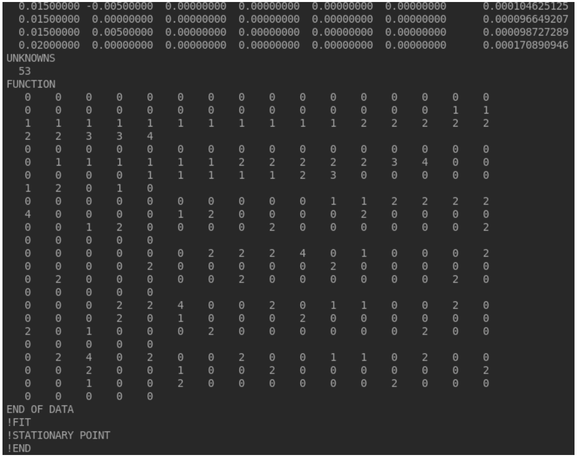
After this:

- Copy anpass1.in to anpass2.in
- Copy the “long line” from anpass1.out to the bottom of anpass1.in, above END OF DATA
- Cut the !STATIONARY POINT line and paste it above the “long line”



```
  230         0.000104625002         0.000104625125       -1.22645445E-10
  231         0.000096649148         0.000096649207       -5.86328577E-11
  232         0.000098727188         0.000098727289       -1.00916064E-10
  233         0.000170890909         0.000170890946       -3.73331127E-11
0WEIGHTED SUM OF SQUARED RESIDUALS IS   1.24638046E-18


0                    M I N I M U M

 WHERE ENERGY IS      -0.000000000547
          AT     -0.0000351027
                  0.0000051470
                  0.0000000000
                  0.0000000000
                  0.0000000000
                  0.0000000000
     -0.000035102685      0.000005147046      0.000000000000      0.000000000000      0.000000000000      0.0000000
00000     -0.000000000547
0EIGENVALUE(S) OF HESSIAN, STARTING WITH LOWEST
0EIGENVALUE     1      0.99123759E-01
            0.00000000      0.00000000      0.00000000      1.00000000
            0.00000000      0.00000000
0EIGENVALUE     2      0.40413757E+00
           -0.08539873     -0.99634686      0.00000000      0.00000000
            0.00000000      0.00000000
0EIGENVALUE     3      0.57672832E+00
            0.00000000      0.00000000      1.00000000      0.00000000
            0.00000000      0.00000000
0EIGENVALUE     4      0.66049638E+00
            0.00000000      0.00000000      0.00000000      0.00000000
            0.00000000      1.00000000
0EIGENVALUE     5      0.77581686E+00
            0.00000000      0.00000000      0.00000000      0.00000000
            1.00000000      0.00000000
```

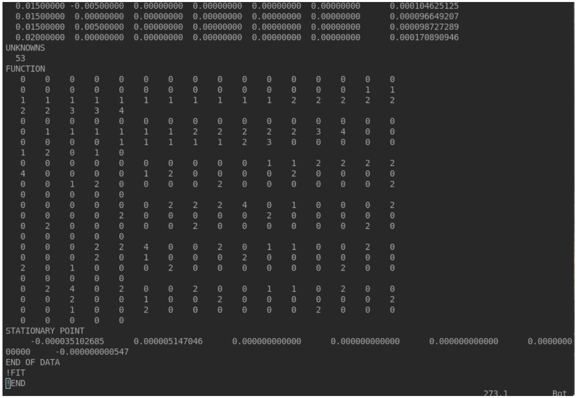
Anpass



```
  0.01500000 -0.00500000  0.00000000  0.00000000  0.00000000  0.00000000       0.000104625125
  0.01500000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000       0.000096649207
  0.01500000  0.00500000  0.00000000  0.00000000  0.00000000  0.00000000       0.000098727289
  0.02000000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000       0.000170890946
UNKNOWNS
  53
FUNCTION
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    1
   1    1    1    1    1    1    1    1    1    1    1    2    2    2    2    2
   2    2    3    3    4
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    1    1    1    1    1    1    2    2    2    2    2    3    4    0    0
   0    0    0    0    1    1    1    1    1    2    3    0    0    0    0    0
   1    2    0    1    0
   0    0    0    0    0    0    0    0    0    0    1    1    2    2    2    2
   4    0    0    0    0    1    2    0    0    0    0    2    0    0    0    0
   0    0    1    2    0    0    0    0    2    0    0    0    0    0    0    2
   0    0    0    0    0
   0    0    0    0    0    0    2    2    2    4    0    1    0    0    0    2
   0    0    0    0    2    0    0    0    0    0    2    0    0    0    0    0
   0    2    0    0    0    0    0    2    0    0    0    0    0    0    2    0
   0    0    0    0    0
   0    0    0    2    2    4    0    0    2    0    1    1    0    0    2    0
   0    0    0    2    0    1    0    0    0    2    0    0    0    0    0    0
   2    0    1    0    0    0    2    0    0    0    0    0    0    2    0    0
   0    0    0    0    0
   0    2    4    0    2    0    0    2    0    0    1    1    0    2    0    0
   0    0    2    0    0    1    0    0    2    0    0    0    0    0    0    2
   0    0    1    0    0    2    0    0    0    0    0    0    2    0    0    0
   0    0    0    0    0
END OF DATA
!FIT
!STATIONARY POINT
!END
```

Figure: anpass1.in bottom

Anpass



```
  0.01500000 -0.00500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000104625125
  0.01500000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000096649207
  0.01500000  0.00500000  0.00000000  0.00000000  0.00000000  0.00000000      0.000098727289
  0.02000000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000      0.000170890946
UNKNOWNS
  53
FUNCTION
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    1
   1    1    1    1    1    1    1    1    1    1    1    2    2    2    2    2
   2    2    3    3    4
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    1    1    1    1    1    1    2    2    2    2    2    3    4    0    0
   0    0    0    0    1    1    1    1    1    2    3    0    0    0    0    0
   1    2    0    1    0
   0    0    0    0    0    0    0    0    0    0    1    1    2    2    2    2
   4    0    0    0    0    1    2    0    0    0    0    2    0    0    0    0
   0    0    1    2    0    0    0    0    2    0    0    0    0    0    0    2
   0    0    0    0    0
   0    0    0    0    0    0    2    2    2    4    0    1    0    0    0    2
   0    0    0    0    2    0    0    0    0    0    2    0    0    0    0    0
   0    2    0    0    0    0    0    2    0    0    0    0    0    0    2    0
   0    0    0    0    0
   0    0    0    2    2    4    0    0    2    0    1    1    0    0    2    0
   0    0    0    2    0    1    0    0    0    2    0    0    0    0    0    0
   2    0    1    0    0    0    2    0    0    0    0    0    0    2    0    0
   0    0    0    0    0
   0    2    4    0    2    0    0    2    0    0    1    1    0    2    0    0
   0    0    2    0    0    1    0    0    2    0    0    0    0    0    0    2
   0    0    1    0    0    2    0    0    0    0    0    0    2    0    0    0
   0    0    0    0    0
STATIONARY POINT
     -0.000035102685      0.000005147046      0.000000000000      0.000000000000      0.000000000000      0.0000000
00000     -0.000000000547
END OF DATA
!FIT
!END
                                                                                        273,1          Bot
```

