Matrix Oriented Perturbation Theory

Mark Boyer, Feb 7. 2024

These comprise a series of notes written while developing a library for performing vibrational perturbation theory. Some of the work, particularly that involving the lattice-path/combinatoric approach is unpublished but the primary theory can be seen in these two papers [https://-doi.org/10.1063/5.0080892, https://doi.org/10.1063/5.0121915]. A library implementing much of this can be found here [https://github.com/McCoyGroup/Psience]

There may be errors and many sections end abruptly, these are/were personal notes. Many sections with code and images are excluded from the PDF export, but the notebook with implementations is available upon request.

Perturbation Theory Formalism

This approach tries to bridge the approach presented by J. J. Sakurai and which you find on the web with the more standard approach we'd see on, e.g., Wikipedia. The idea is to convert things into matrix operations so as to make the logic of the problem more obvious.

$$H = H^{(0)} + W$$

Derivation

We'll write our Hamiltonian out as

$$H = H^{(0)} + \lambda H^{(1)} + \lambda^2 H^{(2)} + \dots$$

and we'll write our states out as

$$|n\rangle = |n\rangle^{(0)} + \lambda |n\rangle^{(1)} + \lambda^2 |n\rangle^{(2)} + \dots$$

where

$$H^{(0)} |n\rangle^{(0)} = E_n^{(0)} |n\rangle^{(0)}$$

Now we want to solve

$$H |n\rangle = E |n\rangle$$

which expands out as

$$\left(H^{(0)}+\lambda H^{(1)}+\lambda^2 H^{(2)}+\ldots\right)\left(\left|n\right\rangle^{(0)}+\lambda\left|n\right\rangle^{(1)}+\lambda^2\left|n\right\rangle^{(2)}+\ldots\right)=$$
$$\left(E_n^{(0)}+\lambda E_n^{(1)}+\lambda^2 E_n^{(2)}+\ldots\right)\left(\left|n\right\rangle^{(0)}+\lambda\left|n\right\rangle^{(1)}+\lambda^2\left|n\right\rangle^{(2)}+\ldots\right)$$

At this point we've got to do some combinatoric nonsense, by collecting terms that have the same order in λ, which gives us

$$H^{(0)}\left|n^{(0)}\right\rangle=E_n^{(0)}\left|n^{(0)}\right\rangle$$
$$H^{(0)}\left|n^{(1)}\right\rangle+H^{(1)}\left|n^{(0)}\right\rangle=E_n^{(0)}\left|n^{(1)}\right\rangle+E_n^{(1)}\left|n^{(0)}\right\rangle$$
$$H^{(0)}\left|n^{(2)}\right\rangle+H^{(1)}\left|n^{(1)}\right\rangle+H^{(2)}\left|n^{(0)}\right\rangle=E_n^{(0)}\left|n^{(2)}\right\rangle+E_n^{(1)}\left|n^{(1)}\right\rangle+E_n^{(2)}\left|n^{(0)}\right\rangle$$

then we can expand our corrections as expansion vectors, like

$$\left|n^{(1)}\right\rangle=\sum_\alpha c_\alpha^{(1)}\left|\alpha^{(0)}\right\rangle$$
$$\left|n^{(2)}\right\rangle=\sum_\alpha c_\alpha^{(2)}\left|\alpha^{(0)}\right\rangle$$

and then we project onto $\left|n^{(0)}\right\rangle$, giving us

$$\left\langle n^{(0)}\right|H^{(0)}\left|n^{(1)}\right\rangle+\left\langle n^{(0)}\right|H^{(1)}\left|n^{(0)}\right\rangle=\left\langle n^{(0)}\right|E_n^{(0)}\left|n^{(1)}\right\rangle+\left\langle n^{(0)}\right|E_n^{(1)}\left|n^{(0)}\right\rangle$$
$$\left\langle n^{(0)}\right|H^{(0)}\left|n^{(2)}\right\rangle+\left\langle n^{(0)}\right|H^{(1)}\left|n^{(1)}\right\rangle+\left\langle n^{(0)}\right|H^{(2)}\left|n^{(0)}\right\rangle=\left\langle n^{(0)}\right|E_n^{(0)}\left|n^{(2)}\right\rangle+\left\langle n^{(0)}\right|E_n^{(1)}\left|n^{(1)}\right\rangle+\left\langle n^{(0)}\right|E_n^{(2)}\left|n^{(0)}\right\rangle$$

which obviously looks like garbage, but don't worry, we're going somewhere.

Generalization & Higher-Order Corrections

Rather than work out every order of correction on by one, we'll reorganize things to give a cleaner formulation. To start, we'll rewrite our Trash Equations in a more general form as

$$\sum_{i=0}^{k}H^{(k-i)}\left|n^{(i)}\right\rangle=\sum_{i=0}^{k}E_n{}^{(k-i)}\left|n^{(i)}\right\rangle$$

we can then operate again on both sides with $\left\langle n^{(0)}\right|$ giving us

$$\sum_{i=0}^{k}\left\langle n^{(0)}\right|H^{(k-i)}\left|n^{(i)}\right\rangle=\sum_{i=0}^{k}\left\langle n^{(0)}\right|E_n{}^{(k-i)}\left|n^{(i)}\right\rangle$$
$$=\sum_{i=0}^{k}E_n{}^{(k-i)}\left\langle n^{(0)}\right|\left.n^{(i)}\right\rangle$$

and in particular this means

$$E_n{}^{(k)}=\left\langle n^{(0)}\right|H^{(k)}\left|n^{(0)}\right\rangle+\sum_{i=1}^{k}\left\langle n^{(0)}\right|H^{(k-i)}\left|n^{(i)}\right\rangle-E_n{}^{(k-i)}\left\langle n^{(0)}\right|\left.n^{(i)}\right\rangle$$

and noting that the k-term in that sum is

$$\langle n^{(0)}|\ H^{(0)}\ |n^{(k)}\rangle - {E_n}^{(0)}\ \langle n^{(0)}\ |\ n^{(k)}\rangle = {E_n}^{(0)}\ \langle n^{(0)}\ |\ n^{(k)}\rangle - {E_n}^{(0)}\ \langle n^{(0)}\ |\ n^{(k)}\rangle$$
$$= 0$$

we get

$${E_n}^{(k)} = \langle n^{(0)}|\ H^{(k)}\ |n^{(0)}\rangle + \sum_{i=1}^{k-1} \langle n^{(0)}|\ H^{(k-i)}\ |n^{(i)}\rangle - {E_n}^{(k-i)}\ \langle n^{(0)}\ |\ n^{(i)}\rangle$$

next, we need an extra condition to get unique solutions, so we choose to preserve orthonormality at all orders of λ, giving us

$$\begin{aligned}\langle n\ |\ m\rangle^{(k)} &= \sum_{i=0}^{k}\sum_{j=0}^{k-i} \langle n^{(i)}\ |\ m^{(j)}\rangle \\ &= \langle n\ |\ m\rangle^{(k-1)} + \sum_{i=0}^{k} \langle n^{(i)}\ |\ m^{(k-i)}\rangle \\ &= \delta_{nm}\end{aligned}$$

then as

$$\langle n\ |\ m\rangle^{(0)} = \langle n^{(0)}\ |\ m^{(0)}\rangle = \delta_{nm}$$

we get

$$\sum_{i=0}^{k} \langle n^{(i)}\ |\ m^{(k-i)}\rangle = 0$$

finally, we'll note that in general

$$\left(H^{(0)} - {E_n}^{(0)}\right)|n^{(k)}\rangle = \sum_{i=0}^{k-1} \left({E_n}^{(k-i)} - H^{(k-i)}\right)|n^{(i)}\rangle$$

and so if we can $|n^{(0)}\rangle ... |n^{(k-1)}\rangle$ and ${E_n}^{(k)}$, we should be able to get $|n^{(k)}\rangle$

First-Order Correction Revisited

We'll redo our first order correction in this context. First, this gives us

$${E_n}^{(1)} = \langle n^{(0)}|\ H^{(1)}\ |n^{(0)}\rangle$$

then we have

$$\langle n^{(1)}\ |\ m^{(0)}\rangle + \langle n^{(0)}\ |\ m^{(1)}\rangle = 0$$

which tells us that

$$\begin{aligned}\langle n^{(1)}\ |\ n^{(0)}\rangle &= -\langle n^{(0)}\ |\ n^{(1)}\rangle = -\langle n^{(0)}\ |\ n^{(1)}\rangle^* \\ \therefore \langle n^{(1)}\ |\ n^{(0)}\rangle &= \langle n^{(0)}\ |\ n^{(1)}\rangle = 0\end{aligned}$$

Next, we want to get an expression for $\left|n^{(1)}\right\rangle$ itself, so we expand it out in our initial basis

$$\left|n^{(1)}\right\rangle = \sum_m c_m^{(1)} \left|m^{(0)}\right\rangle$$

where we know that $c_n^{(1)}=0$

Then returning to our original equation, we have

$$\begin{aligned}
\left(H^{(0)}-E_n{}^{(0)}\right) \sum_m c_m^{(1)} \left|m^{(0)}\right\rangle &= \sum_m c_m^{(1)} \left(H^{(0)}-E_n{}^{(0)}\right) \left|m^{(0)}\right\rangle \\
&= \sum_m c_m^{(1)} \left(H^{(0)} \left|m^{(0)}\right\rangle - E_n{}^{(0)} \left|m^{(0)}\right\rangle\right) \\
&= \sum_m c_m^{(1)} \left(E_m{}^{(0)}-E_n{}^{(0)}\right) \left|m^{(0)}\right\rangle \\
\left(E_n{}^{(1)}-H^{(1)}\right) \left|n^{(0)}\right\rangle + \sum_{i=1}^{k-1} \left(E_n{}^{(k-i)}-H^{(k-i)}\right) \left|n^{(i)}\right\rangle &= \left(E_n{}^{(1)}-H^{(1)}\right) \left|n^{(0)}\right\rangle + \sum_{i=1}^{0} \left(E_n{}^{(k-i)}-H^{(k-i)}\right) \left|n^{(i)}\right\rangle \\
&= \left(E_n{}^{(1)}-H^{(1)}\right) \left|n^{(0)}\right\rangle \\
\sum_m c_m^{(1)} \left(E_m{}^{(0)}-E_n{}^{(0)}\right) \left|m^{(0)}\right\rangle &= \left(E_n{}^{(1)}-H^{(1)}\right) \left|n^{(0)}\right\rangle
\end{aligned}$$

and so assuming $m \neq n$, by projecting with $\left\langle m^{(0)}\right|$ we find that

$$\begin{aligned}
c_m^{(1)} &= -\frac{\left\langle m^{(0)}\right| H^{(1)} \left|n^{(0)}\right\rangle}{E_m{}^{(0)}-E_n{}^{(0)}} \\
&= \frac{\left\langle m^{(0)}\right| H^{(1)} \left|n^{(0)}\right\rangle}{E_n{}^{(0)}-E_m{}^{(0)}}
\end{aligned}$$

The Perturbation Operator

We will find it convenient to define two operators so that we can do less term-manipulation, first we'd like to introduce an operator that projects out the n^{th} state from our terms

$$R_n = \left(I - \left|n^{(0)}\right\rangle \left\langle n^{(0)}\right|\right) = \sum_{\alpha \neq n} \left|m^{(0)}\right\rangle \left\langle m^{(0)}\right|$$

with this defined, we can then directly construct our coefficient vectors by inverting $H^{(0)}$ and defining our "perturbation operator"

$$\Pi_n = \left(E_n^{(0)} - H^{(0)}\right)^{-1} R_n$$

that can be directly applied to $H^{(1)}$ to give

$$\left|n^{(1)}\right\rangle = \Pi_n H^{(1)} \left|n^{(0)}\right\rangle$$

Second-Order Correction

We'll return to our motivating equations

$$E_n^{(k)} = \langle n^{(0)} | \; H^{(k)} \; | n^{(0)} \rangle + \sum_{i=1}^{k} \langle n^{(0)} | \; H^{(k-i)} \; | n^{(i)} \rangle - E_n^{(k-i)} \langle n^{(0)} \, | \, n^{(i)} \rangle$$

$$\sum_{i=0}^{k} \langle n^{(i)} \, | \, m^{(k-i)} \rangle = 0$$

$$\left(H^{(0)} - E_n^{(0)} \right) | n^{(k)} \rangle = \left(E_n^{(k)} - H^{(k)} \right) | n^{(0)} \rangle + \sum_{i=1}^{k-1} \left(E_n^{(k-i)} - H^{(k-i)} \right) | n^{(i)} \rangle$$

and then letting k=2 we have

$$E_n^{(2)} = \langle n^{(0)} | \; H^{(2)} \; | n^{(0)} \rangle + \sum_{i=1}^{2} \langle n^{(0)} | \; H^{(2-i)} \; | n^{(i)} \rangle - E_n^{(2-i)} \langle n^{(0)} \, | \, n^{(i)} \rangle$$

$$\sum_{i=0}^{2} \langle n^{(i)} \, | \, m^{(2-i)} \rangle = 0$$

$$\left(H^{(0)} - E_n^{(0)} \right) | n^{(2)} \rangle = \left(E_n^{(2)} - H^{(2)} \right) | n^{(0)} \rangle + \sum_{i=1}^{1} \left(E_n^{(2-i)} - H^{(2-i)} \right) | n^{(i)} \rangle$$

taking these one-by-one we have

$$\begin{aligned} E_n^{(2)} &= \langle n^{(0)} | \; H^{(2)} \; | n^{(0)} \rangle + \langle n^{(0)} | \; H^{(1)} \; | n^{(1)} \rangle - E_n^{(1)} \langle n^{(0)} \, | \, n^{(1)} \rangle + \\ &\qquad \langle n^{(0)} | \; H^{(0)} \; | n^{(2)} \rangle - E_n^{(0)} \langle n^{(0)} \, | \, n^{(2)} \rangle \\ &= \langle n^{(0)} | \; H^{(2)} \; | n^{(0)} \rangle + \langle n^{(0)} | \; H^{(1)} \; | n^{(1)} \rangle - E_n^{(1)} \langle n^{(0)} \, | \, n^{(1)} \rangle + \\ &\qquad E_n^{(0)} \langle n^{(0)} \, | \, n^{(2)} \rangle - E_n^{(0)} \langle n^{(0)} \, | \, n^{(2)} \rangle \\ &= \langle n^{(0)} | \; H^{(2)} \; | n^{(0)} \rangle + \langle n^{(0)} | \; H^{(1)} \; | n^{(1)} \rangle \\ &= \langle n^{(0)} | \; H^{(2)} \; | n^{(0)} \rangle + \langle n^{(0)} | \; H^{(1)} \, \Pi_n \, H^{(1)} \; | n^{(0)} \rangle \end{aligned}$$

$$\langle n^{(0)} \, | \, m^{(2)} \rangle + \langle n^{(1)} \, | \, m^{(1)} \rangle + \langle n^{(2)} \, | \, m^{(0)} \rangle = 0$$

$$\langle n^{(0)} \, | \, n^{(2)} \rangle = -\frac{1}{2} \langle n^{(1)} \, | \, n^{(1)} \rangle$$

$$\begin{aligned} \left(H^{(0)} - E_n^{(0)} \right) | n^{(2)} \rangle &= \left(E_n^{(2)} - H^{(2)} \right) | n^{(0)} \rangle + \left(E_n^{(1)} - H^{(1)} \right) | n^{(1)} \rangle \\ &= \left(E_n^{(2)} - H^{(2)} \right) | n^{(0)} \rangle + \left(E_n^{(1)} - H^{(1)} \right) \Pi_n \, H^{(1)} \; | n^{(0)} \rangle \end{aligned}$$

then we'll note that the final equation gives us

$$\begin{aligned} R_n \, | n^{(2)} \rangle &= \Pi_n \left(H^{(2)} + H^{(1)} \, \Pi_n \, H^{(1)} - E_n^{(2)} - E_n^{(1)} \, \Pi_n \, H^{(1)} \right) | n^{(0)} \rangle \\ &= \left(\Pi_n \, H^{(2)} + \left(\Pi_n \, H^{(1)} \right)^2 - E_n^{(1)} (\Pi_n)^2 \, H^{(1)} \right) | n^{(0)} \rangle \end{aligned}$$

and then adding on the projection in the direction of $\left|n^{(0)}\right\rangle$ we get

$$\left|n^{(2)}\right\rangle = R_n \left|n^{(2)}\right\rangle + \left|n^{(0)}\right\rangle \left\langle n^{(0)} \middle| n^{(2)}\right\rangle$$

and so in total we have

$$\left|n^{(2)}\right\rangle = \left(\Pi_n H^{(2)} + \left(\Pi_n H^{(1)}\right)^2 - {E_n}^{(1)} (\Pi_n)^2 H^{(1)} - \frac{1}{2} \left\langle n^{(1)} \middle| n^{(1)}\right\rangle\right) \left|n^{(0)}\right\rangle$$

Higher-Order Corrections

We'll take a moment to note that we didn't need to totally expand the second-order correction. And in the name of building a general form for these corrections, it's worth thinking about how this would work iteratively.