Figure: anpass2.in bottom

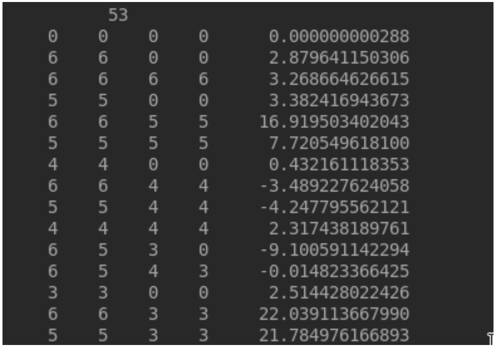
Anpass

What is the long line?

- In fitting our function, anpass can find a new minimum that is slightly different from the one from Molpro
- The long line is the series of displacements that correspond to this minimum, along with its energy
- The "!" is a comment, so we are telling anpass that the long line is a stationary point and to run a slightly different calculation

Anpass

The other important thing anpass gives us are the force constants in fort.9903



```
          53
    0    0    0    0      0.000000000288
    6    6    0    0      2.879641150306
    6    6    6    6      3.268664626615
    5    5    0    0      3.382416943673
    6    6    5    5     16.919503402043
    5    5    5    5      7.720549618100
    4    4    0    0      0.432161118353
    6    6    4    4     -3.489227624058
    5    5    4    4     -4.247795562121
    4    4    4    4      2.317438189761
    6    5    3    0     -9.100591142294
    6    5    4    3     -0.014823366425
    3    3    0    0      2.514428022426
    6    6    3    3     22.039113667990
    5    5    3    3     21.784976166893
```

Figure: fort.9903

This takes us back into intder

Intder

intder_geom.in

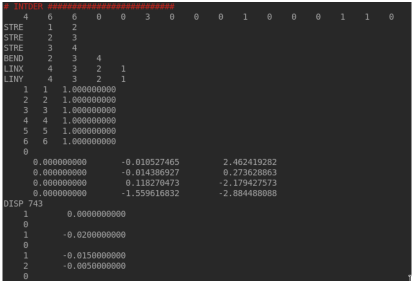
Anpass gave us a new equilibrium geometry in terms of symmetry internal coordinate displacements. We need to use Intder to turn that into a Cartesian geometry. Intder's main purpose is undertaking these transformations for us, hence calling it the glue

- Copy your intder.in file from pts and call it intder_geom.in
- Change the number of displacements to 1 and delete all of the displacements
  - You may want to keep at least one or two for alignment reference
- Take the nonzero lines from under WHERE ENERGY IS AT in anpass1.out to be the new displacements

Intder

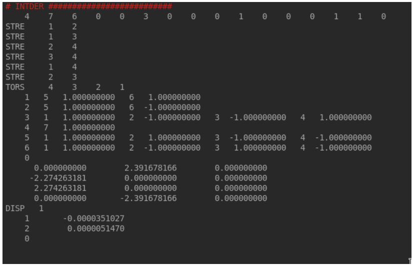
intder_geom.in

I'm mixing my examples here unfortunately, so pay attention to the displacements section only



```
# INTDER ##########################
    4    6    6    0    0    3    0    0    0    1    0    0    0    1    1    0
STRE     1    2
STRE     2    3
STRE     3    4
BEND     2    3    4
LINX     4    3    2    1
LINY     4    3    2    1
    1   1   1.000000000
    2   2   1.000000000
    3   3   1.000000000
    4   4   1.000000000
    5   5   1.000000000
    6   6   1.000000000
    0
      0.000000000       -0.010527465         2.462419282
      0.000000000       -0.014386927         0.273628863
      0.000000000        0.118270473        -2.179427573
      0.000000000       -1.559616832        -2.884488088
DISP 743
    1        0.0000000000
    0
    1       -0.0200000000
    0
    1       -0.0150000000
    2       -0.0050000000
    0
```

Figure: intder.in from pts



```
# INTDER ##########################
    4    7    6    0    0    3    0    0    0    1    0    0    0    1    1    0
STRE     1    2
STRE     1    3
STRE     2    4
STRE     3    4
STRE     1    4
STRE     2    3
TORS     4    3    2    1
    1   5   1.000000000   6   1.000000000
    2   5   1.000000000   6  -1.000000000
    3   1   1.000000000   2  -1.000000000   3  -1.000000000   4   1.000000000
    4   7   1.000000000
    5   1   1.000000000   2   1.000000000   3  -1.000000000   4  -1.000000000
    6   1   1.000000000   2  -1.000000000   3   1.000000000   4  -1.000000000
    0
      0.000000000        2.391678166        0.000000000
     -2.274263181        0.000000000        0.000000000
      2.274263181        0.000000000        0.000000000
      0.000000000       -2.391678166        0.000000000
DISP   1
    1       -0.0000351027
    2        0.0000051470
    0
```

Figure: intder_geom.in

These are the same numbers from the long line, but formatted more easily for our use