So let's go back to our equations

$${E_n}^{(k)} = \left\langle n^{(0)}\right| H^{(k)} \left|n^{(0)}\right\rangle + \sum_{i=1}^{k-1} \left\langle n^{(0)}\right| H^{(k-i)} \left|n^{(i)}\right\rangle - {E_n}^{(k-i)} \left\langle n^{(0)} \middle| n^{(i)}\right\rangle$$

$$\sum_{i=0}^{k} \left\langle n^{(i)} \middle| m^{(k-i)}\right\rangle = 0$$

$$\left(H^{(0)} - {E_n}^{(0)}\right) \left|n^{(k)}\right\rangle = \sum_{i=0}^{k-1} \left({E_n}^{(k-i)} - H^{(k-i)}\right) \left|n^{(i)}\right\rangle$$

and now let's assume that we've gotten all of the corrections up to order $k-1$. That means we have ${E_n}^{(k)}$ directly. Now we'll turn to the second term and note that it gives us

$$\left\langle n^{(0)} \middle| m^{(k)}\right\rangle + \left\langle n^{(k)} \middle| m^{(0)}\right\rangle = -\sum_{i=1}^{k-1} \left\langle n^{(i)} \middle| m^{(k-i)}\right\rangle$$

and when $m=n$ we have

$$\left\langle n^{(0)} \middle| n^{(k)}\right\rangle = -\frac{1}{2} \sum_{i=1}^{k-1} \left\langle n^{(i)} \middle| n^{(k-i)}\right\rangle$$

finally following our previous prescription we have

$$\left|n^{(k)}\right\rangle = R_n \left|n^{(k)}\right\rangle + \left|n^{(0)}\right\rangle \left\langle n^{(0)} \middle| n^{(k)}\right\rangle$$

$$R_n \left|n^{(k)}\right\rangle = \sum_{i=0}^{k-1} \Pi_n \left({E_n}^{(k-i)} - H^{(k-i)}\right) \left|n^{(i)}\right\rangle$$

$$\therefore \left|n^{(k)}\right\rangle = \sum_{i=0}^{k-1} \Pi_n \left({E_n}^{(k-i)} - H^{(k-i)}\right) \left|n^{(i)}\right\rangle + \left|n^{(0)}\right\rangle \left\langle n^{(0)} \middle| n^{(k)}\right\rangle$$

Degenerate Perturbation Theory

Expressions

Here are the relevant equations for any level of approximation.

No Degeneracies in $H^{(0)}$

If there are no degeneracies in $H^{(0)}$ we write

$$E_n{}^{(k)} = \langle n^{(0)} | \, H^{(k)} \, | n^{(0)} \rangle + \sum_{i=1}^{k-1} \langle n^{(0)} | \, H^{(k-i)} \, | n^{(i)} \rangle - E_n{}^{(k-i)} \langle n^{(0)} | n^{(i)} \rangle$$

$$\langle n^{(0)} | n^{(k)} \rangle = -\frac{1}{2} \sum_{i=1}^{k-1} \langle n^{(i)} | n^{(k-i)} \rangle$$

$$P_n | n^{(k)} \rangle = \Pi_n \left(\sum_{i=0}^{k-1} \left(E_n{}^{(k-i)} - H^{(k-i)} \right) | n^{(i)} \rangle \right)$$

Degeneracies in $H^{(0)}$

If $P_D \, H^{(1)} \, P_D$ has entirely unique eigenvalues, then we say

$$H^{(1)} \, P_D \, | n^{(0)} \rangle = E_n^{(1)} \, P_D \, | n^{(0)} \rangle$$

$$P_U \, | n^{(k)} \rangle = \Pi_U \left(\sum_{i=1}^{k-1} \left(E_n^{(k-i)} - H^{(k-i)} \right) | n^{(i)} \rangle - H^{(k)} \, | n^{(0)} \rangle \right)$$

$$E^{(k+1)} = \langle n^{(0)} | \, P_D \, H^{(k+1)} \, P_D \, | n^{(0)} \rangle + \langle n^{(0)} | \, P_D \left(H^{(1)} - E_n^{(1)} \right) P_U \, | n^{(k)} \rangle + \sum_{i=1}^{k-1} \langle n^{(0)} | \, P_D \left(H^{(k+1-i)} - E_n^{(k+1-i)} \right)$$

$$\Pi_{Dn} = P_D \, P_n \left(H^{(1)} - E_n^{(1)} \right)^{-1} P_n \, P_D$$

$$P_{Dn} \, | n^{(k)} \rangle = \Pi_{Dn} \left(\sum_{i=0}^{k-1} \left(E_n^{(k+1-i)} - H^{(k+1-i)} \right) | n^{(i)} \rangle + \left(E_n^{(1)} - H^{(1)} \right) P_U \, | n^{(k)} \rangle \right)$$

$$\langle n^{(0)} | n^{(k)} \rangle = -\frac{1}{2} \sum_{i=1}^{k-1} \langle n^{(i)} | n^{(k-i)} \rangle$$

where it's important to note that $E^{(k+1)}$ is needed to get $P_D \, | n^{(k)} \rangle$, but the reverse is not true

Degeneracies in $P_D \, H^{(1)} \, P_D$

Degeneracies in $P_G \left(H^{(2)} - H^{(1)} \, \Pi_U \, H^{(1)} \right) P_G$

I think it's unlikely we'll get here so...we'll cross that bridge when we get there?

[WORKING OFF BAD FORMS OF EQUATIONS] Connections

States Required To Evaluate

Formally, $H^{(0)}$, $H^{(1)}$, and $H^{(2)}$ should be evaluated in

Implementations

Visualizations

Nearly-Degenerate Analysis

Table of Evaluated Terms

Annoyingly, this arrangement of the terms isn't very popular. Instead scientists usually write out direct expanded expressions in terms of the zero-order states. So we're gonna tabulate first, second, and third order correction expressions to make working with existing stuff cleaner. We won't do full expansions of the terms, because that's a pain in the ass, but we'll get part of the way there.

First Order

$$
\begin{aligned}
E_n^{(1)} &= \langle n^{(0)} | \, H^{(1)} \, | n^{(0)} \rangle \\
\langle n^{(0)} | \, n^{(1)} \rangle &= 0 \\
| n^{(1)} \rangle &= \Pi_n \left(H^{(1)} - E_n^{(1)} \right) | n^{(0)} \rangle \\
&= \Pi_n \, H^{(1)} \, | n^{(0)} \rangle
\end{aligned}
$$

Second Order

$$
\begin{aligned}
E_n^{(2)} &= \langle n^{(0)} | \, H^{(2)} \, | n^{(0)} \rangle + \langle n^{(0)} | \, H^{(1)} \, | n^{(1)} \rangle - E_n^{(1)} \langle n^{(0)} | \, n^{(1)} \rangle \\
\langle n^{(0)} | \, n^{(2)} \rangle &= -\frac{1}{2} \langle n^{(1)} | \, n^{(1)} \rangle \\
| n^{(2)} \rangle &= \Pi_n \left(H^{(2)} - E_n^{(2)} \right) | n^{(0)} \rangle + \Pi_n \left(H^{(1)} - E_n^{(1)} \right) | n^{(1)} \rangle + \langle n^{(0)} | \, n^{(2)} \rangle \, | n^{(0)} \rangle \\
&= \Pi_n \, H^{(2)} \, | n^{(0)} \rangle + \Pi_n \left(H^{(1)} - E_n^{(1)} \right) \Pi_n \, H^{(1)} \, | n^{(0)} \rangle - \frac{1}{2} \langle n^{(1)} | \, n^{(1)} \rangle \, | n^{(0)} \rangle \\
&= \Pi_n \, H^{(2)} \, | n^{(0)} \rangle + \Pi_n \, H^{(1)} \, \Pi_n \, H^{(1)} \, | n^{(0)} \rangle - E_n^{(1)} \, \Pi_n \, \Pi_n \, H^{(1)} \, | n^{(0)} \rangle - \frac{1}{2} \langle n^{(1)} | \, n^{(1)} \rangle \, | n^{(0)} \rangle
\end{aligned}
$$

Third Order

$$
\begin{aligned}
E_n^{(3)} &= \langle n^{(0)} | \, H^{(3)} \, | n^{(0)} \rangle + \sum_{i=1}^{2} \langle n^{(0)} | \, H^{(3-i)} \, | n^{(i)} \rangle - E_n^{(3-i)} \langle n^{(0)} | \, n^{(i)} \rangle \\
&= \langle n^{(0)} | \, H^{(3)} \, | n^{(0)} \rangle + \langle n^{(0)} | \, H^{(2)} \, | n^{(1)} \rangle - E_n^{(2)} \langle n^{(0)} | \, n^{(1)} \rangle + \langle n^{(0)} | \, H^{(1)} \, | n^{(2)} \rangle - E_n^{(1)} \langle n^{(0)} | \, n^{(2)} \rangle
\end{aligned}
$$

$$\langle n^{(0)} | n^{(3)} \rangle = -\frac{1}{2} \sum_{i=1}^{2} \langle n^{(i)} | n^{(3-i)} \rangle$$
$$= -\frac{1}{2} \left(\langle n^{(1)} | n^{(2)} \rangle + \langle n^{(2)} | n^{(1)} \rangle \right)$$
$$= -\langle n^{(1)} | n^{(2)} \rangle$$
$$| n^{(3)} \rangle = \sum_{i=0}^{2} \Pi_n \left(H^{(3-i)} - E_n^{(3-i)} \right) | n^{(i)} \rangle + | n^{(0)} \rangle \langle n^{(0)} | n^{(3)} \rangle$$
$$= \Pi_n \left(H^{(3)} - E_n^{(3)} \right) | n^{(0)} \rangle + \Pi_n \left(H^{(2)} - E_n^{(2)} \right) | n^{(1)} \rangle + \Pi_n \left(H^{(1)} - E_n^{(1)} \right) | n^{(2)} \rangle + | n^{(0)} \rangle \langle n^{(0)} | n^{(3)} \rangle$$
$$= \Pi_n H^{(3)} | n^{(0)} \rangle + \Pi_n H^{(2)} | n^{(1)} \rangle + \Pi_n H^{(1)} | n^{(2)} \rangle - E_n^{(2)} \Pi_n | n^{(1)} \rangle - \langle n^{(1)} | n^{(2)} \rangle | n^{(0)} \rangle$$

Fourth Order Energy

$$E_n^{(k)} = \langle n^{(0)} | H^{(4)} | n^{(0)} \rangle + \langle n^{(0)} | H^{(3)} | n^{(1)} \rangle + \langle n^{(0)} | H^{(2)} | n^{(2)} \rangle + \langle n^{(0)} | H^{(1)} | n^{(3)} \rangle - E_n^{(2)} \langle n^{(0)} | n^{(2)} \rangle - E_n^{(1)} \langle n^{(0)} | n^{(3)}$$

$$| n^{(2)} \rangle = \Pi_n H^{(2)} | n^{(0)} \rangle + \Pi_n H^{(1)} \Pi_n H^{(1)} | n^{(0)} \rangle - E_n^{(1)} \Pi_n \Pi_n H^{(1)} | n^{(0)} \rangle - \frac{1}{2} \langle n^{(1)} | n^{(1)} \rangle | n^{(0)} \rangle$$

$$| n^{(3)} \rangle = \Pi_n H^{(3)} | n^{(0)} \rangle + \Pi_n H^{(2)} | n^{(1)} \rangle + \Pi_n H^{(1)} | n^{(2)} \rangle - E_n^{(2)} \Pi_n | n^{(1)} \rangle - \langle n^{(1)} | n^{(2)} \rangle | n^{(0)} \rangle$$

Operator Representations

Often we have an operator A and need to get something like

$$\langle n | A | m \rangle = \left(\sum_k \langle n^{(k)} | \lambda^k \right) A \left(\sum_k | m^{(l)} \rangle \lambda^k \right)$$

naively, we might think that we can just evaluate this out directly. Unfortunately, because of the truncation done in the perturbation expansion, this would introduce errors. Instead, we need to first expand A itself out as a series of perturbations

$$A = \sum_k A^{(k)} \lambda^k$$

and then we can compute the perturbative correction to the expectation value by matching up terms with the same total value of λ^k. Since we're going to be adding up terms that look like

$$\langle n^{(\alpha)} | A^{(\gamma)} | m^{(\beta)} \rangle$$

we unfortunately have three indices to sum over, i.e. we're looking for non-negative, integral solutions to

$$\alpha + \beta + \gamma = k$$

which we'll want to do in two steps:

- Fix a value for $\alpha \in \{0 \ldots k\}$
- Sum over the solutions to $\beta + \gamma = k - \alpha$

or written out explicitly

$$\langle n| A |m\rangle^{(k)} = \sum_{\alpha=0}^{k} \sum_{\beta=0}^{k-\alpha} \langle n^{(\alpha)}| A^{(k-\alpha-\beta)} |m^{(\beta)}\rangle$$

which if done out explicitly for k=2 gives us

$$\langle n| A |m\rangle^{(2)} = \langle n^{(0)}| A^{(2)} |m^{(0)}\rangle + \langle n^{(0)}| A^{(1)} |m^{(1)}\rangle + \langle n^{(0)}| A^{(0)} |m^{(2)}\rangle + \langle n^{(1)}| A^{(1)} |m^{(0)}\rangle + \langle n^{(1)}| A^{(0)} |m^{(1)}\rangle + \langle n^{(2)}| A^{(0)} |m^{(0)}\rangle$$

Selection Rules

It's worth considering how this play with selection rules, to improve efficiency. First off, we will note that when we have

$$\langle n^{(\alpha)}| A^{(k-\alpha-\beta)} |m^{(\beta)}\rangle$$

in general $\langle n^{(\alpha)}|$ will be an expansion over a select set of initial states and $|m^{(\beta)}\rangle$ will be an expansion over a different one. We'll write this as

$$\langle n^{(\alpha)}| = \sum_k c_k^{(n)} \langle \phi_k^{(0)}|$$
$$|m^{(\beta)}\rangle = \sum_j c_j^{(m)} |\phi_j{}^{(0)}\rangle$$

Now if $A^{(k-\alpha-\beta)}$ has an associated set of selection rules, $\{\Delta\}$, then we can generate the set of transformed states

$$\{\Delta\} \otimes \{|\phi_j{}^{(0)}\rangle\}$$

and take the intersection of this set with $\{\langle \phi_k^{(0)}|\}$ to determine what our active space is.

Alternately, as long as our operator is Hermitian, we can apply $\{\Delta\}$ to $\{\langle \phi_k^{(0)}|\}$

Degeneracies Derivation

Nearly-Degenerate PT Orthonormality

Sample Implementation

Perturbation Theory Extensions

We're going to redo what we did before, but with more insight this time.