Intder

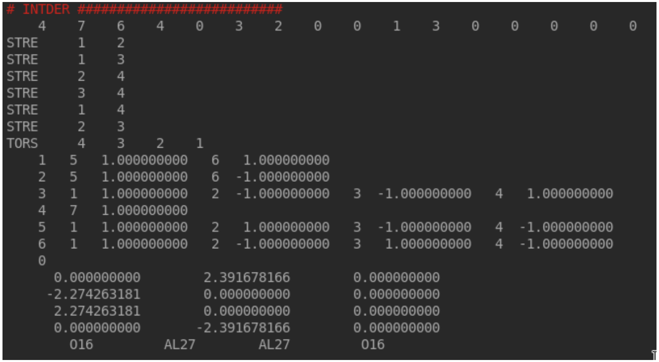
intder.in

intder_geom will produce a new equilibrium geometry in Cartesian coordinates that we can stick back in to a slightly different type of intder file

- Copy any intder.in to the freqs directory
- Take the new geometry from the very bottom of intder_geom.out and replace the one in intder.in
- Delete everything below the Cartesian coordinates
- The next line needs to be the atoms, with their masses
    - You almost definitely want to copy one of these lines from another file to get the alignment right

Intder

intder.in



```
# INTDER ##########################
    4    7    6    4    0    3    2    0    0    1    3    0    0    0    0    0
STRE      1    2
STRE      1    3
STRE      2    4
STRE      3    4
STRE      1    4
STRE      2    3
TORS      4    3    2    1
    1   5   1.000000000   6   1.000000000
    2   5   1.000000000   6  -1.000000000
    3   1   1.000000000   2  -1.000000000   3  -1.000000000   4   1.000000000
    4   7   1.000000000
    5   1   1.000000000   2   1.000000000   3  -1.000000000   4  -1.000000000
    6   1   1.000000000   2  -1.000000000   3   1.000000000   4  -1.000000000
    0
      0.000000000        2.391678166        0.000000000
     -2.274263181        0.000000000        0.000000000
      2.274263181        0.000000000        0.000000000
      0.000000000       -2.391678166        0.000000000
        O16          AL27         AL27         O16
```

Figure: Our intder.in file so far

Intder

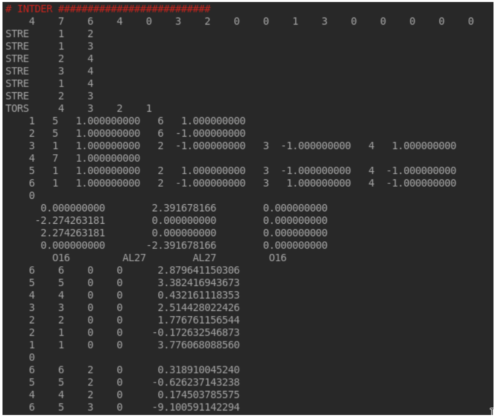
intder.in

Now we need to read in the force constants from fort.9903

- You can read fort.9903 in directly with :r fort.9903, but there is a lot of processing to do
- Instead, use format.sh; there are two ways to use it:
  - My preferred way is to run it from inside vim with :r !format.sh
  - But you can also run it at the command line $ format.sh > intder.bot
    and save the output into a file that you then read in to intder.in

Intder

intder.in



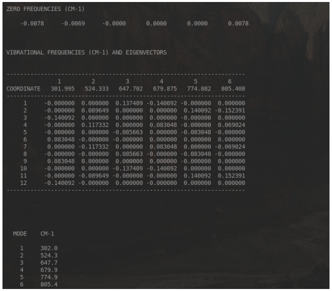
```
# INTDER ##########################
    4    7    6    4    0    3    2    0    0    1    3    0    0    0    0    0
STRE     1    2
STRE     1    3
STRE     2    4
STRE     3    4
STRE     1    4
STRE     2    3
TORS     4    3    2    1
    1   5   1.000000000   6   1.000000000
    2   5   1.000000000   6  -1.000000000
    3   1   1.000000000   2  -1.000000000   3  -1.000000000   4   1.000000000
    4   7   1.000000000
    5   1   1.000000000   2   1.000000000   3  -1.000000000   4  -1.000000000
    6   1   1.000000000   2  -1.000000000   3   1.000000000   4  -1.000000000
    0
      0.000000000       2.391678166       0.000000000
     -2.274263181       0.000000000       0.000000000
      2.274263181       0.000000000       0.000000000
      0.000000000      -2.391678166       0.000000000
        O16         AL27        AL27         O16
    6    6    0    0      2.879641150306
    5    5    0    0      3.382416943673
    4    4    0    0      0.432161118353
    3    3    0    0      2.514428022426
    2    2    0    0      1.776761156544
    2    1    0    0     -0.172632546873
    1    1    0    0      3.776068088560
    0
    6    6    2    0      0.318910045240
    5    5    2    0     -0.626237143238
    4    4    2    0      0.174503785575
    6    5    3    0     -9.100591142294
```

Figure: Part of the final intder.in file

Intder

intder.out

- Run intder: `$ intder < intder.in > intder.out`
- Verify that the harmonic frequencies at the very bottom correspond to those from Molpro (if those have finished)
- Run `$ tennis.sh` to move the intder output files to the filenames expected by spectro

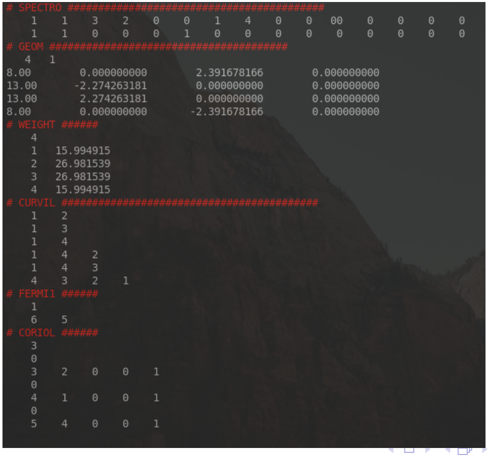


```
ZERO FREQUENCIES (CM-1)

    -0.0078     -0.0069     -0.0000      0.0000      0.0000      0.0078



VIBRATIONAL FREQUENCIES (CM-1) AND EIGENVECTORS


----------------------------------------------------------------------
                1         2         3         4         5         6
COORDINATE   301.995   524.333   647.702   679.875   774.882   805.408
----------------------------------------------------------------------
      1     -0.000000  0.000000  0.137409 -0.140092 -0.000000  0.000000
      2     -0.000000  0.089649  0.000000  0.000000  0.140092 -0.152391
      3     -0.140092  0.000000  0.000000  0.000000 -0.000000  0.000000
      4     -0.000000  0.117332  0.000000  0.083048 -0.000000  0.069024
      5     -0.000000  0.000000 -0.085663  0.000000 -0.083048 -0.000000
      6      0.083048 -0.000000 -0.000000  0.000000  0.000000  0.000000
      7      0.000000 -0.117332  0.000000  0.083048  0.000000 -0.069024
      8     -0.000000 -0.000000  0.085663 -0.000000 -0.083048 -0.000000
      9      0.083048  0.000000  0.000000  0.000000 -0.000000  0.000000
     10     -0.000000  0.000000 -0.137409 -0.140092  0.000000  0.000000
     11     -0.000000 -0.089649 -0.000000 -0.000000  0.140092  0.152391
     12     -0.140092 -0.000000  0.000000 -0.000000  0.000000  0.000000
----------------------------------------------------------------------




  MODE    CM-1

    1     302.0
    2     524.3
    3     647.7
    4     679.9
    5     774.9
    6     805.4
```

Spectro

spectro.in

You should be given a spectro.in file looking something like this

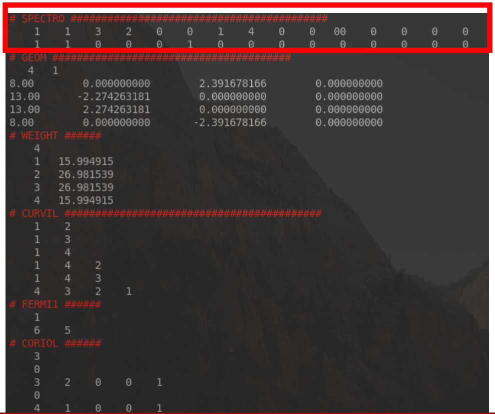


```
# SPECTRO ##########################################
    1    1    3    2    0    0    1    4    0    0   00    0    0    0    0
    1    1    0    0    0    1    0    0    0    0    0    0    0    0    0
# GEOM #######################################
   4   1
8.00        0.000000000        2.391678166        0.000000000
13.00      -2.274263181        0.000000000        0.000000000
13.00       2.274263181        0.000000000        0.000000000
8.00        0.000000000       -2.391678166        0.000000000
# WEIGHT ######
    4
    1   15.994915
    2   26.981539
    3   26.981539
    4   15.994915
# CURVIL ##########################################
    1    2
    1    3
    1    4
    1    4    2
    1    4    3
    4    3    2    1
# FERMI1 ######
    1
    6    5
# CORIOL ######
    3
    0
    3    2    0    0    1
    0
    4    1    0    0    1
    0
    5    4    0    0    1
```

Spectro

spectro.in

Don't worry about these, probably



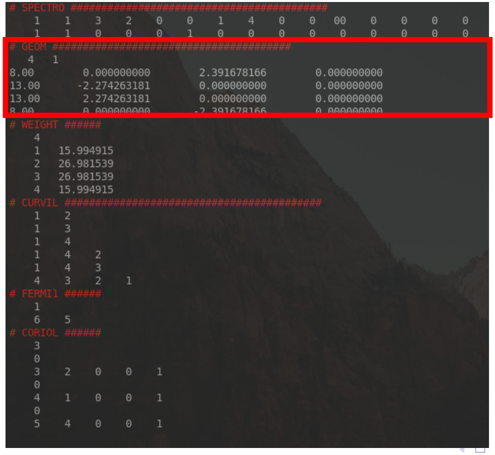
```
# SPECTRO ##########################################
    1    1    3    2    0    0    1    4    0    0   00    0    0    0    0
    1    1    0    0    0    1    0    0    0    0    0    0    0    0    0
# GEOM #######################################
   4   1
8.00        0.000000000        2.391678166        0.000000000
13.00      -2.274263181        0.000000000        0.000000000
13.00       2.274263181        0.000000000        0.000000000
8.00        0.000000000       -2.391678166        0.000000000
# WEIGHT ######
    4
    1   15.994915
    2   26.981539
    3   26.981539
    4   15.994915
# CURVIL ##########################################
    1    2
    1    3
    1    4
    1    4    2
    1    4    3
    4    3    2    1
# FERMI1 ######
    1
    6    5
# CORIOL ######
    3
    0
    3    2    0    0    1
    0
    4    1    0    0    1
```

Input directives, see manual

Spectro

spectro.in

You will definitely need to change these



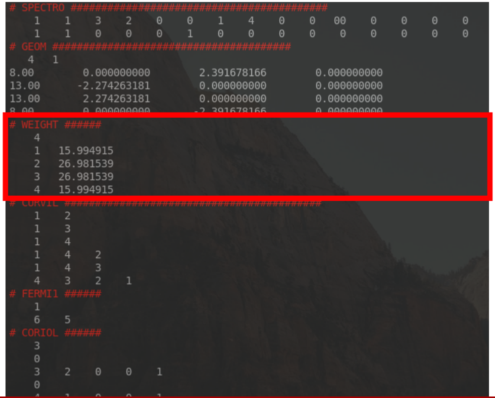
```
# SPECTRO ############################################
    1    1    3    2    0    0    1    4    0    0   00    0    0    0    0
    1    1    0    0    0    1    0    0    0    0    0    0    0    0    0
# GEOM ########################################
   4   1
8.00         0.000000000        2.391678166        0.000000000
13.00       -2.274263181        0.000000000        0.000000000
13.00        2.274263181        0.000000000        0.000000000
8.00         0.000000000       -2.391678166        0.000000000
# WEIGHT ######
    4
    1   15.994915
    2   26.981539
    3   26.981539
    4   15.994915
# CURVIL ##########################################
    1    2
    1    3
    1    4
    1    4    2
    1    4    3
    4    3    2    1
# FERMI1 ######
    1
    6    5
# CORIOL ######
    3
    0
    3    2    0    0    1
    0
    4    1    0    0    1
    0
    5    4    0    0    1
```

Geometry

Spectro

spectro.in

Change these too, from nist.gov/pml/atomic-weights-and-isotopic-compositions-relative-atomic-masses