We start out by defining

$$\Delta \mathrm{H}_n^{(i)} = \left(H^{(i)} - E_n^{(i)}\right)$$

the benefit of this definition is that if $\left|n\right\rangle$ is an eigenstate of $H^{(i)}$ then $\Delta \mathrm{H}_n^{(i)} \left|n\right\rangle = \emptyset$. This will allow us to understand the operations in the perturbation theory as accounting for the deviation of each applied state from being an eigenstate of the operator.

Next we'll introduce the concept of a perturbation operator which will be an approximate inverse to a term looking like

$$P_S^\sigma = P_S - \left|s^{(0)}\right\rangle \left\langle s^{(0)}\right|$$
$$\Pi_S^\sigma = -P_S^\sigma \left(H_S^{(0)} - E_\sigma^{(0)}\right)^{-1} P_S^\sigma$$

where $H_S^{(0)}$ is an effective zero-order Hamiltonian in the space S and $E_\sigma^{(0)}$ is the effective energy of the zero-order state $\left|\sigma^{(0)}\right\rangle$ in this space. This will be important when we handle degeneracies, as we will do this by choosing subspaces where the degeneracies go away.

The systems of equations we're solving are given by

$$\emptyset = \sum_{i=0}^{k} \Delta \mathrm{H}_n^{(k-i)} \left|n^{(i)}\right\rangle$$

up to some maximal order of correction k.

This leads to a straightforward expression for the energy at order k

$$E_n^{(k)} = \left\langle n^{(0)}\right| H^{(k)} \left|n^{(0)}\right\rangle + \sum_{i=1}^{k-1} \left\langle n^{(0)}\right| \Delta \mathrm{H}_n^{(k-i)} \left|n^{(i)}\right\rangle$$

where we've made use of $\left\langle n^{(0)}\right| \Delta \mathrm{H}_n^{(0)} = \emptyset$

And similarly we able to obtain expressions for most of the correction to the wave functions by calling our total state space T and writing

$$P_S^\sigma \left|n^{(k)}\right\rangle = \Pi_S^\sigma \sum_{i=0}^{k-1} \Delta \mathrm{H}_n^{(k-i)} \left|n^{(i)}\right\rangle$$

which coms from

$$\begin{aligned} P_U\left(-\Delta \mathrm{H}_n^{(0)}\right) \left|n^{(k)}\right\rangle &= P_U\left(-\Delta \mathrm{H}_n^{(0)}\right) P_U \left|n^{(k)}\right\rangle + P_U\left(-\Delta \mathrm{H}_n^{(0)}\right) P_D \left|n^{(k)}\right\rangle \\ &= P_U\left(-\Delta \mathrm{H}_n^{(0)}\right) P_U \left|n^{(k)}\right\rangle \text{ (by commutativity)} \\ &= P_U \sum_{i=0}^{k-1} \Delta \mathrm{H}_n^{(k-i)} \left|n^{(i)}\right\rangle \end{aligned}$$

so

$$P_S^\sigma \left|n^{(k)}\right\rangle = \Pi_S^\sigma \sum_{i=0}^{k-1} \Delta \mathrm{H}_n^{(k-i)} \left|n^{(i)}\right\rangle$$

finally by enforcing a normalization criterion and using the fact that $\left\langle n^{(k)} \middle| n^{(0)} \right\rangle$ is the complex conjugate of $\left\langle n^{(0)} \middle| n^{(k)} \right\rangle$ we have

$$\begin{aligned}
\left\langle n^{(0)} \middle| n^{(k)} \right\rangle + \left\langle n^{(k)} \middle| n^{(0)} \right\rangle &= -\sum_{i=1}^{k-1} \left\langle n^{(i)} \middle| n^{(k-i)} \right\rangle \\
\mathrm{Re}\left(\left\langle n^{(0)} \middle| n^{(k)} \right\rangle\right) &= \frac{\left\langle n^{(0)} \middle| n^{(k)} \right\rangle + \left\langle n^{(k)} \middle| n^{(0)} \right\rangle}{2} \\
&= -\frac{1}{2} \sum_{i=1}^{k-1} \left\langle n^{(i)} \middle| n^{(k-i)} \right\rangle
\end{aligned}$$

and the imaginary part is indeterminate, **so we can choose it to be 0** *(check this)*

Formal Analysis of Non-Degenerate Wavefunctions

We'll suppose that for every to order $j < k$ we can write

$$P_U \left|n^{(j)}\right\rangle = \Pi_n A_n^{(j)} \left|n^{(0)}\right\rangle$$

well then clearly

$$P_U \left|n^{(k)}\right\rangle = \Pi_n \sum_{i=0}^{k-1} \Delta \mathrm{H}_n^{(k-i)} \Pi_n A_n^{(i)} \left|n^{(0)}\right\rangle + \Delta \mathrm{H}_n^{(k-i)} \left|n^{(0)}\right\rangle \left\langle n^{(0)} \middle| n^{(i)} \right\rangle$$

which means

$$A_n^{(k)} = \sum_{i=0}^{k-1} \Delta \mathrm{H}_n^{(k-i)} \left(\Pi_n A_n^{(i)} + \left\langle n^{(0)} \middle| n^{(i)} \right\rangle\right)$$

and

$$\left|n^{(k)}\right\rangle = \left(\Pi_n A_n^{(k)} + \left\langle n^{(0)} \middle| n^{(k)} \right\rangle\right) \left|n^{(0)}\right\rangle$$

this definition can also be nicely related to our key sums, as

$$\begin{aligned}
\Delta \mathrm{H}_n^{(k-i)} \left|n^{(i)}\right\rangle &= \Delta \mathrm{H}_n^{(k-i)} \left(\Pi_n A^{(i)} + \left\langle n^{(0)} \middle| n^{(i)} \right\rangle\right) \left|n^{(0)}\right\rangle \\
\sum_{i=0}^{k-1} \Delta \mathrm{H}_n^{(k-i)} \left|n^{(i)}\right\rangle &= \sum_{i=0}^{k-1} \Delta \mathrm{H}_n^{(k-i)} \left(\Pi_n A^{(i)} + \left\langle n^{(0)} \middle| n^{(i)} \right\rangle\right) \left|n^{(0)}\right\rangle \\
&= A^{(k)} \left|n^{(0)}\right\rangle
\end{aligned}$$

and then we recall that

$$\sum_{i=0}^{k} \Delta \mathrm{H}_n^{(k-i)} \left| n^{(i)} \right\rangle = \emptyset$$

but also

$$\begin{aligned} \left\langle n^{(0)} \right| \sum_{i=0}^{k} \Delta \mathrm{H}_n^{(k-i)} \left| n^{(i)} \right\rangle &= \left\langle n^{(0)} \right| \sum_{i=0}^{k-1} \Delta \mathrm{H}_n^{(k-i)} \left| n^{(i)} \right\rangle \\ &= \left\langle n^{(0)} \right| A_n^{(k)} \left| n^{(0)} \right\rangle \end{aligned}$$

therefore for every k we get

$$\left\langle n^{(0)} \right| A_n^{(k)} \left| n^{(0)} \right\rangle = 0$$

Energy Ideas

We'll note that

$$E_n^{(k)} = \left\langle n^{(0)} \right| H^{(k)} \left| n^{(0)} \right\rangle + \sum_{i=1}^{k-1} \left\langle n^{(0)} \right| \Delta \mathrm{H}_n^{(k-i)} \left| n^{(i)} \right\rangle$$

which means we will want to define an operator $B_n^{(k)}$ such that

$$B_n^{(k)} \left| n^{(0)} \right\rangle = \left(A_n^{(k)} - \Delta \mathrm{H}_n^{(k)} \right) \left| n^{(0)} \right\rangle = \sum_{i=1}^{k-1} \Delta \mathrm{H}_n^{(k-i)} \left| n^{(i)} \right\rangle$$

giving us

$$E_n^{(k)} = \left\langle n^{(0)} \right| H^{(k)} \left| n^{(0)} \right\rangle + \left\langle n^{(0)} \right| B_n^{(k)} \left| n^{(0)} \right\rangle$$

which means we can transform $B_n^{(k)}$ iteratively, so when we write

$$A_n^{(k)} \left| n^{(0)} \right\rangle = \sum_{i=0}^{k-1} \Delta \mathrm{H}_n^{(k-i)} \left(\Pi_n A_n^{(i)} + \left\langle n^{(0)} \middle| n^{(i)} \right\rangle \right) \left| n^{(0)} \right\rangle$$

we can instead have

$$B_n^{(k)} + \Delta \mathrm{H}_n^{(k)} = \sum_{i=0}^{k-1} \Delta \mathrm{H}_n^{(k-i)} \left(\Pi_n \left(B_n^{(i)} + \Delta \mathrm{H}_n^{(i)} \right) + \left\langle n^{(0)} \middle| n^{(i)} \right\rangle \right)$$

$$\begin{aligned} \therefore B_n^{(k)} &= -\Delta \mathrm{H}_n^{(k)} + \sum_{i=0}^{k-1} \Delta \mathrm{H}_n^{(k-i)} \left(\Pi_n B_n^{(i)} + \Pi_n \Delta \mathrm{H}_n^{(i)} + \left\langle n^{(0)} \middle| n^{(i)} \right\rangle \right) \\ &= \sum_{i=1}^{k-1} \Delta \mathrm{H}_n^{(k-i)} \left(\Pi_n B_n^{(i)} + \Pi_n \Delta \mathrm{H}_n^{(i)} + \left\langle n^{(0)} \middle| n^{(i)} \right\rangle \right) + \Delta \mathrm{H}_n^{(k)} \left(\Pi_n \left(B_n^{(0)} + \Delta \mathrm{H}_n^{(0)} \right) + \left\langle n^{(0)} \middle| n^{(0)} \right\rangle \right) - \Delta \mathrm{H}_n^{(k)} \\ &= \sum_{i=1}^{k-1} \Delta \mathrm{H}_n^{(k-i)} \left(\Pi_n B_n^{(i)} + \Pi_n \Delta \mathrm{H}_n^{(i)} + \left\langle n^{(0)} \middle| n^{(i)} \right\rangle \right) + \Delta \mathrm{H}_n^{(k)} - \Delta \mathrm{H}_n^{(k)} \end{aligned}$$

$$= \sum_{i=1}^{k-1} \Delta \mathrm{H}_n^{(k-i)} \left(\Pi_n \, B_n^{(i)} + \Pi_n \, \Delta \mathrm{H}_n^{(i)} + \left\langle n^{(0)} \middle| n^{(i)} \right\rangle \right)$$

$$= \sum_{i=1}^{k-1} \Delta \mathrm{H}_n^{(k-i)} \left\langle n^{(0)} \middle| n^{(i)} \right\rangle + \Delta \mathrm{H}_n^{(k-i)} \, \Pi_n \left(B_n^{(i)} + \Delta \mathrm{H}_n^{(i)} \right)$$

which provides a not totally terrible expression we can calculate to keep track of to build $B_n^{(k)}$ which then gives us

$$E_n^{(k)} = \left\langle n^{(0)} \middle| H^{(k)} \middle| n^{(0)} \right\rangle + \left\langle n^{(0)} \middle| B_n^{(k)} \middle| n^{(0)} \right\rangle$$

$$\left| n^{(k)} \right\rangle = \Pi_n \left(B_n^{(k)} + \Delta \mathrm{H}_n^{(k)} \right) \left| n^{(0)} \right\rangle + \left| n^{(0)} \right\rangle \left\langle n^{(0)} \middle| n^{(i)} \right\rangle$$

furthermore, at order k+1 this gives

$$) \left| n^{(0)} \right\rangle + \left\langle n^{(0)} \middle| B_n^{(k+1)} \middle| n^{(0)} \right\rangle$$

$$) \left| n^{(0)} \right\rangle + \sum_{i=1}^{k+1} \left\langle n^{(0)} \middle| \Delta \mathrm{H}_n^{(k+1-i)} \left\langle n^{(0)} \middle| n^{(i)} \right\rangle + \Delta \mathrm{H}_n^{(k+1-i)} \, \Pi_n \left(B_n^{(i)} + \Delta \mathrm{H}_n^{(i)} \right) \middle| n^{(0)} \right\rangle$$

$$) \left| n^{(0)} \right\rangle + \sum_{i=1}^{k+1} \left\langle n^{(0)} \middle| \Delta \mathrm{H}_n^{(k+1-i)} \middle| n^{(0)} \right\rangle \left\langle n^{(0)} \middle| n^{(i)} \right\rangle + \left\langle n^{(0)} \middle| H^{(k+1-i)} \, \Pi_n \, H^{(i)} \middle| n^{(0)} \right\rangle + \left\langle n^{(0)} \middle| H^{(k+1-i)} \, \Pi_n \, B_n^{(i)} \middle| n^{(0)} \right\rangle$$

$$) \left| n^{(0)} \right\rangle + \sum_{i=1}^{k+1} \left\langle n^{(0)} \middle| H^{(k+1-i)} \middle| n^{(0)} \right\rangle \left\langle n^{(0)} \middle| n^{(i)} \right\rangle + \left\langle n^{(0)} \middle| H^{(k+1-i)} \, \Pi_n \, H^{(i)} \middle| n^{(0)} \right\rangle + \left\langle n^{(0)} \middle| H^{(k+1-i)} \, \Pi_n \, B_n^{(i)} \middle| n^{(0)} \right\rangle -$$

which shows that we can nicely partition $B_n^{(k)}$ into a portion that only requires transformations of the Hamiltonian and a portion will all of the energy shifts, i.e. we'll be able to write $B_n^{(k)} = \mathfrak{H}_n^{(k)} - \epsilon_n^{(k)}$.

To go further, we have (expanding $B_n^{(k)} = \mathfrak{H}_n^{(k)} + \epsilon_n^{(k)}$ where it matters)

$$)) \left| n^{(i)} \right\rangle + \Delta \mathrm{H}_n^{(k-i)} \, \Pi_n \left(B_n^{(i)} + \Delta \mathrm{H}_n^{(i)} \right)$$

$$\left| n^{(i)} \right\rangle + \; H^{(k-i)} \, \Pi_n \, H^{(i)} + H_n^{(k-i)} \, \Pi_n \, B_n^{(i)} - E_n^{(k-i)} \, \Pi_n \, B_n^{(i)} - E_n^{(k-i)} \, \Pi_n \, H^{(i)} - E^{(i)} \, \Delta \mathrm{H}_n^{(k-i)} \, \Pi_n - E^{(k-i)} \left\langle n^{(0)} \right.$$

$$n^{(i)} \rangle + \; H^{(k-i)} \, \Pi_n \, H^{(i)} + H_n^{(k-i)} \, \Pi_n \, \mathfrak{H}_n^{(i)} - H_n^{(k-i)} \, \Pi_n \, \epsilon_n^{(i)} - E_n^{(k-i)} \, \Pi_n \, B_n^{(i)} - E_n^{(k-i)} \, \Pi_n \, H^{(i)} - E^{(i)} \, \Delta \mathrm{H}_n^{(k-i)} \, \Pi$$

which pretty clearly gives us

$$= \sum_{i=1}^{k-1} H_n^{(k-i)} \left(\left\langle n^{(0)} \middle| n^{(i)} \right\rangle + \; \Pi_n \, H^{(i)} + \Pi_n \, \mathfrak{H}_n^{(i)} \right)$$

$$= \sum_{i=1}^{k-1} H_n^{(k-i)} \, \Pi_n \, \epsilon_n^{(i)} + E_n^{(k-i)} \, \Pi_n \, B_n^{(i)} + E_n^{(k-i)} \, \Pi_n \, H^{(i)} + E^{(i)} \, \Delta \mathrm{H}_n^{(k-i)} \, \Pi_n + E_n^{(k-i)} \left\langle n^{(0)} \middle| n^{(i)} \right\rangle$$

$$= \sum_{i=1}^{k-1} E_n^{(k-i)} \left\langle n^{(0)} \middle| n^{(i)} \right\rangle + E_n^{(k-i)} \Pi_n \mathcal{H}_n^{(i)} + E_n^{(k-i)} \Pi_n H^{(i)} + \Delta \mathrm{H}_n^{(k-i)} \Pi_n \epsilon_n^{(i)} + H_n^{(k-i)} \Pi_n E_n^{(i)} - E_n^{(i)} E_n^{(k-i)}$$

$$= \sum_{i=1}^{k-1} E_n^{(k-i)} \left(\left\langle n^{(0)} \middle| n^{(i)} \right\rangle + \Pi_n H^{(i)} + \Pi_n \mathcal{H}_n^{(i)} \right) + \Delta \mathrm{H}_n^{(k-i)} \Pi_n \left(\epsilon_n^{(i)} + E_n^{(i)} \right)$$

but it turns out this is less useful that I thought...