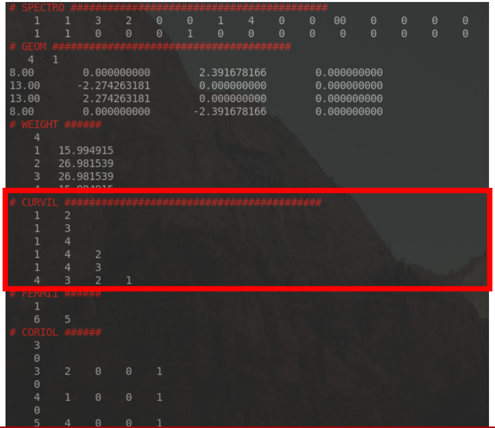
```
# SPECTRO ##########################################
    1    1    3    2    0    0    1    4    0    0   00    0    0    0    0
    1    1    0    0    0    1    0    0    0    0    0    0    0    0    0
# GEOM ####################################
   4   1
8.00         0.000000000         2.391678166         0.000000000
13.00       -2.274263181         0.000000000         0.000000000
13.00        2.274263181         0.000000000         0.000000000
8.00         0.000000000        -2.391678166         0.000000000
# WEIGHT ######
    4
    1   15.994915
    2   26.981539
    3   26.981539
    4   15.994915
# CURVIL #########################################
    1    2
    1    3
    1    4
    1    4    2
    1    4    3
    4    3    2    1
# FERMI1 ######
    1
    6    5
# CORIOL ######
    3
    0
    3    2    0    0    1
    0
    4    1    0    0    1
```

Masses

Spectro

spectro.in

You shouldn't need to touch these, they should resemble the simple internals from intder



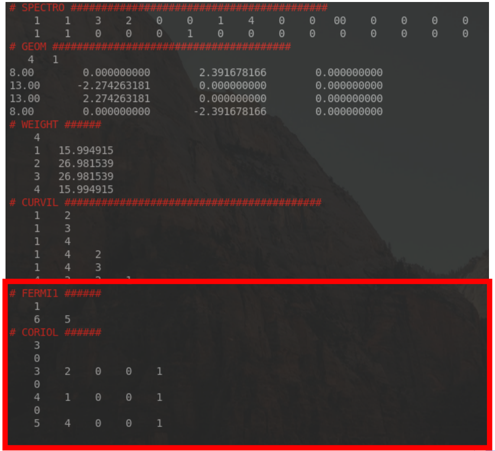
```
# SPECTRO ##########################################
    1    1    3    2    0    0    1    4    0    0   00    0    0    0    0
    1    1    0    0    0    1    0    0    0    0    0    0    0    0    0
# GEOM #######################################
   4   1
8.00        0.000000000         2.391678166        0.000000000
13.00      -2.274263181         0.000000000        0.000000000
13.00       2.274263181         0.000000000        0.000000000
8.00        0.000000000        -2.391678166        0.000000000
# WEIGHT ######
    4
    1   15.994915
    2   26.981539
    3   26.981539
    4   15.994915
# CURVIL ##########################################
    1    2
    1    3
    1    4
    1    4    2
    1    4    3
    4    3    2    1
# FERMI1 ######
    1
    6    5
# CORIOL ######
    3
    0
    3    2    0    0    1
    0
    4    1    0    0    1
    0
    5    4    0    0    1
```

Curvilinear coordinates

Spectro

spectro.in

We'll talk more about these in a minute



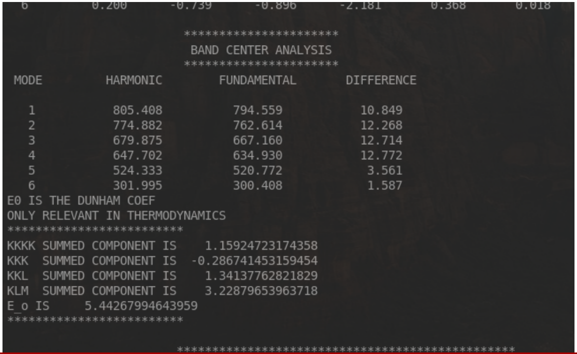
```
# SPECTRO ##########################################
    1    1    3    2    0    0    1    4    0    0   00    0    0    0    0
    1    1    0    0    0    1    0    0    0    0    0    0    0    0    0
# GEOM ########################################
   4   1
8.00        0.000000000         2.391678166         0.000000000
13.00      -2.274263181         0.000000000         0.000000000
13.00       2.274263181         0.000000000         0.000000000
8.00        0.000000000        -2.391678166         0.000000000
# WEIGHT ######
    4
    1   15.994915
    2   26.981539
    3   26.981539
    4   15.994915
# CURVIL ##########################################
    1    2
    1    3
    1    4
    1    4    2
    1    4    3
    4    3    2    1
# FERMI1 ######
    1
    6    5
# CORIOL ######
    3
    0
    3    2    0    0    1
    0
    4    1    0    0    1
    0
    5    4    0    0    1
```

Resonances

Spectro

spectro.out

- Run spectro with
  `$ spectro < spectro.in > spectro.out`
- Open spectro.out and search for “FUND” and it should look something like below

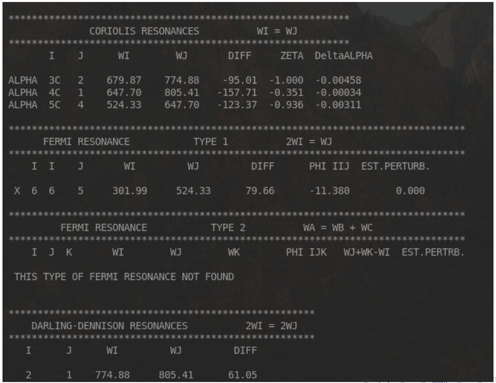


```
  6         0.200      -0.739      -0.896      -2.181      0.368      0.018

                          *********************
                          BAND CENTER ANALYSIS
                          *********************
 MODE          HARMONIC        FUNDAMENTAL       DIFFERENCE

   1             805.408         794.559           10.849
   2             774.882         762.614           12.268
   3             679.875         667.160           12.714
   4             647.702         634.930           12.772
   5             524.333         520.772            3.561
   6             301.995         300.408            1.587
E0 IS THE DUNHAM COEF
ONLY RELEVANT IN THERMODYNAMICS
*************************
KKKK SUMMED COMPONENT IS    1.15924723174358
KKK  SUMMED COMPONENT IS  -0.286741453159454
KKL  SUMMED COMPONENT IS    1.34137762821829
KLM  SUMMED COMPONENT IS    3.22879653963718
E_o IS      5.44267994643959
*************************

                         ************************************************
```