Condensed B operator

We can actually do something pretty convenient. We note that

$$E_n^{(k)} = \left\langle n^{(0)} \middle| H^{(k)} + B_n^{(k)} \middle| n^{(0)} \right\rangle$$

and

$$\begin{aligned} A_n^{(k)} \left| n^{(0)} \right\rangle &= \left(H^{(k)} + B_n^{(k)} - E_n^{(k)} \right) \left| n^{(0)} \right\rangle \\ &= \left(H^{(k)} + B_n^{(k)} - E_n^{(k)} \right) \left| n^{(0)} \right\rangle \\ &= \left(H^{(k)} + B_n^{(k)} - \left\langle n^{(0)} \middle| H^{(k)} + B_n^{(k)} \middle| n^{(0)} \right\rangle \right) \left| n^{(0)} \right\rangle \end{aligned}$$

which means instead we could write all of this as

$$\begin{aligned} \mathcal{H}_n^{(k)} &= H^{(k)} + B_n^{(k)} \\ E_n^{(k)} &= \left\langle n^{(0)} \middle| \mathcal{H}_n^{(k)} \middle| n^{(0)} \right\rangle \\ \left| n^{(k)} \right\rangle &= \Pi_n \Delta \mathcal{H}_n^{(k)} \left| n^{(0)} \right\rangle + \left| n^{(0)} \right\rangle \left\langle n^{(0)} \middle| n^{(k)} \right\rangle \end{aligned}$$

Application to Nearly-Degenerate Problems

Energy Expansion in Nearly-Degenerate Cases

Expansion in Terms that Depend on $E_n^{(k)}$

Explicit Expansion

Basic Expressions

Hermiticity

Improved Forms

Orthogonality

Degeneracies

Near Degeneracies

By *near-degeneracies* we just mean cases where the perturbation expansions either do not con-

verge or, alternately, converge very slowly. To counter this, we will introduce a two-parameter expansion to our wave functions and energies

$$E_n = \sum_{i=0}^{\infty}\sum_{a=0}^{\infty} E_n^{(\lambda,a)}\,\lambda^{(i)}\,\epsilon^{(a)}$$
$$\left|n\right\rangle = \sum_{i=0}^{\infty}\sum_{a=0}^{\infty} \left|n^{(\lambda,a)}\right\rangle \lambda^{(i)}\,\epsilon^{(a)}$$

or alternately, for any order k in λ we have

$$E_n^{(k)} = \sum_{a=0}^{\infty} E_n^{(k,a)}\,\epsilon^{(a)}$$
$$\left|n^{(k)}\right\rangle = \sum_{a=0}^{\infty} \left|n^{(k,a)}\right\rangle \epsilon^{(a)}$$

which allows us to reexpress the perturbation theory equations as

$$\begin{aligned}\emptyset &= \sum_{i=0}^{k} \Delta \mathrm{H}_n^{(k-i)} \left|n^{(i)}\right\rangle \\ &= \sum_{m=0}^{M}\sum_{i=0}^{k}\sum_{a=0}^{m} \Delta \mathrm{H}_n^{(k-i,\ m-a)} \left|n^{(i,\ a)}\right\rangle\end{aligned}$$

which so far is no different from everything we have presented before. But what we can do is say for this to converge, we should be able to truncate at some maximal M. The idea with the non-convergence is that there is no such M.

will note that the energies come directly from our wavefunctions

$$E_n^{(k)} = \left\langle n^{(0)}\right| H^{(k)} \left|n^{(0)}\right\rangle + \sum_{i=1}^{k-1} \left\langle n^{(0)}\right| \Delta \mathrm{H}_n^{(k-i)} \left|n^{(i)}\right\rangle$$

and so if the *perturbation theory* is going to yield non-convergent expansions for the energies and wave functions

Expressing in Terms of Zero-Order State & Selection Rule Math

---

Perturbation Theory Hamiltonians

Background

In our representation of our Hamiltonian, we have

$$\mathrm{H}^{(1)} = \sum_{ijk} \frac{1}{2} \frac{\partial g_{ij}}{\partial Q_k} p_i Q_k p_j + \frac{1}{6} \frac{\partial^3 V}{\partial Q_i Q_j Q_k} Q_i Q_j Q_k$$

then if we ask for element of this

$$\begin{aligned}\langle n | \mathrm{H}^{(1)} | m \rangle &= \langle n | \sum_{ijk} \frac{1}{2} \frac{\partial g_{ij}}{\partial Q_k} p_i Q_k p_j + \frac{1}{6} \frac{\partial^3 V}{\partial Q_i Q_j Q_k} Q_i Q_j Q_k | m \rangle \\ &= \sum_{ijk} \frac{1}{2} \frac{\partial g_{ij}}{\partial Q_k} \langle n | p_i Q_k p_j | m \rangle + \frac{1}{6} \frac{\partial^3 V}{\partial Q_i Q_j Q_k} \langle n | Q_i Q_j Q_k | m \rangle\end{aligned}$$

we realize that we need to get representations of the operators pQp and QQQ

The form that these operator representations should take may not be entirely obvious.

A 2D Example

The first thing we should do is consider a 2-dimensional example, e.g. the operator xy. As it turns out, the representation for this in some general basis, $\{\phi\}$, unlike the case of something like x^2+y^2 is unpleasant since it cannot be cleanly separated into some combination of 1D representations.

However, if we set up a direct product basis, i.e. one such that

$$\phi_n = \phi_{n_x} \phi_{n_y}$$

this becomes much nicer

In this case we have

$$\begin{aligned}\langle n | xy | m \rangle &= \langle n_y | \langle n_x | xy | m_x \rangle | m_y \rangle \\ &= \langle n_x | x | m_x \rangle \langle n_y | y | m_y \rangle\end{aligned}$$

and so if we can set up representations for both x and y, we can take the direct product of those representations and get a total representation

What we *do* need to keep in mind with this is that it will take us from a 2D matrix to a 4D tensor, as we have two quantum numbers $(n_x\ n_y)$ for every n

Representation

What might this look like, then, for a product of harmonic oscillator wavefunction basis?

Let's assume we have N_x wavefunctions in x and N_y in y. This means the representations will be

$$X=\begin{pmatrix} 0 & \sqrt{1} & & & \\ \sqrt{1} & 0 & \sqrt{2} & & \\ & \sqrt{2} & \ddots & \ddots & \\ & & \ddots & & \sqrt{N_x} \\ & & & \sqrt{N_x} & 0 \end{pmatrix}$$

$$Y=\begin{pmatrix} 0 & \sqrt{1} & & & \\ \sqrt{1} & 0 & \sqrt{2} & & \\ & \sqrt{2} & \ddots & \ddots & \\ & & \ddots & & \sqrt{N_y} \\ & & & \sqrt{N_y} & 0 \end{pmatrix}$$

But then what happens when we have our total basis of size $N_x\, N_y$? Well we can assume that our basis functions are ordered like

$$\begin{aligned} \phi_1 &= \phi_{x=1}\, \phi_{y=1} \\ \phi_2 &= \phi_{x=2}\, \phi_{y=1} \\ &\vdots \\ \phi_{N_x} &= \phi_{x=N_x}\, \phi_{y=1} \\ \phi_{N_x+1} &= \phi_{x=1}\, \phi_{y=2} \\ &\vdots \\ \phi_{N_x\, N_y} &= \phi_{x=N_x}\, \phi_{y=N_y} \end{aligned}$$

and so when we look at our total array, it will look like

$$X\,Y=\begin{pmatrix} X_{11}\; Y_{11} & X_{12}\; Y_{11} & \dots & X_{1N_x}\, Y_{11} & X_{11}\; Y_{12} & \dots & X_{1N_x}\; Y_{1N_y} \\ X_{21}\; Y_{11} & X_{22}\; Y_{11} & & & & & \vdots \\ \vdots & & \ddots & \vdots & & & \\ X_{N_x1}\; Y_{11} & & \dots & X_{N_xN_x}\; Y_{11} & & & \\ X_{11}\; Y_{21} & & & \ddots & & & \\ \vdots & & & & & & \\ & & & & & & \\ X_{N_x1}\; Y_{N_y1} & & \dots & & & & X_{N_xN_x}\; Y_{N_yN_y} \end{pmatrix}$$

which we could rewrite as

$$X\,Y=\begin{pmatrix} X\,Y_{11} & X\,Y_{12} & \dots & X\,Y_{1N_y} \\ X\,Y_{21} & X\,Y_{22} & & \\ \vdots & & \ddots & \vdots \\ \mathrm{XY}_{N_y1} & & \dots & X\,Y_{N_yN_y} \end{pmatrix}$$

which is the Kronecker product of the matrices X and Y, or in more standard notation

$$XY = X \otimes Y$$

The ND case follows the same pattern, being

$$X = X_1 \otimes X_2 \otimes \ldots \otimes X_N$$

Representation of $H^{(1)}$

With all this in place, we can now think about how we will handle just the second part of $H^{(1)}$

$$\sum_{ijk} \frac{\partial^3 V}{\partial Q_i \, Q_j \, Q_k} \, Q_i \, Q_j \, Q_k$$

as we've shown previously, we can get a full tensor representation of the derivatives, i.e.

$$\left(V_{Q^3}\right)_{ijk} = \frac{\partial^3 V}{\partial Q_i \, Q_j \, Q_k}$$

This term has dimension (M, M, M) where M is the total number of coordinates in our system.

The question, then, is what to do with Q^3?

Here we need to decide how many quanta of excitation we will allow to be in each of our modes. We will let this be N_i. This means our basis elements look like

$$\phi_n = \prod_{i=1}^{M} \varphi_{n_i}$$

and we have a total basis size of

$$N = \prod_{i=1}^{M} N_i$$

and so we expect that $H^{(1)}$ should be have dimension (N, N)

Thus we expect to have representation of QQQ that has dimension (M, M, M, N, N), which can of course get memory intensive *very* quickly unless we can control the size of N or by computing individual elements of $H^{(1)}$, like

$$\begin{aligned} \left(H^{(1)}\right)_{n,m} &= \left(H^{(1)}\right)_{n_{x_1} \, n_{x_2} \ldots n_{x_M}, m_{x_1} \, m_{x_2} \ldots m_{x_M}} \\ &= V_{Q^3} \odot \left(Q^3_{\ldots \, \mathrm{nm}}\right) \end{aligned}$$

where the $\odot$ is telling us to contract the axes of V_{Q^3} with the first three axes of Q^3

But, saving that for another day, we're still left with the question of how to represent Q^3, given that for $M > 3$ the indices i, j, k don't cover the full set of combinations of axes.

In this case, we can think of this as

$$Q^3 = Q^3\ I^{(M-3)}$$

and so we end up having

$$Q^3 = I_{N_1} \otimes I_{N_2} \otimes \ldots \otimes Q_i \otimes I_{N_{i+1}} \otimes \ldots \otimes Q_j \otimes I_{N_{j+1}} \otimes \ldots\ Q_k \otimes I_{N_{k+1}} \otimes \ldots \mathrm{I}_{N_M}$$

which is a truly massive tensor

On the other hand, it is also an extremely *sparse* tensor, and so by using sparse matrix methods we can end up storing very little of it, if we decide that we *do* want to do a direct calculation of the tensor.

Explicit Form

There is also a mild complication that we haven't dealt with here, either, which is that sometimes we won't have three distinct indices. That is sometimes we'll have $Q_x\ Q_y\ Q_z$ but other times we might have $Q_x\ Q_y\ Q_x$. These terms will have different representations:

$$Q_x\ Q_y\ Q_z \Rightarrow\ Q_x \otimes Q_y \otimes Q_z$$
$$Q_x\ Q_y\ Q_x \Rightarrow {Q_x}^2 \otimes Q_y$$

where that ${Q_x}^2$ term is, as usual, a matrix power of the representation of Q_x, not a direct term wise multiplication.

This can also make indexing something of a challenge to think about. Let's consider three modes (we'll call them x, y, and z) and in each of these we put up to 1 quantum of excitation.

Then if we're asking for QQQ_{nm} we are really asking for this tensor

$$\mathrm{QQQ}_{nm} = \begin{pmatrix} \begin{pmatrix} (Q_x\ Q_x\ Q_x)_{nm} & (Q_x\ Q_x\ Q_y)_{nm} & (Q_x\ Q_x\ Q_z)_{nm} \\ (Q_x\ Q_y\ Q_x)_{nm} & (Q_x\ Q_y\ Q_y)_{nm} & (Q_x\ Q_y\ Q_z)_{nm} \\ (Q_x\ Q_z\ Q_x)_{nm} & (Q_x\ Q_z\ Q_y)_{nm} & (Q_x\ Q_z\ Q_z)_{nm} \end{pmatrix} \\ \begin{pmatrix} (Q_y\ Q_x\ Q_x)_{nm} & (Q_y\ Q_x\ Q_y)_{nm} & (Q_y\ Q_x\ Q_z)_{nm} \\ (Q_y\ Q_y\ Q_x)_{nm} & (Q_y\ Q_y\ Q_y)_{nm} & (Q_y\ Q_y\ Q_z)_{nm} \\ (Q_y\ Q_z\ Q_x)_{nm} & (Q_y\ Q_z\ Q_y)_{nm} & (Q_y\ Q_z\ Q_z)_{nm} \end{pmatrix} \\ \begin{pmatrix} (Q_z\ Q_x\ Q_x)_{nm} & (Q_z\ Q_x\ Q_y)_{nm} & (Q_z\ Q_x\ Q_z)_{nm} \\ (Q_z\ Q_y\ Q_x)_{nm} & (Q_z\ Q_y\ Q_y)_{nm} & (Q_z\ Q_y\ Q_z)_{nm} \\ (Q_z\ Q_z\ Q_x)_{nm} & (Q_z\ Q_z\ Q_y)_{nm} & (Q_z\ Q_z\ Q_z)_{nm} \end{pmatrix} \end{pmatrix}$$

$$=\begin{pmatrix}\begin{pmatrix} (Q_x{}^3)_{nm} & (Q_x{}^2\,Q_y)_{nm} & (Q_x{}^2\,Q_z)_{nm} \\ (Q_x{}^2\,Q_y)_{nm} & (Q_x\,Q_y{}^2)_{nm} & (Q_x\,Q_y\,Q_z)_{nm} \\ (Q_x{}^2\,Q_z)_{nm} & (Q_x\,Q_z\,Q_y)_{nm} & (Q_x\,Q_z{}^2)_{nm} \end{pmatrix} \\ \begin{pmatrix} (Q_x{}^2\,Q_y)_{nm} & (Q_x\,Q_y{}^2)_{nm} & (Q_y\,Q_x\,Q_z)_{nm} \\ (Q_x\,Q_y{}^2)_{nm} & (Q_y{}^3)_{nm} & (Q_y{}^2\,Q_z)_{nm} \\ (Q_y\,Q_z\,Q_x)_{nm} & (Q_y{}^2\,Q_z)_{nm} & (Q_y\,Q_z{}^2)_{nm} \end{pmatrix} \\ \begin{pmatrix} (Q_x{}^2\,Q_z)_{nm} & (Q_z\,Q_x\,Q_y)_{nm} & (Q_x\,Q_z{}^2)_{nm} \\ (Q_z\,Q_y\,Q_x)_{nm} & (Q_y{}^2\,Q_z)_{nm} & (Q_y\,Q_z{}^2)_{nm} \\ (Q_x\,Q_z{}^2)_{nm} & (Q_y\,Q_z{}^2)_{nm} & (Q_z{}^3)_{nm} \end{pmatrix} \end{pmatrix}$$

Then the question is: what is nm?

To answer this we need to think about how our total Hamiltonian is structured. A given element of this looks like

$$\langle n|\,\mathrm{H}\,|m\rangle=\langle n|\,\mathrm{T}\,|m\rangle+\langle n|\,\mathrm{V}\,|m\rangle$$

and keeping in mind that our basis is the direct product of the basis sets

$$\Phi_x=\{|0\rangle_x,\,|1\rangle_x\}$$
$$\Phi_y=\{|0\rangle_y,\,|1\rangle_y\}$$
$$\Phi_z=\{|0\rangle_z,\,|1\rangle_z\}$$

so our overall basis is

$$\Phi=\{|0\rangle_x|0\rangle_y|0\rangle_z,\,|1\rangle_x|0\rangle_y|0\rangle_z,\,|0\rangle_x|1\rangle_y|0\rangle_z,\,|0\rangle_x|0\rangle_y|1\rangle_z,$$
$$|1\rangle_x|1\rangle_y|0\rangle_z,\,|1\rangle_x|0\rangle_y|1\rangle_z,\,|0\rangle_x|1\rangle_y|1\rangle_z,\,|1\rangle_x|1\rangle_y|1\rangle_z,\}$$

and then n refers to one of these basis functions and m to another, i.e. (removing the redundant subscripts)

$$\Phi_n=|n_x\rangle\,|n_y\rangle\,|n_z\rangle$$

So overall

$$\mathrm{QQQ}_{nm} = \langle n_x| \langle n_y| \langle n_z| \begin{pmatrix} \begin{pmatrix} (Q_x{}^3) & (Q_x{}^2\, Q_y) & (Q_x{}^2\, Q_z) \\ (Q_x{}^2\, Q_y) & (Q_x\, Q_y{}^2) & (Q_x\, Q_y\, Q_z) \\ (Q_x{}^2\, Q_z) & (Q_x\, Q_z\, Q_y) & (Q_x\, Q_z{}^2) \end{pmatrix} \\ \begin{pmatrix} (Q_x{}^2\, Q_y) & (Q_x\, Q_y{}^2) & (Q_y\, Q_x\, Q_z) \\ (Q_x\, Q_y{}^2) & (Q_y{}^3) & (Q_y{}^2\, Q_z) \\ (Q_y\, Q_z\, Q_x) & (Q_y{}^2\, Q_z) & (Q_y\, Q_z{}^2) \end{pmatrix} \\ \begin{pmatrix} (Q_x{}^2\, Q_z) & (Q_z\, Q_x\, Q_y) & (Q_x\, Q_z{}^2) \\ (Q_z\, Q_y\, Q_x) & (Q_y{}^2\, Q_z) & (Q_y\, Q_z{}^2) \\ (Q_x\, Q_z{}^2) & (Q_y\, Q_z{}^2) & (Q_z{}^3) \end{pmatrix} \end{pmatrix} |m_x\rangle |m_y\rangle |m_z\rangle$$

this is a real pain to show in full, so we'll just focus on the three types of terms in the middle block

$$\begin{aligned}
\langle n_x| \langle n_y| \langle n_z|\, Q_y{}^3\, |m_x\rangle |m_y\rangle |m_z\rangle &= \langle n_y|\, Q_y{}^3\, |m_y\rangle \langle n_x| m_x\rangle \langle n_z| m_z\rangle \\
\langle n_x| \langle n_y| \langle n_z|\, Q_x{}^2\, Q_y\, |m_x\rangle |m_y\rangle |m_z\rangle &= \langle n_x|\, Q_x{}^2\, |m_x\rangle \langle n_y|\, Q_y\, |m_y\rangle \langle n_z| m_z\rangle \\
\langle n_x| \langle n_y| \langle n_z|\, Q_x\, Q_y{}^2\, |m_x\rangle |m_y\rangle |m_z\rangle &= \langle n_x|\, Q_x\, |m_x\rangle \langle n_y|\, Q_y{}^2\, |m_y\rangle \langle n_z| m_z\rangle \\
\langle n_x| \langle n_y| \langle n_z|\, Q_y{}^2\, Q_z\, |m_x\rangle |m_y\rangle |m_z\rangle &= \langle n_y|\, Q_y{}^2\, |m_y\rangle \langle n_z|\, Q_z\, |m_z\rangle \langle n_x| m_x\rangle \\
\langle n_x| \langle n_y| \langle n_z|\, Q_y\, Q_z{}^2\, |m_x\rangle |m_y\rangle |m_z\rangle &= \langle n_y|\, Q_y\, |m_y\rangle \langle n_z|\, Q_z{}^2\, |m_z\rangle \langle n_x| m_x\rangle \\
\langle n_x| \langle n_y| \langle n_z|\, Q_y\, Q_x\, Q_z\, |m_x\rangle |m_y\rangle |m_z\rangle &= \langle n_x|\, Q_x\, |m_x\rangle \langle n_y|\, Q_y\, |m_y\rangle \langle n_z|\, Q_z\, |m_z\rangle
\end{aligned}$$

and assuming that x, y, and z all have the same representation for Q this means the whole tensor is

$$\mathrm{QQQ}_{nm} = \begin{pmatrix} \begin{pmatrix} Q^3{}_{n_x m_x}\, \delta_{n_y m_y}\, \delta_{n_z m_z} & Q^2{}_{n_x m_x}\, Q_{n_y m_y}\, \delta_{n_z m_z} & Q^2{}_{n_x m_x}\, Q_{n_z m_z}\, \delta_{n_y m_y} \\ Q^2{}_{n_x m_x}\, Q_{n_y m_y}\, \delta_{n_z m_z} & Q_{n_x m_x}\, Q^2{}_{n_y m_y}\, \delta_{n_z m_z} & Q_{n_x m_x}\, Q_{n_y m_y}\, Q_{n_z m_z} \\ Q^2{}_{n_x m_x}\, Q_{n_z m_z}\, \delta_{n_y m_y} & Q_{n_x m_x}\, Q_{n_y m_y}\, Q_{n_z m_z} & Q_{n_x m_x}\, Q^2{}_{n_z m_z}\, \delta_{n_y m_y} \end{pmatrix} \\ \begin{pmatrix} Q^2{}_{n_x m_x}\, Q_{n_y m_y}\, \delta_{n_z m_z} & Q_{n_x m_x}\, Q^2{}_{n_y m_y}\, \delta_{n_z m_z} & Q_{n_x m_x}\, Q_{n_y m_y}\, Q_{n_z m_z} \\ Q_{n_x m_x}\, Q^2{}_{n_y m_y}\, \delta_{n_z m_z} & Q^3{}_{n_y m_y}\, \delta_{n_x m_x}\, \delta_{n_z m_z} & Q^2{}_{n_y m_y}\, Q_{n_z m_z}\, \delta_{n_x m_x} \\ Q_{n_x m_x}\, Q_{n_y m_y}\, Q_{n_z m_z} & Q^2{}_{n_y m_y}\, Q_{n_z m_z}\, \delta_{n_x m_x} & Q_{n_y m_y}\, Q^2{}_{n_z m_z}\, \delta_{n_x m_x} \end{pmatrix} \\ \begin{pmatrix} Q^2{}_{n_x m_x}\, Q_{n_z m_z}\, \delta_{n_y m_y} & Q_{n_x m_x}\, Q_{n_y m_y}\, Q_{n_z m_z} & Q_{n_x m_x}\, Q^2{}_{n_z m_z}\, \delta_{n_y m_y} \\ Q_{n_x m_x}\, Q_{n_y m_y}\, Q_{n_z m_z} & Q^2{}_{n_y m_y}\, Q_{n_z m_z}\, \delta_{n_x m_x} & Q_{n_y m_y}\, Q^2{}_{n_z m_z}\, \delta_{n_x m_x} \\ Q_{n_x m_x}\, Q^2{}_{n_z m_z}\, \delta_{n_y m_y} & Q_{n_y m_y}\, Q^2{}_{n_z m_z}\, \delta_{n_x m_x} & Q^3{}_{n_z m_z}\, \delta_{n_x m_x}\, \delta_{n_y m_y} \end{pmatrix} \end{pmatrix}$$

which looks imposing, but really ends up being quite simple. If, for instance, we have

$$\begin{aligned} n_x{=}1 \;\; m_x{=}0 \\ n_y{=}0 \;\; m_y{=}1 \end{aligned}$$

$$n_z=1 \quad m_z=1$$

based on ordering our basis as

$$\Phi=\Big\{|0\rangle_x|0\rangle_y|0\rangle_z,\ |1\rangle_x|0\rangle_y|0\rangle_z,\ |0\rangle_x|1\rangle_y|0\rangle_z,\ |0\rangle_x|0\rangle_y|1\rangle_z,$$
$$|1\rangle_x|1\rangle_y|0\rangle_z,\ |1\rangle_x|0\rangle_y|1\rangle_z,\ |0\rangle_x|1\rangle_y|1\rangle_z,\ |1\rangle_x|1\rangle_y|1\rangle_z,\Big\}$$

we get

$$\mathrm{QQQ}_{65}=\begin{pmatrix}\begin{pmatrix} {Q^3}_{10}\,\delta_{01}\,\delta_{11} & {Q^2}_{10}\,Q_{01}\,\delta_{11} & {Q^2}_{10}\,Q_{11}\,\delta_{01}\\ {Q^2}_{10}\,Q_{01}\,\delta_{11} & Q_{10}\,{Q^2}_{01}\,\delta_{11} & Q_{10}\,Q_{01}\,Q_{11}\\ {Q^2}_{10}\,Q_{11}\,\delta_{01} & Q_{10}\,Q_{01}\,Q_{11} & Q_{10}\,{Q^2}_{11}\,\delta_{01}\end{pmatrix}\\ \begin{pmatrix} {Q^2}_{10}\,Q_{01}\,\delta_{11} & Q_{10}\,{Q^2}_{01}\,\delta_{11} & Q_{10}\,Q_{01}\,Q_{11}\\ Q_{10}\,{Q^2}_{01}\,\delta_{11} & {Q^3}_{01}\,\delta_{10}\,\delta_{11} & {Q^2}_{01}\,Q_{11}\,\delta_{10}\\ Q_{10}\,Q_{01}\,Q_{11} & {Q^2}_{01}\,Q_{11}\,\delta_{10} & Q_{01}\,{Q^2}_{11}\,\delta_{10}\end{pmatrix}\\ \begin{pmatrix} {Q^2}_{10}\,Q_{11}\,\delta_{01} & Q_{10}\,Q_{01}\,Q_{11} & Q_{10}\,{Q^2}_{11}\,\delta_{01}\\ Q_{10}\,Q_{01}\,Q_{11} & {Q^2}_{01}\,Q_{11}\,\delta_{10} & Q_{01}\,{Q^2}_{11}\,\delta_{10}\\ Q_{10}\,{Q^2}_{11}\,\delta_{01} & Q_{01}\,{Q^2}_{11}\,\delta_{10} & {Q^3}_{11}\,\delta_{10}\,\delta_{01}\end{pmatrix}\end{pmatrix}$$

assuming we're working with harmonic oscillator wavefunctions, this gives us

$$Q=\begin{pmatrix} 0 & \frac{1}{\sqrt{2}} & 0\\ \frac{1}{\sqrt{2}} & 0 & 1\\ 0 & 1 & 0\end{pmatrix}\quad Q^2=\begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{\sqrt{2}}\\ 0 & \frac{3}{2} & 0\\ \frac{1}{\sqrt{2}} & 0 & \frac{5}{2}\end{pmatrix}\quad Q^3=\begin{pmatrix} 0 & \frac{3}{2\sqrt{2}} & 0\\ \frac{3}{2\sqrt{2}} & 0 & 3\\ 0 & 3 & 0\end{pmatrix}$$

and so

$$Q_{10}=\frac{1}{\sqrt{2}}\quad Q_{11}=0\quad Q_{01}=\frac{3}{\sqrt{2}}$$
$$Q^2{}_{10}=0\quad Q^2{}_{11}=\frac{3}{2}\quad Q^2{}_{01}=0$$
$$Q^3{}_{10}=\frac{3}{\sqrt{2}}\quad Q^3{}_{11}=0\quad Q^3{}_{01}=\frac{3}{\sqrt{2}}$$

and so

$$\mathrm{QQQ}_{65}=\begin{pmatrix}\begin{pmatrix}\frac{3}{2\sqrt{2}}\,\delta_{01}\,\delta_{11} & 0*\frac{1}{\sqrt{2}}\,\delta_{11} & 0*0\,\delta_{01}\\ 0*\frac{1}{\sqrt{2}}\,\delta_{11} & \frac{1}{\sqrt{2}}*0\,\delta_{11} & \frac{1}{\sqrt{2}}*\frac{1}{\sqrt{2}}*0\\ 0*0\,\delta_{01} & \frac{1}{\sqrt{2}}*\frac{1}{\sqrt{2}}*0 & \frac{1}{\sqrt{2}}*\frac{3}{2}\,\delta_{01}\end{pmatrix}\\ \begin{pmatrix}0*\frac{1}{\sqrt{2}}\,\delta_{11} & \frac{1}{\sqrt{2}}*0\,\delta_{11} & \frac{1}{\sqrt{2}}*\frac{1}{\sqrt{2}}*0\\ \frac{1}{\sqrt{2}}*0\,\delta_{11} & \frac{3}{2\sqrt{2}}\,\delta_{10}\,\delta_{11} & 0*0\,\delta_{10}\\ \frac{1}{\sqrt{2}}*\frac{1}{\sqrt{2}}*0 & 0*0\,\delta_{10} & \frac{1}{\sqrt{2}}\,\frac{3}{2}\,\delta_{10}\end{pmatrix}\\ \begin{pmatrix}0*0\,\delta_{01} & \frac{1}{\sqrt{2}}*\frac{1}{\sqrt{2}}*0 & \frac{1}{\sqrt{2}}*\frac{3}{2}\,\delta_{01}\\ \frac{1}{\sqrt{2}}*\frac{1}{\sqrt{2}}*0 & 0*0\,\delta_{10} & \frac{1}{\sqrt{2}}*\frac{3}{2}\,\delta_{10}\\ \frac{1}{\sqrt{2}}*\frac{3}{2}\,\delta_{01} & \frac{1}{\sqrt{2}}*\frac{3}{2}\,\delta_{10} & 0\,\delta_{10}\,\delta_{01}\end{pmatrix}\end{pmatrix}=\begin{pmatrix}\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix}\\ \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix}\\ \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix}\end{pmatrix}$$

And so looked like a very large tensor turns out to be identically 0. This is a pretty common phenomenon with these and if we are trying to be very computationally efficient we can use selection rules to guide which terms we actually need to consider.

Operator Tensor Symmetries

When we build our representations, one key thing we can take advantage of is the symmetry of the problem. In particular, for *unshared* indices, we should be able to symmetrize at will, so

$$p_i\ Q_j\ p_k = p_i\ p_k\ Q_j = p_k\ p_i\ Q_j$$
$$p_i\ Q_j\ p_i = p_i\ p_i\ Q_j$$

but when indices are shared we lose that flexibility, as we note that

$$p_i\ Q_i\ p_j \neq Q_i\ p_i\ p_j$$

this is effectively due to the lack of commutativity of these terms

This means that when we are determining if two terms are equivalent we need to ask ourselves one main thing: *have we transposed our terms?*

The issue is that this can be tough to figure out. Algorithmically we need to first know which of our terms can be treated as equivalent (e.g. Q and Q, but not Q and p). Then we check to see if

any of our terms have been swapped around.