Spectro

spectro.out

- Now, jump to the top and search for “FERMI”
  - This will take you to the resonances
  - We only worry about the Fermi and Coriolis ones

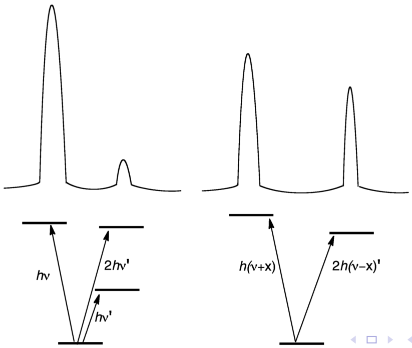


```
*************************************************************
              CORIOLIS RESONANCES          WI = WJ
*************************************************************
       I    J      WI        WJ       DIFF     ZETA  DeltaALPHA

ALPHA  3C   2    679.87    774.88    -95.01  -1.000  -0.00458
ALPHA  4C   1    647.70    805.41   -157.71  -0.351  -0.00034
ALPHA  5C   4    524.33    647.70   -123.37  -0.936  -0.00311

*********************************************************************************
      FERMI RESONANCE           TYPE 1          2WI = WJ
*********************************************************************************
    I  I    J       WI         WJ         DIFF       PHI IIJ  EST.PERTURB.

 X  6  6    5     301.99     524.33      79.66      -11.380        0.000

*********************************************************************************
         FERMI RESONANCE           TYPE 2          WA = WB + WC
*********************************************************************************
    I  J  K       WI        WJ        WK        PHI IJK   WJ+WK-WI  EST.PERTRB.

 THIS TYPE OF FERMI RESONANCE NOT FOUND


*******************************************************
    DARLING-DENNISON RESONANCES          2WI = 2WJ
*******************************************************
   I      J      WI         WJ         DIFF

   2      1    774.88     805.41      61.05
```

What is a resonance?

Fermi resonance

If two vibrational modes have the same symmetry and similar energies, the higher energy mode will shift to higher energy and the lower to lower energy, basically to get away from each other



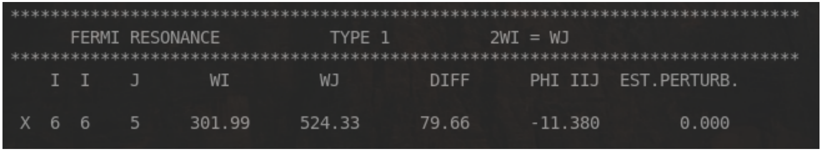
What is a resonance?

The definition may be slightly different for Coriolis resonances, but there wasn't a Wikipedia page

Fermi type 1

An overtone band, an integer multiple of one frequency, has a frequency similar to that of another band.

We see an example of this in our spectro.out file. $2\nu_6 \approx \nu_5$



```
*****************************************************************************
      FERMI RESONANCE           TYPE 1          2WI = WJ
*****************************************************************************
    I  I    J       WI         WJ         DIFF       PHI IIJ  EST.PERTURB.

 X  6  6    5     301.99     524.33      79.66      -11.380        0.000
```

80 cm^{-1} is a pretty huge difference, so the estimated perturbation is zero, but often these frequencies line up perfectly and have a substantial effect

What is a resonance?

Fermi type 2

The combination of two lower frequency modes have a frequency similar to one higher frequency mode.

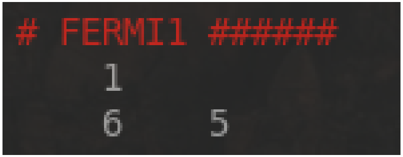
Example

Let $\nu_6 = 234.5$, $\nu_5 = 364.3$, and $\nu_2 = 600.1$ cm^{-1}.
Then $\nu_6 + \nu_5 = 598.8 \approx \nu_2$, and we could expect a larger perturbation than seen for our type 1 example.

Spectro

spectro.in again

How do we account for these resonances? We already saw how to handle type I resonances in the example spectro file



```
# FERMI1 #######
      1
      6     5
```

This says that we have one type 1 Fermi resonance, in which $2\nu_6 = \nu_5$.

Spectro

spectro.in again

To handle a type 2 Fermi resonance, such as the one in our example, we would add another block like this:

```
# FERMI2 #######
1
6        5        2
```

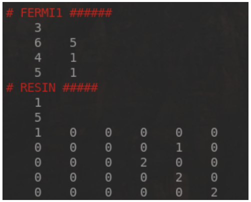
which again says that $\nu_6 + \nu_5 = \nu_2$

Spectro

spectro.in again

Fermi polyad

We can also have a situation where multiple combinations come close to a single frequency. For example, both $\nu_6 + \nu_5$ and $2\nu_4$ could be close to ν_2. This results in a Fermi polyad, which should be handled separately.



```
# FERMI1 ######
    3
    6    5
    4    1
    5    1
# RESIN #####
    1
    5
    1    0    0    0    0    0
    0    0    0    0    1    0
    0    0    0    2    0    0
    0    0    0    0    2    0
    0    0    0    0    0    2
```

Figure: Handling a Fermi polyad for Ti_2O_2

Spectro

spectro.in again

Basically if you have a polyad you need to make one of those RESIN arrays. Everything on the right hand side of the equation gets a row with a one in its column, and everything on the left hand side gets a row. In the previous example, we had a row

```
1    0    0    0    0    0
```

which represents ν_1, which appeared in two type 1 Fermi resonances, $2\nu_4 = \nu_1$ and $2\nu_5 = \nu_1$. The corresponding row for ν_5 is

```
0    0    0    0    1    0
```

Spectro

spectro.in again

The combinations of modes that make ν_5 and ν_1 get their own rows. Since in this case they are type 1 Fermi resonances, there are twos in those positions. For $2\nu_4$, which equals ν_1, we get the row

```
0    0    0    2    0    0
```

and so on. If we had a type 2 Fermi resonance, say $\nu_3 + \nu_4 = \nu_1$ too, we would add a row that looks like

```
0    0    1    1    0    0
```

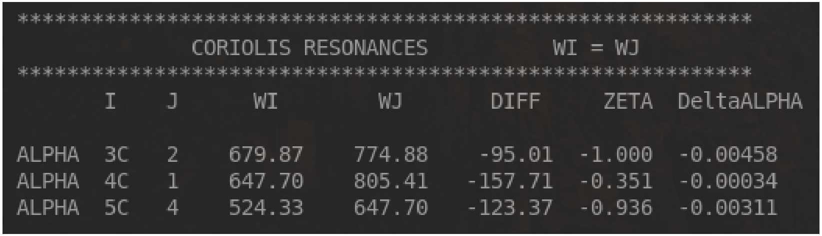
The ordering of these rows does not matter, just make sure that every left and right hand side is accounted for in equations like $\nu_3 + \nu_4 = \nu_1$.