Sample Code

This provides Mathematica code to obtain explicit symbolic forms of tensor elements

$$\left(p_i\ Q_j\ Q_k\ p_l\right)_{1111}=p_1\ Q_1\ Q_1\ p_1$$

productOperatorTensor

testQQ

Tests

State Orthogonality

We'll consider

$$\begin{aligned}\left\langle n\middle|\, m\right\rangle &= \sum_k \left\langle n\middle|\, m\right\rangle^{(k)} \\ &= \sum_k \sum_{i=0}^{k} \left\langle n^{(k-i)}\middle|\, m^{(i)}\right\rangle\end{aligned}$$

and by the recursive definition of $\left|n^{(i)}\right\rangle$ we have

$$\left\langle n\middle|\, m\right\rangle^{(k)} = \sum_{i=0}^{k} \sum_{s\in P(k-i)} \sum_{r\in P(i)} \left\langle n^{(0)}\right| \left(\prod_{s_a\in s} \Delta \mathrm{H}_n^{(s_a)}\, \Pi_n\right)\left(\prod_{r_b\in r} \Pi_m\, \Delta \mathrm{H}_m^{(r_b)}\right)\left|m^{(0)}\right\rangle$$

where $P(k-i)$ is the set of permutations of integer partitions of $k-i$

Our goal will to be to find rules that allow every pair of sequences (s,r) to be at least partially canceled by another sequence

To do this, we will need a bit of helpful notation, first we will say

$$\begin{aligned}S^{(k)} &= \Delta \mathrm{H}_n^{(s_1)}\, \Pi_n \ldots \Delta \mathrm{H}_n^{(s_k)}\, \Pi_n \\ R^{(j)} &= \Pi_m\, \Delta \mathrm{H}_m^{(r_j)} \ldots \Pi_m\, \Delta \mathrm{H}_m^{(r_1)}\end{aligned}$$

and then we'll define the "reduced" sequences

$$\begin{aligned}S_-^{(k)} &= \Delta \mathrm{H}_n^{(s_1)}\, \Pi_n \ldots \Delta \mathrm{H}_n^{(s_k)} \\ R_-^{(j)} &= \Delta \mathrm{H}_m^{(r_j)} \ldots \Pi_m\, \Delta \mathrm{H}_m^{(r_1)}\end{aligned}$$

we do this because we have two sequence rules that we can express in terms of these reduced sequences

First, any sequence $S^{(k)}$ unpaired with an $R^{(j)}$ can be expressed in terms of its reduced sequence and $\Delta\mathrm{E} = E_n^{(0)} - E_m^{(0)}$ and vice versa

$$\begin{aligned}\left\langle n^{(0)}\right| S^{(k)} \left|m^{(0)}\right\rangle &= \left\langle n^{(0)}\right| S_-^{(k)}\, \Pi_n \left|m^{(0)}\right\rangle \\ &= \frac{1}{\Delta\mathrm{E}} \left\langle n^{(0)}\right| S_-^{(k)} \left|m^{(0)}\right\rangle \\ \left\langle n^{(0)}\right| R^{(j)} \left|m^{(0)}\right\rangle &= \left\langle n^{(0)}\right| \Pi_m\, R_-^{(j)} \left|m^{(0)}\right\rangle \\ &= -\frac{1}{\Delta\mathrm{E}} \left\langle n^{(0)}\right| R_-^{(j)} \left|m^{(0)}\right\rangle\end{aligned}$$

Second, by properties of the resolvent, for any paired sequence, we have

$$\begin{aligned}\left\langle n^{(0)}\right| S^{(k)}\, R^{(j)} \left|m^{(0)}\right\rangle &= \left\langle n^{(0)}\right| S_-^{(k)}\, \Pi_n\, \Pi_m\, R_-^{(j)} \left|m^{(0)}\right\rangle \\ &= \frac{1}{\Delta\mathrm{E}} \left\langle n^{(0)}\right| S_-^{(k)} (\Pi_m - \Pi_n)\, R_-^{(j)} \left|m^{(0)}\right\rangle \\ &= \frac{1}{\Delta\mathrm{E}} \left(\left\langle n^{(0)}\right| S_-^{(k)}\, R^{(j)} \left|m^{(0)}\right\rangle - \left\langle n^{(0)}\right| S^{(k)}\, R_-^{(j)} \left|m^{(0)}\right\rangle\right)\end{aligned}$$

We can look at this now with an instructive example

$$\begin{aligned}{}^{1)}_{n}\, \Pi_n\, \Delta\mathrm{H}_n^{(2)}\, \Pi_n\, \Delta\mathrm{H}_n^{(1)}\, \Pi_n \left|m^{(0)}\right\rangle &= \frac{1}{\Delta\mathrm{E}} \left\langle n^{(0)}\right| \Delta\mathrm{H}_n^{(1)}\, \Pi_n\, \Delta\mathrm{H}_n^{(2)}\, \Pi_n\, \Delta\mathrm{H}_n^{(1)} \left|m^{(0)}\right\rangle \\ {}^{1)}_{,}\, \Pi_n\, \Delta\mathrm{H}_n^{(2)}\, \Pi_n\, \Pi_m\, \Delta\mathrm{H}_m^{(1)} \left|m^{(0)}\right\rangle &= \frac{1}{\Delta\mathrm{E}} \Big[\left\langle n^{(0)}\right| \Delta\mathrm{H}_n^{(1)}\, \Pi_n\, \Delta\mathrm{H}_n^{(2)}\, \Pi_m\, \Delta\mathrm{H}_m^{(1)} \left|m^{(0)}\right\rangle - \left\langle n^{(0)}\right| \Delta\mathrm{H}_n^{(1)}\, \Pi_n\, \Delta\mathrm{H}_n^{(2)}\, \Pi_n\, \Delta\mathrm{H}\end{aligned}$$

This is low-key a pain to look at, we we'll simplify notation once again and write

$$\begin{aligned}\Delta\mathrm{H}_\lambda^{(i)} &= \binom{i}{\lambda} \\ \Pi_\lambda &= \binom{}{\lambda} \\ H^{(i)} &= \binom{i}{}\end{aligned}$$

Giving us

$$\begin{aligned}\binom{1}{n}\binom{}{n}\binom{2}{n}\binom{}{n}\binom{1}{n}\binom{}{n} &= \frac{1}{\Delta\mathrm{E}} \binom{1}{n}\binom{}{n}\binom{2}{n}\binom{}{n}\binom{1}{n} \\ \binom{1}{n}\binom{}{n}\binom{2}{n}\binom{}{n}\binom{}{m}\binom{1}{m} &= \frac{1}{\Delta\mathrm{E}} \left[\binom{1}{n}\binom{}{n}\binom{2}{n}\binom{}{m}\binom{1}{m} - \binom{1}{n}\binom{}{n}\binom{2}{n}\binom{}{n}\binom{1}{m}\right]\end{aligned}$$

and we'll note that the term $\binom{1}{n}\binom{}{n}\binom{2}{n}\binom{}{n}\binom{1}{n}$ *almost* cancels $-\binom{1}{n}\binom{}{n}\binom{2}{n}\binom{}{n}\binom{1}{m}$, and indeed from this we have sequence composition rule

$$S^{(k)}\, \Delta\mathrm{H}_n^{(i)}\, R^{(j)} - S^{(k)}\, \Delta\mathrm{H}_m^{(i)}\, R^{(j)} = \left(E_n^{(i)} - E_m^{(i)}\right) S^{(k)}\, R^{(j)}$$

Why is this useful? Well we can write a reduced sequence as a product of a regular sequence and

some $\Delta H_\lambda^{(i)}$

$$S_-^{(k)} = S^{(k-i)}\,\Delta H_n^{(i)}$$

and so generically we have

$$\left\langle n^{(0)}\right| S^{(k)}\,R^{(j)}\left|m^{(0)}\right\rangle = \frac{1}{\Delta E}\left(\left\langle n^{(0)}\right| S^{(k-\lambda)}\,\Delta H_n^{(\lambda)}\,R^{(j)}\left|m^{(0)}\right\rangle - \left\langle n^{(0)}\right| S^{(k)}\,\Delta H_m^{(i)}\,R^{(j-i)}\left|m^{(0)}\right\rangle\right)$$

then we'll note that there is a corresponding term (for $S^{(k+i)}$, assuming $j>i$)

$$\left\langle n^{(0)}\right| S^{(k)}\,\Delta H_n^{(i)}\,\Pi_n R^{(j-i)}\left|m^{(0)}\right\rangle = \frac{1}{\Delta E}\left(\left\langle n^{(0)}\right| S^{(k)}\,\Delta H_n^{(i)}\,R^{(j-i)}\left|m^{(0)}\right\rangle - \left\langle n^{(0)}\right| S^{(k+i)}\,\Delta H_m^{(l)}\,R^{(j-l)}\left|m^{(0)}\right\rangle\right)$$

And so adding these two terms we have

$$S^{(k)}\,R^{(j)}\left|m^{(0)}\right\rangle + \left\langle n^{(0)}\right| S^{(k)}\,\Delta H_n^{(i)}\,\Pi_n R^{(j-i)}\left|m^{(0)}\right\rangle =$$
$$\frac{1}{\Delta E}\left[\left\langle n^{(0)}\right| S^{(k-\lambda)}\,\Delta H_n^{(\lambda)}\,R^{(j)}\left|m^{(0)}\right\rangle - \left\langle n^{(0)}\right| S^{(k+i)}\,\Delta H_m^{(l)}\,R^{(j-l)}\left|m^{(0)}\right\rangle + \left(E_n^{(i)} - E_m^{(i)}\right)\left\langle n^{(0)}\right| S^{(k)}\,R^{(j-i)}\left|n\right.$$

and importantly we can add on terms that will lead to a similar reduction for the λ and l terms as well, with the caveat that we will eventually run into edge terms like

$$\left\langle n^{(0)}\right| S^{(k+i)}\left|m^{(0)}\right\rangle = \frac{1}{\Delta E}\left\langle n^{(0)}\right| S^{(k)}\,\Delta H_n^{(i)}\left|m^{(0)}\right\rangle$$
$$\left\langle n^{(0)}\right| R^{(j+i)}\left|m^{(0)}\right\rangle = -\frac{1}{\Delta E}\left\langle n^{(0)}\right| \Delta H_m^{(i)}\,R^{(j)}\left|m^{(0)}\right\rangle$$

but even then, these will simply partially cancel with terms generated by other sequences

Next, we note that this occurs for every order i and every possible sequence $S^{(k)}$ and $R^{(j)}$, meaning we end up with

$$\left\langle n\middle|\,m\right\rangle^{(k)} = \sum_{j=1}^{k}\sum_{i=0}^{k-j}\left(E_n^{(j)} - E_m^{(j)}\right)\sum_{s\in P(k-i-j)}\sum_{r\in P(i)}\left\langle n^{(0)}\right|\left(\prod_{s_a\in s}\Delta H_n^{(s_a)}\,\Pi_n\right)\left(\prod_{r_b\in r}\Pi_m\,\Delta H_m^{(r_b)}\right)\left|m^{(0)}\right\rangle$$

but that inner loop is the same as $\left\langle n\middle|\,m\right\rangle^{(j)}$ meaning at the end of the day we have

$$\left\langle n\middle|\,m\right\rangle^{(k)} = \sum_{i=1}^{k}\left(E_n^{(i)} - E_m^{(i)}\right)\left\langle n\middle|\,m\right\rangle^{(k-i)}$$

and if $\left\langle n\middle|\,m\right\rangle^{(k-i)} = 0$ for $i>0$, inductively

$$\left\langle n\middle|\,m\right\rangle^{(k)} = 0$$

Operator Simplifications

$$\left\langle n\right| A\left|m\right\rangle^{(2)} = \left\langle n^{(1)}\right| A^{(0)}\left|m^{(1)}\right\rangle + \left\langle n^{(2)}\right| A^{(0)}\left|m^{(0)}\right\rangle + \left\langle n^{(0)}\right| A^{(0)}\left|m^{(2)}\right\rangle$$

$$= \left\langle n^{(1)} \right| A^{(0)} \left| m^{(1)} \right\rangle + \left\langle n^{(1)} \right| \Delta \mathrm{H}_1 \, A^{(0)} \left| m^{(0)} \right\rangle + c_{20} \left\langle n^{(0)} \right| A^{(0)} \left| m^{(0)} \right\rangle$$

We'll consider

$$\left\langle n \right| A \left| m \right\rangle^{(k)} = \sum_{i=0}^{k} \sum_{j=0}^{i} \sum_{s \in P(k-i-j)} \sum_{r \in P(i)} \left\langle n^{(0)} \right| \left(\prod_{s_a \in s} \Delta \mathrm{H}_n^{(s_a)} \, \Pi_n \right) A^{(j)} \left(\prod_{r_b \in r} \Pi_m \, \Delta \mathrm{H}_m^{(r_b)} \right) \left| m^{(0)} \right\rangle$$

which can be expanded out in terms of the sequences from above like

$$\begin{aligned} \left\langle n^{(0)} \right| S^{(k)} \, A^{(i)} \, R^{(j)} \left| m^{(0)} \right\rangle &= \left\langle n^{(0)} \right| S^{(k)} \, A^{(i)} \, \Pi_m \, \Delta \mathrm{H}_m^{(a)} \, R^{(j-a)} \left| m^{(0)} \right\rangle \\ &= \left\langle n^{(0)} \right| S^{(k-b)} \, \Delta \mathrm{H}_n^{(b)} \, \Pi_n \, A^{(i)} \, R^{(j)} \left| m^{(0)} \right\rangle \end{aligned}$$

and we can pair up these sequences like

$$\left\langle n^{(0)} \right| S^{(k)} \, A^{(i)} \, \Pi_m \, \Delta \mathrm{H}_m^{(a)} \, R^{(j-a)} \left| m^{(0)} \right\rangle + \left\langle n^{(0)} \right| S^{(k)} \, \Delta \mathrm{H}_n^{(a)} \, \Pi_n \, A^{(i)} \, R^{(j-a)} \left| m^{(0)} \right\rangle$$

$$\mathrm{H}_n^{(a)} \, \Pi_n \, A^{(i)} + A^{(i)} \, \Pi_m \, \Delta \mathrm{H}_m^{(a)} = H^{(a)} \, \Pi_n \, A^{(i)} + A^{(i)} \, \Pi_m \, H^{(a)} - \left(E_n^{(a)} \, \Pi_n \, A^{(i)} + E_m^{(a)} \, A^{(i)} \, \Pi_m \right)$$

$$\begin{aligned} H^{(a)} \, \Pi_n \, A^{(i)} + A^{(i)} \, \Pi_m \, H^{(a)} &= \sum_k \left(\frac{H^{(a)} \left| k^{(0)} \right\rangle \left\langle k^{(0)} \right| A^{(i)}}{E_n - E_k} + \frac{A^{(i)} \left| k^{(0)} \right\rangle \left\langle k^{(0)} \right| H^{(a)}}{E_m - E_k} \right) \\ &= \sum_k \left(\frac{(V_a)_{xk} \, (A_a)_{ky} \, Q^a \left| k^{(0)} \right\rangle \left\langle k^{(0)} \right| Q^a}{E_n - E_k} + \frac{(A_a)_{nk} \, (V_a)_{km} \, Q^a \left| k^{(0)} \right\rangle \left\langle k^{(0)} \right| Q^a}{E_m - E_k} \right) \\ &= \sum_k \left(\frac{(V_a)_{xk} \, (A_a)_{ky}}{E_n - E_k} + \frac{(A_a)_{xk} \, (V_a)_{ky}}{E_m - E_k} \right) Q^a \left| k^{(0)} \right\rangle \left\langle k^{(0)} \right| Q^a \\ &= \sum_k \left(\frac{(V_a)_{xk} \, (A_a)_{ky} \, (E_m - E_k)}{(E_n - E_k) \, (E_m - E_k)} + \frac{(E_n - E_k) \, (A_a)_{xk} \, (V_a)_{ky}}{(E_n - E_k) \, (E_m - E_k)} \right) Q^a \left| k^{(0)} \right\rangle \left\langle k^{(0)} \right| Q \end{aligned}$$

Next, assuming $H^{(a)}$ and $A^{(i)}$ *operate on states in a similar manner* (that is, they are both say 3rd order in the raising and lowering operators), then...IDK they'll feed the same states into Π_m or something and so the same selection rules will apply???

We'll assume specifically for the moment that

$$\begin{aligned} H^{(a)} &= V_a \odot Q_a \\ A^{(i)} &= A_a \odot Q_a \end{aligned}$$

then

$$H^{(a)} \, \Pi_n \, A^{(i)} + A^{(i)} \, \Pi_m \, H^{(a)} = V_a \odot \left(\frac{Q_a}{E_n - E_m} \right)$$

Therefore, every sequence will have a set of partner sequences given by different values of a and b

$$\left\{ \left\langle n^{(0)} \right| S^{(k)} \, A^{(i)} \, R^{(j)} \left| m^{(0)} \right\rangle, \left\langle n^{(0)} \right| S^{(k-b)} \, A^{(i+b)} \, R^{(j)} \left| m^{(0)} \right\rangle, \left\langle n^{(0)} \right| S^{(k)} \, A^{(i+a)} \, R^{(j-a)} \left| m^{(0)} \right\rangle \right\}$$

and in particular we will have terms that look like

$$\left\langle n^{(0)}\right| S^{(k)}\left(A^{(i)}\,\Pi_m\,\Delta \mathrm{H}_m^{(a)} \;+\; A^{(i+a)}\right) R^{(j-a)}\left|m^{(0)}\right\rangle$$

and notably, if

where

$$\begin{aligned}\left\langle n^{(0)}\right| S^{(k)}\,R^{(j)}\left|m^{(0)}\right\rangle &= \left\langle n^{(0)}\right| S_-^{(k)}\,\Pi_n\,\Pi_m\,R_-^{(j)}\left|m^{(0)}\right\rangle \\ &= \frac{1}{\Delta \mathrm{E}}\left\langle n^{(0)}\right| S_-^{(k)}(\Pi_m - \Pi_n)\,R_-^{(j)}\left|m^{(0)}\right\rangle \\ &= \frac{1}{\Delta \mathrm{E}}\left(\left\langle n^{(0)}\right| S_-^{(k)}\,R^{(j)}\left|m^{(0)}\right\rangle - \left\langle n^{(0)}\right| S^{(k)}\,R_-^{(j)}\left|m^{(0)}\right\rangle\right)\end{aligned}$$

Diagonal Terms

$$\begin{aligned}{}_n H^{(0)}\,\Pi_m\,\Delta \mathrm{H}^{(1)}\left|m^{(0)}\right\rangle &= \left\langle n^{(0)}\right| H^{(1)}\,\Pi_n\,H^{(0)}\,\Pi_m\,H^{(1)}\left|m^{(0)}\right\rangle \\ &= \left\langle n^{(0)}\right| (V_3)_{\mathrm{xyz}} \odot \left(Q^3\right)_{\mathrm{xyz}}\left(\frac{1}{W_v \odot (\Delta_3)_{v\mathrm{xyz}}}\right) H^{(0)}\,(V_3)_{\mathrm{xyz}} \odot \left(Q^3\right)_{\mathrm{xyz}}\left(\frac{1}{W_v \odot (\Delta_3)_{v\mathrm{xyz}}}\right)\Pi\end{aligned}$$

Note that every path we do on the left has a corresponding path on the right and all of those are treated *identically* until we get to $H^{(0)}$, where we filter by state, which allows us to filter out by the *admissible* paths, which are the ones that account for the difference in quantum numbers between n and m

If, however, we want to transform entire spaces, i.e. we want the space of all m that differ from n by two quantum numbers, we can do this somewhat differently, as we know the transform $H^{(1)}$

We can get a bit further with diagonal operators

$$\begin{aligned}\left\langle n^{(0)}\right| S_-^{(k)}\,\Pi_n\,H^{(0)}\,\Pi_m\,R_-^{(j)}\left|m^{(0)}\right\rangle &= \left\langle n^{(0)}\right| S_-^{(k)}\,\Pi_n\,\Pi_m\,H^{(0)}\,R_-^{(j)}\left|m^{(0)}\right\rangle \\ &= \frac{1}{\Delta \mathrm{E}}\left[\left\langle n^{(0)}\right| S_-^{(k)}\,\Pi_n\,H^{(0)}\,R_-^{(j)}\left|m^{(0)}\right\rangle - \left\langle n^{(0)}\right| S_-^{(k)}\,\Pi_m\,H^{(0)}\,R_-^{(j)}\left|m^{(0)}\right\rangle\right] \\ &= \frac{1}{\Delta \mathrm{E}}\left[\left\langle n^{(0)}\right| S^{(k)}\,H^{(0)}\,R_-^{(j)}\left|m^{(0)}\right\rangle - \left\langle n^{(0)}\right| S_-^{(k)}\,H^{(0)}\,R^{(j)}\left|m^{(0)}\right\rangle\right]\end{aligned}$$

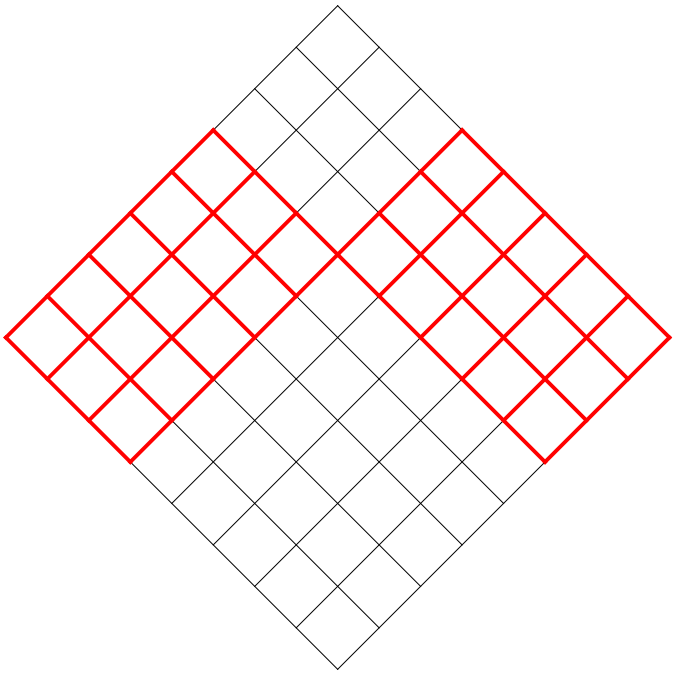
Lattice Paths / Raising-Lowering Relations

We can encode any string of raising/lowering operations as a lattice paths on a diagonal grid, for instance, here are the paths of length 16 that have value 6 after 8 steps

In[2228]:=

```
quickBlock[8, 2]
```

Out[2228]=



For any two blocks that meet at a point, we can express the total polynomial from the combination of those paths as the product of two polynomial terms, e.g. for the block above the total polynomial for those paths is $K_3^5(n)\,K_3^5(n)$, where K_b^a indicates that we have gone up a steps and down b steps

Naturally, from this we see that to get the total polynomial over these 16 steps, we can add up the polynomials induced by each point in the center (and in fact we can add up the polynomials induced by *any* column), giving us

$$\begin{aligned} K_8^8 &= K_0^8(n)\,K_8^0(n) + K_1^7(n)\,K_7^1(n) + K_2^6(n)\,K_6^2(n) + \ldots \\ &= \sum_{i=0}^{8} K_i^{8-i}(n)\,K_{8-i}^i(n) \end{aligned}$$

We can show, moreover, that

$$K_b^a(n) = \left(\sum_{i=0}^{|a-b|/2} c_i^{(a,b)}\, n^i\right) \begin{cases} \prod_{j=1}^{a-b} \sqrt{n+j} & a \geq b \\ \prod_{j=0}^{b-a-1} \sqrt{n-j} & a < b \end{cases}$$

can be written more nicely as

$$K_b^a(n) = P_b^a(n)\, S_b^a(n)$$

where P_b^a is a polynomial of degree $|a-b|/2$ and $S_b^a(n)$ is a product of $|a-b|$ square roots

Alternate Recursion

Action-Based Evaluation of P_b^a

Generic Evaluation of P_b^a

Best Formula

$$P_b^a(n) = \binom{a+b}{b} \sum_{\omega=0}^{b} \frac{1}{2^\omega}\, \omega! \binom{b}{\omega}\binom{a}{\omega} (n)_{b-\omega}$$

We can make a combinatoric model of this where $\binom{a+b}{b}$ is the total number of paths where we go up a steps and down b steps, $\omega!\binom{b}{\omega}\binom{a}{\omega}$ is a falling factorial connection coefficient which correspond to the numbers of ways to pull together ω raising/lowering operations, and the $1/2^\omega$ is keeping us from over-counting as we would naturally double-count (and so recursively multiply) for each of the ω choices

Reverse Direction

We know that for $b > a$

$$(P_-)_b^a(n) = P_a^b(n - (a-b))$$

where we simply switched the roles of a and b (so where b was originally the number of lowering operations, now it's the number of raising operations) and so if $b > a$

$$\begin{aligned} P_b^a(n) &= \binom{a+b}{b} \sum_{\omega=0}^{a} \frac{1}{2^\omega}\, \omega! \binom{b}{\omega}\binom{a}{\omega} (n-(a-b))_{a-\omega} \\ &= \binom{a+b}{b} \sum_{\omega=0}^{a} \frac{1}{2^\omega}\, \omega! \binom{b}{\omega}\binom{a}{\omega} \sum_{j=0}^{a-\omega} \binom{b-a}{\omega} (b-a)_{[a-\omega-j]}\, (n)_j \end{aligned}$$

Power Series Equivalent

For computational convenience it's nice to maintain a power-series form so we'll expand in the Stirling numbers as

In[225]:=
```
polyGenDirect[5, 2] // Expand
```

Out[225]=
```
105 + 84 n + 21 n^2
```

In[224]:=

```
polyGen[5, 2] // Expand
```

Out[224]=

$$105 + 84\,\underset{\cdot}{n} + 21\,\underset{\cdot}{n}^2$$

In[220]:=

```
polyGenDirect[a_, b_] := Binomial[a + b, a] * Sum[
   w! / 2^w * Binomial[b, w] *
     Binomial[a, w] * FactorialPower[n, b - w] // FunctionExpand,
   {w, 0, b}
  ]
```

$$\begin{aligned}
P_b^a(n) &= \binom{a+b}{b} \sum_{\omega=0}^{b} \frac{1}{2^\omega}\, \omega! \binom{b}{\omega}\binom{a}{\omega} (n)_{b-\omega} \\
&= \binom{a+b}{b} \sum_{\omega=0}^{b} \frac{1}{2^\omega}\, \omega! \binom{b}{\omega}\binom{a}{\omega} \sum_{j=0}^{b-\omega} S_1(b-\omega,\, j)\, n^j \\
&= \binom{a+b}{b} \sum_{j=0}^{b} \sum_{\omega=0}^{b-j} \frac{1}{2^\omega}\, \omega! \binom{b}{\omega}\binom{a}{\omega} S_1(b-\omega,\, j)\, n^j \\
&= \binom{a+b}{b} \sum_{j=0}^{b} \left(\sum_{\omega=0}^{b-j} \frac{1}{2^\omega}\, \omega! \binom{b}{\omega}\binom{a}{\omega} S_1(b-\omega,\, j) \right) n^j
\end{aligned}$$

```
polyGenCoeffs[a_, b_] := Binomial[a + b, a] Table[
   Sum[
    w! / 2^w * Binomial[b, w] * Binomial[a, w] * StirlingS1[b - w, j],
    {w, 0, b - j}
   ],
   {j, 0, b}
  ]
```

Setup

We can also note that this yields some nice recursions, in particular, assuming $a > 1$

$$\begin{aligned}
K_1^a &= K_0^a\, K_1^0(n+a) \,+\, K_1^{a-1}\, K_0^1(n+a-1) \\
&= (n+a)\, S_0^{a-1}(n) \,+\, P_1^{a-1}(n)\, S_1^{a-1}(n)\, \sqrt{n+a} \\
&= \left((n+a) \,+\, P_1^{a-1}\right) S_0^{a-1}(n) \\
&= \left((n+a) \,+\, P_1^{a-1}\right) S_1^{a}(n) \\
\Rightarrow\; P_1^a &= (n+a) \,+\, P_1^{a-1} \\
K_2^a &= K_1^a \sqrt{n+a-1} \,+\, K_2^{a-1}\, K_0^1(n+a-3) \\
&= (n+a-1)\, P_1^a\, S_0^{a-2}(n) \,+ P_2^{a-1} S_2^{a-1}\, \sqrt{n+a-2} \\
&= \left((n+a-1)\, P_1^a \,+ P_2^{a-1}\right) S_0^{a-2}(n)
\end{aligned}$$

$$\Rightarrow P_2^a = (n+a-1)\, P_1^a + P_2^{a-1}$$

And now (noting that $P_0^a = 1$) we can conjecture that for $k < a$

$$P_k^a = (n+a+1-k)\, P_{k-1}^a + P_k^{a-1}$$

which is in retrospect pretty obviously true

We can potentially make this better, by expanding out these recursions, but first we'll need an expression for $k > a$, we start again with

$$\begin{aligned} K_a^1 &= \sqrt{n-a+1}\; S_a^0 + P_{a-1}^1\, S_{a-1}^1\, \sqrt{n-a+2} \\ &= (n-a+1)\, S_a^1 + P_{a-1}^1\, S_a^1 \\ \Rightarrow P_a^1 &= (n-a+1) + P_{a-1}^1 \end{aligned}$$

and we find

$$P_a^k = (n-a+k)\, P_a^{k-1} + P_{a-1}^k$$

and finally

$$P_k^k = (n+1)\, P_{k-1}^k + (n)\, P_k^{k-1}$$

so now we can start building our recursions, starting with P_1

$$\begin{aligned} P_1^a &= (n+a) + P_{a-1}^1 \\ \Rightarrow P_1^a &= \sum_{i=0}^{a} (n+i) \\ &= \frac{1}{2}\,(1+a)\,(2n+a) \\ P_a^1 &= (n-(a-1)) + P_1^a \\ \Rightarrow P_a^1 &= \sum_{i=-1}^{a-1} (n-i) \\ &= \frac{1}{2}\,(1+a)\,(2(n+1)-a) \end{aligned}$$

And now for P_2 we get

$$\begin{aligned} \Rightarrow P_2^a &= (n+a-1)\left(\frac{1}{2}\,(a+1)\,(a+2n)\right) + P_2^{a-1} \\ &= (n+a-1)\left(\frac{1}{2}\,(a+1)\,(a+2n)\right) + (n+a-2)\left(\frac{1}{2}\,(a)\,(a-1+2n)\right) + P_2^{a-2} \\ &= \sum_{i=0}^{a-3} \frac{1}{2}\,(a-i+1)\,(a-i+2n)\,(a-i+n-1) + P_2^2 \\ P_2^2 &= (n+1)\,\frac{1}{2}\,(a-(a-2)+1)\,(a-(a-2)+2n) + (n)\,\frac{1}{2}\,(1+2)\,(2(n+1)-2) \end{aligned}$$

$$= \frac{1}{2}\,(a-i+1)\,(a-i+2n)\,(n+(a-i)-1)\Big|_{i=a-2} + \frac{1}{2}\,((a-i)+1+1)\,(2n+(a-i)-1)\,(n+(a-i)-1)\Big|_{i=a-1}$$

$$P_2^2 = \sum_{i=a-2}^{a-1} \frac{1}{2}\,(a-i+1)\,(a-i+2n)\,(a-i+n-1) + \frac{1}{2}\,n(2n-2)$$

$$= \sum_{i=a-2}^{a} \frac{1}{2}\,(a-i+1)\,(a-i+2n)\,(a-i+n-1)$$

$$P_2^a = \sum_{i=0}^{a} \frac{1}{2}\,(a-i+1)\,(a-i+2n)\,(a-i+n-1)$$

$$= \sum_{i=0}^{a} \frac{1}{2}\,(i+1)\,(i+2n)\,(i+n-1)$$

$$= \frac{1}{8}\,(1+a)\,(2+a)\left(a^2+a(-1+4n)+4\,(-1+n)n\right)$$

Finally, for P_3 we have

$$P_3^a = (n+a-2)\,P_2^a + (n+(a-1)-2)\,P_2^{a-1} + \ldots + (n+(a-(a-4))-2)\,P_2^4 + P_3^3$$