Spectro

spectro.in again

Coriolis resonances

I'm not as sure about the physical meaning of these, but think of them the same way. Some combination of frequencies bumps into each other and wants to push the other away. Regardless, we need to handle them.



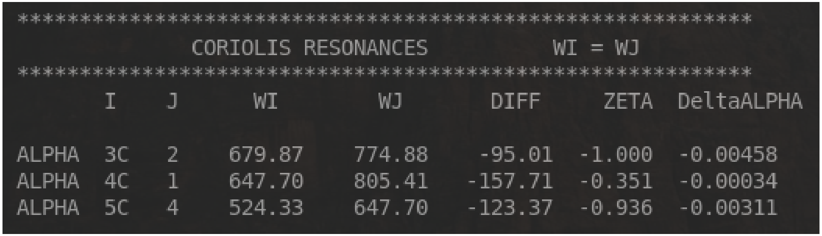
```
***********************************************************
             CORIOLIS RESONANCES          WI = WJ
***********************************************************
       I    J      WI        WJ        DIFF      ZETA  DeltaALPHA

ALPHA  3C   2    679.87    774.88     -95.01   -1.000  -0.00458
ALPHA  4C   1    647.70    805.41    -157.71   -0.351  -0.00034
ALPHA  5C   4    524.33    647.70    -123.37   -0.936  -0.00311
```

Figure: Coriolis resonances in spectro.out

Spectro

spectro.in again

The way I read these are the first number, then the second, then the letter. So 3, 2, C in the case of the first one here.



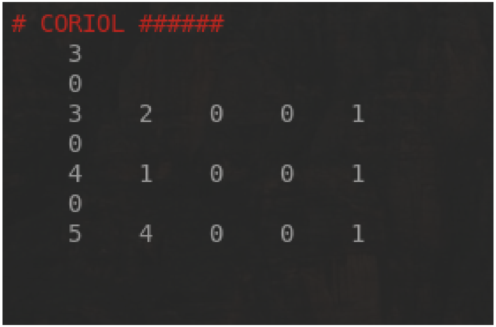
```
************************************************************
             CORIOLIS RESONANCES             WI = WJ
************************************************************
       I    J       WI        WJ        DIFF      ZETA   DeltaALPHA

ALPHA  3C   2     679.87    774.88     -95.01   -1.000  -0.00458
ALPHA  4C   1     647.70    805.41    -157.71   -0.351  -0.00034
ALPHA  5C   4     524.33    647.70    -123.37   -0.936  -0.00311
```

Figure: Coriolis resonances in spectro.out

Spectro

spectro.in again

And that's basically the way you put them in. The weirdest thing is that the letters are given by a vector. A $\rightarrow$ 1 0 0, B $\rightarrow$ 0 1 0, and C $\rightarrow$ 0 0 1



```
# CORIOL ######
    3
    0
    3    2    0    0    1
    0
    4    1    0    0    1
    0
    5    4    0    0    1
```

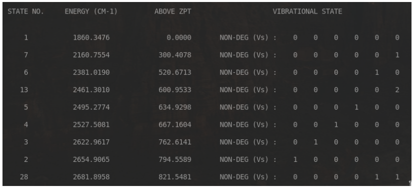
Figure: Coriolis resonances in spectro.in

Spectro
Vibrational data

- Run spectro again to finish up
- You can check FUND again to make sure it still looks right, but to see where the resonances are taken into account, we need to search instead for ZPT
  - Here, look for the "STATE NO." corresponding to 1 plus whatever mode. State 7 for ν_6, for example.
  - You can also use my script inspectro, which will print the harmonic, anharmonic, and resonance corrected frequencies, although it sometimes prints a lot more too
- You also want State 1, which is the ZPT. We report this as zero-point vibrational energy (ZPVE)

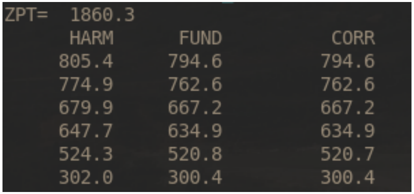
Spectro

Vibrational data



```
STATE NO.     ENERGY (CM-1)          ABOVE ZPT                 VIBRATIONAL STATE


    1           1860.3476              0.0000      NON-DEG (Vs) :    0    0    0    0    0    0
    7           2160.7554            300.4078      NON-DEG (Vs) :    0    0    0    0    0    1
    6           2381.0190            520.6713      NON-DEG (Vs) :    0    0    0    0    1    0
   13           2461.3010            600.9533      NON-DEG (Vs) :    0    0    0    0    0    2
    5           2495.2774            634.9298      NON-DEG (Vs) :    0    0    0    1    0    0
    4           2527.5081            667.1604      NON-DEG (Vs) :    0    0    1    0    0    0
    3           2622.9617            762.6141      NON-DEG (Vs) :    0    1    0    0    0    0
    2           2654.9065            794.5589      NON-DEG (Vs) :    1    0    0    0    0    0
   28           2681.8958            821.5481      NON-DEG (Vs) :    0    0    0    0    1    1
```

Figure: Sample ZPT section



```
ZPT=   1860.3
       HARM       FUND        CORR
      805.4      794.6       794.6
      774.9      762.6       762.6
      679.9      667.2       667.2
      647.7      634.9       634.9
      524.3      520.8       520.7
      302.0      300.4       300.4
```

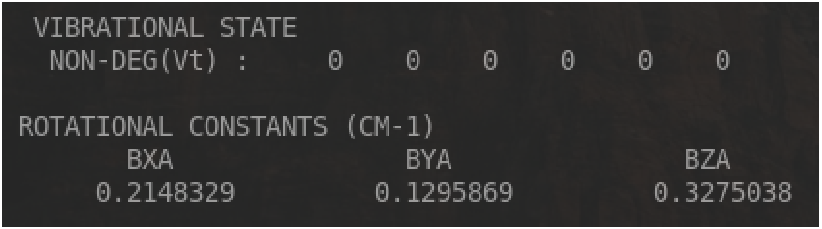
Figure: Sample inspectro output

Spectro

Rotational data

Spectro also outputs rotational constants, which can be found by searching:

- BZA - A, B, and C rotational constants for every vibrational state. BZA is usually A, BXA is B, and BYA is C. In this case we have A_0, B_0, and C_0.



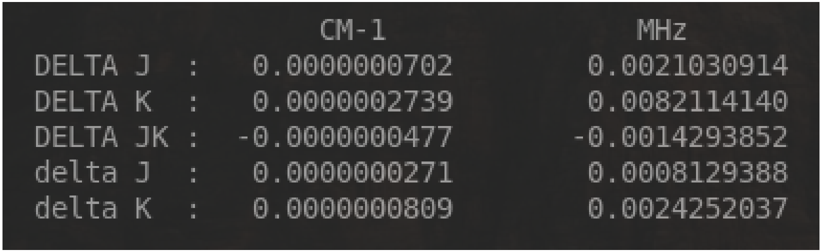
```
  VIBRATIONAL STATE
   NON-DEG(Vt) :      0     0     0     0     0     0

 ROTATIONAL CONSTANTS (CM-1)
        BXA                BYA                BZA
     0.2148329          0.1295869          0.3275038
```

Spectro

Rotational data

Spectro also outputs rotational constants, which can be found by searching:

- DELTA, quartic distortion constants

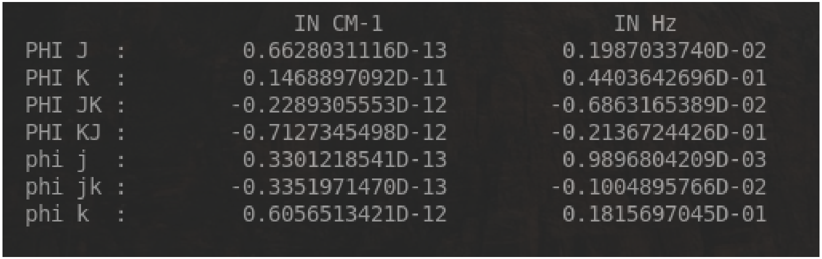


```
                     CM-1                    MHz
  DELTA J   :    0.0000000702          0.0021030914
  DELTA K   :    0.0000002739          0.0082114140
  DELTA JK  :   -0.0000000477         -0.0014293852
  delta J   :    0.0000000271          0.0008129388
  delta K   :    0.0000000809          0.0024252037
```

Spectro
Rotational data

Spectro also outputs rotational constants, which can be found by searching:

- PHI, sextic distortion constants



```
                   IN CM-1                IN Hz
PHI J  :       0.6628031116D-13      0.1987033740D-02
PHI K  :       0.1468897092D-11      0.4403642696D-01
PHI JK :      -0.2289305553D-12     -0.6863165389D-02
PHI KJ :      -0.7127345498D-12     -0.2136724426D-01
phi j  :       0.3301218541D-13      0.9896804209D-03
phi jk :      -0.3351971470D-13     -0.1004895766D-02
phi k  :       0.6056513421D-12      0.1815697045D-01
```

Reporting

Units and precision

- Vibrational frequencies to 1 decimal place in cm^{-1}, ex. 627.4 cm^{-1}
  - Harmonic frequencies are denoted ω_n (omega)
  - Anharmonic frequencies are denoted ν_n (nu)
- A, B, and C, rotational constants to 1 decimal place in MHz
- Δ and Φ constants to 3 decimal place but no more than 5 sig figs. Adjust units accordingly. For example, instead of 35154.123 kHz, use 35.154 MHz
- Bond lengths to 5 decimal places
- Bond angles (I think) to 3 decimal places

Reporting

Mode descriptions

Given symmetry internal coordinates, S_n, like those on the left, which go in the Computational Details, we often want to describe the resulting vibrational modes as linear combinations of them, as on the right

$$S_1(a_1) = r(H_1 - C_1) \quad (1)$$

$$S_2(a_1) = r(H_1 - C_2) \quad (2)$$

$$S_3(a_1) = \frac{1}{\sqrt{2}}[r(C_2 - C_3) + r(C_2 - C_4)] \quad (3)$$

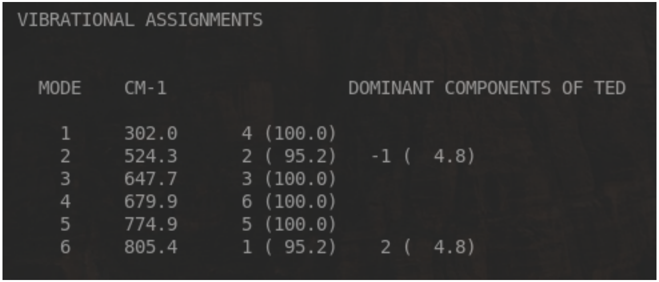
$$S_4(a_1) = \frac{1}{\sqrt{2}}[r(C_3 - H_2) + r(C_4 - H_3)] \quad (4)$$

Mode	Description	Symmetry
ω_1	$0.96S_1$	a_1
ω_2	$0.95S_4$	a_1
ω_3	$1.00S_8$	b_2
ω_4	$0.68S_2 + 0.28S_3$	a_1
ω_5	$0.74S_5 + 0.24S_2$	a_1
ω_6	$0.90S_{10} + 0.12S_9$	b_2
ω_7	$0.58S_{11} + 0.31S_9 + 0.09S_{10}$	b_2

Reporting

Mode descriptions

Fortunately, you can find this information by searching "ASSIGNMENT" in intder.out



```
VIBRATIONAL ASSIGNMENTS


  MODE     CM-1             DOMINANT COMPONENTS OF TED

    1      302.0      4 (100.0)
    2      524.3      2 ( 95.2)   -1 (   4.8)
    3      647.7      3 (100.0)
    4      679.9      6 (100.0)
    5      774.9      5 (100.0)
    6      805.4      1 ( 95.2)    2 (   4.8)
```

In this case, basically every frequency lines up 1:1 with a symmetry internal coordinate, but as shown on the previous slide this is not always the case