Generically, we can generate these as polynomial expressions from

$$P_b^a = \sum_{i=0}^{a} (n+i-(b-1))\,P_{b-1}^i(n)$$

where $\tilde{P}_b^a(n)$ is a degree b polynomial in n and a $2b$ degree polynomial in a, given by

$$P_b^a(n) = \sum_{k=0}^{b} \sum_{l=0}^{2b-k} c_{kl}^{(b)}\,a^l n^k$$

Note that for the terms of degree k in n, we end up with a $2b-k$ degree polynomial in a, i.e. the sum of the polynomial degrees in a and n are equal to $2b$, so we'll reduce this one step further to

$$P_b^a(n) = \sum_{k=0}^{b} Q_k^{(b)}(a)n^k$$

Expanding this out, we get

$$P_b^a = \sum_{i=0}^{a} (n+i-(b-1)) \sum_{k=0}^{b-1} Q_k^{(b-1)}(i)n^k$$

$$= \sum_{i=0}^{a} (n+i-(b-1)) \sum_{k=0}^{b-1} Q_k^{(b-1)}(i)n^k$$

$$= \sum_{i=0}^{a} \sum_{k=0}^{b-1} (i-b+1)\,Q_k^{(b-1)}(i)\,n^k + Q_k^{(b-1)}(i)\,n^{k+1}$$

$$= \sum_{i=0}^{a} \sum_{k=0}^{b} \left((i-b+1)\, Q_k^{(b-1)}(i) \;+\; Q_{k-1}^{(b-1)}(i) \right) n^k$$

In[200]:=

```
Clear[polyGen];
polyGen[0] = 1;
polyGen[1] = 1/2 (1 + a) (a + 2 n);

polyGen[b_] := polyGen[b] = FullSimplify[Sum[(n + a - b + 1) polyGen[b - 1] /. i -> a, {a, 0, i}]];

reducedPolyGen[b_] := polyGen[b] 2^b / Binomial[a + b, b] // Simplify;
polyGen[a_, b_] := If[a == 0 || b == 0,
   1,
   If[b > a,
    polyGenNeg[a] /. a -> b,
    polyGen[b] /. a -> a
   ]
  ];
polyGen[a_, b_, shift_] :=
  polyGen[a, b] /. n -> n + shift;
polySqrtGen[a_, b_] :=
  polyGen[a, b] * If[b > a,
    Product[Sqrt[n - i], {i, 0, b - a - 1}],
    Product[Sqrt[n + i], {i, 1, a - b}]
   ];
polySqrtGen[a_, b_, shift_] :=
  polySqrtGen[a, b] /. n -> n + shift;
```

Cursed Attempt with the Bernoulli Numbers and Explicit Factorization

Successful Attempt with Stirling Numbers over the Falling Factorials

Final Checks

None Of This Matters Any More

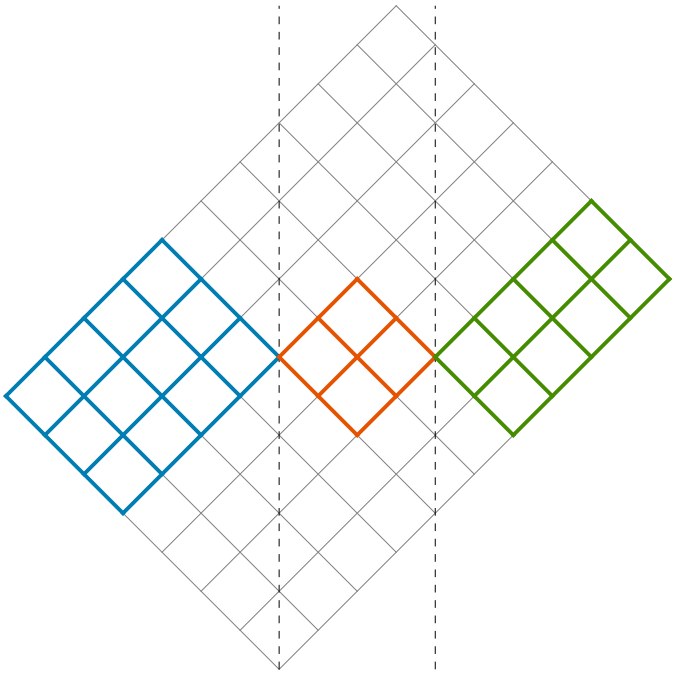
Products

We will have things like

In[882]:=

```
permutationOperatorPathDiagram[{4, 3}, {2, 2}, {4, 2}]
```

Out[882]=



and we want to write out the left-to-right polynomial in reverse order as (I think)

$$K_2^4(n+1)\, K_2^2(n+1)\, K_3^4(n)$$

Examples

Intuitive Understanding of S-products

Recursive Construction of K-product

When constructing the set of terms like

$$K_{b_1}^{a_1}(n)\, K_{b_2}^{a_2}(n+(a_1-b_2))\ldots\, K_{b_N}^{a_N}(n+k-(a_N-b_N)) =$$
$$P_{b_1 b_2\ldots}^{a_1 a_2\cdots}(n)\; S_{b_1}^{a_1}(n)\, S_{b_2}^{a_2}(n+(a_1-b_2))\ldots\, S_{b_N}^{a_N}(n+k-(a_N-b_N))$$

it ends up being fastest to do this recursively, where we'll build off of our processes for building the possible paths

In that process, we built up the possible K products one by one, adding on newer terms at the end, subject to the constraint that we will eventually end up at the appropriate change in quanta, as so with N terms that change quanta by k, we have

$$S_{b_1}^{a_1}(n)\, S_{b_2}^{a_2}(n+(a_1-b_2))\ldots\, S_{b_N}^{a_N}(n+k-(a_N-b_N)) = R_{b_1 b_2\ldots}^{a_1 a_2\cdots}(n)\, S_0^k(n)$$

where this remainder term looks like

$$R^{a_1 a_2 \cdots}_{b_1 b_2 \ldots}(n) = \prod_{i=-(b_T-1)}^{a_T} (n+i)^{p_i}$$

where p_i is a sequence of powers determined by the turns in the sequence of raising and lowering operations

This means to add on another term, we check where we currently are (i.e. k) and then if we are heading *towards* zero we add on as many terms as steps we have taken, in practice, this means

$$R^{a_1 a_2 \cdots}_{b_1 b_2 \ldots}(n)\, S^{a_{N+1}}_{b_{N+1}}(n+k) = R^{a_1 a_2 \cdots}_{b_1 b_2 \ldots}(n) \begin{cases} 1 & \text{sign}(k) = \text{sign}(a_{N+1} - b_{N+1}) \\ \left(S^{a_{N+1}}_{b_{N+1}}(n+k)\right)^2 & \text{else} \end{cases}$$

then we can note that if $a_{N+1} > b_{N+1}$

$$\begin{aligned} \left(S^{a_{N+1}}_{b_{N+1}}(n+k)\right)^2 &= (n+k)^{[k_{N+1}]} \\ &= (n + k + k_{N+1})_{[k_{N+1}]} \\ &= \sum_{j=0}^{k_{N+1}} \binom{k_{N+1}}{j} (k + k_{N+1})_{[k_{N+1} - j]}\, (n)_{[j]} \end{aligned}$$

and so we can take quick convolutions using FFTs in the basis of ... ah shit no... products of basis functions don't work that nicely... probably still best to expand everything as

$$\begin{aligned} \left(S^{a_{N+1}}_{b_{N+1}}(n+k)\right)^2 &= (n+k)^{[k_{N+1}]} \\ &= \sum_{j=0}^{k_{N+1}} (-1)^j S_1(k_{N+1}, j)\, (n+k)^j \\ &= \sum_{j=0}^{k_{N+1}} \sum_{l=0}^{j} \binom{j}{l} k^{j-l} (-1)^j S_1(k_{N+1}, j)\, n^l \\ &= \sum_{l=0}^{k_{N+1}} \left(\sum_{j=l}^{k_{N+1}} \binom{j}{l} k^{j-l} (-1)^j S_1(k_{N+1}, j) \right) n^l \end{aligned}$$

Falling / Rising Poly Examples

In[161]:=

```
With[{k = 2, a = 3, b = 1},
  Pochhammer[n + k, a - b]
 ] // Expand
```

Out[161]=

$$6 + 5\,n + n^2$$

In[164]:=
```
With[{k = 2, d = 3 - 1},
  Sum[(-1)^j * StirlingS1[d, j] * (n + k)^j, {j, 0, d}]
 ] // Expand
```
Out[164]=
$$6 + 5\,n + n^2$$

In[178]:=
```
With[{k = 2, d = 3 - 1},
  Sum[
   Sum[(-1)^l * StirlingS1[d, l] * Binomial[l, j] * k^(l - j), {l, j, d}] * n^j,
   {j, 0, d}
  ]
 ] // Expand
```
Out[178]=
$$6 + 5\,n + n^2$$

In[184]:=
```
With[{k = 1, b = 6, a = 1},
   FactorialPower[n + k, b - a]
  ] // FunctionExpand // Expand
```
Out[184]=
$$-6\,n + 5\,n^2 + 5\,n^3 - 5\,n^4 + n^5$$

In[185]:=
```
With[{k = 1, d = 6 - 1},
  Sum[
   Sum[StirlingS1[d, l] * Binomial[l, j] * k^(l - j), {l, j, d}] * n^j,
   {j, 0, d}
  ]
 ] // Expand
```
Out[185]=
$$-6\,n + 5\,n^2 + 5\,n^3 - 5\,n^4 + n^5$$

Exploration

Experiments

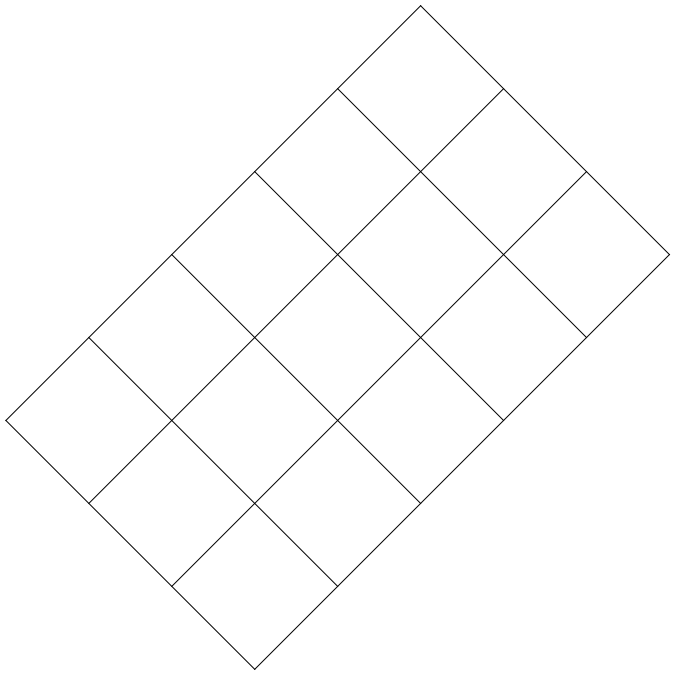
Numbers of Paths

Another key feature here is how *many* paths are there when we change quanta by k, which looking at one of our diagrams

In[3072]:=

```
permutationPathDiagram[5, 3]
```

Out[3072]=



is pretty clearly given by

$$N(a,\ b) = N(a{-}1,\ b) + N(a,\ b{-}1)$$

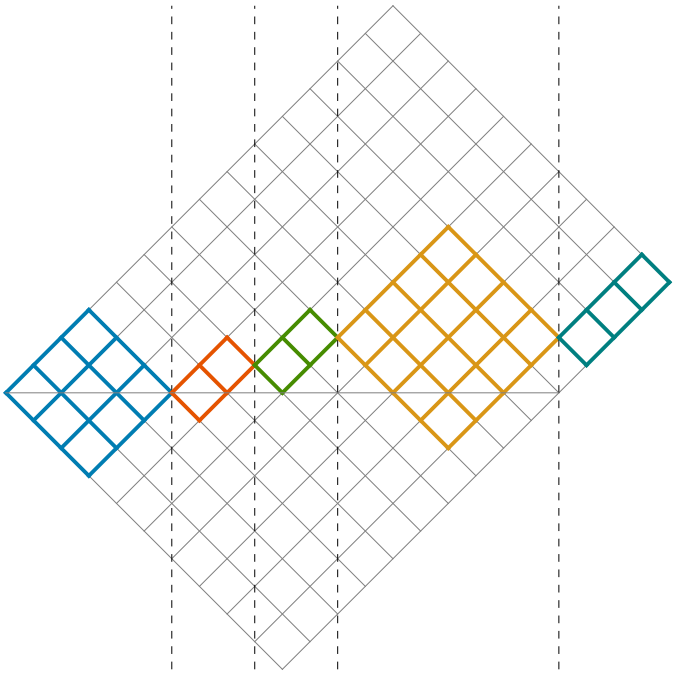
which we can recognize pretty quickly as the binomial coefficient $\binom{a+b}{b}$ which should be unsurprising as we are essentially trying to fit b lowering operations into $a+b$ steps

And so now when we enumerate possible paths this provides an easy termination metric, for example taking the path

In[3183]:=

```
permutationOperatorPathDiagram[{3, 3}, {2, 1}, {2, 1}, {4, 4}, {3, 1}]
```

Out[3183]=



In[3186]:=

```
enumeratePossibleChangePaths[{6, 3, 3, 8, 4}, 4] // Length // AbsoluteTiming
```

Out[3186]=

```
{0.003847, 440}
```

In[3202]:=

```
Reduce[((2 a1 - s1) - k) ≤ sT, a1]
```

Out[3202]=

$$(k \mid s1 \mid sT) \in \mathbb{R} \;\&\&\; a1 \leq \frac{1}{2}\,(k + s1 + sT)$$

In[3203]:=

```
Reduce[(k - (2 a1 - s1)) ≤ sT, a1]
```

Out[3203]=

$$(k \mid s1 \mid sT) \in \mathbb{R} \;\&\&\; a1 \geq \frac{1}{2}\,(k + s1 - sT)$$

$$\frac{1}{2}\,(k + s1 - sT)$$

In[3205]:=

```
1/2 (k + s1 - sT) /. {
  k → 4,
  s1 → 3,
  sT → 12
 }
```

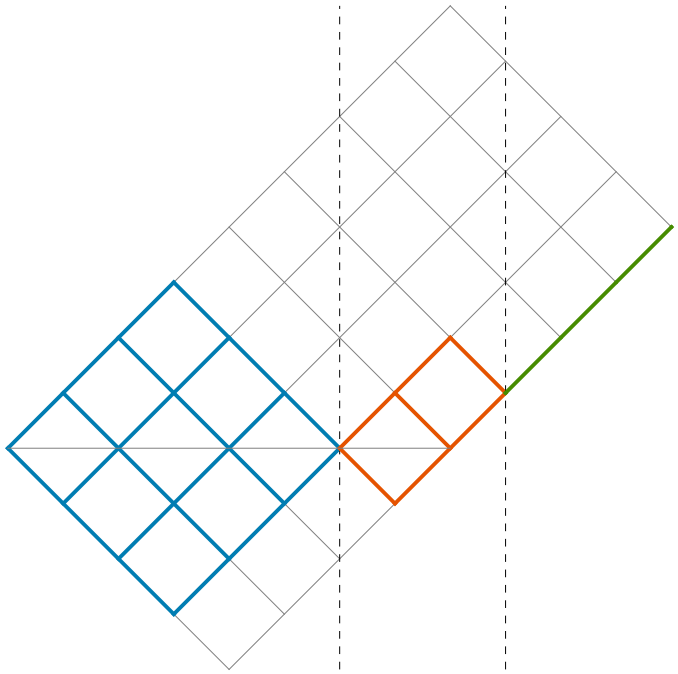
Out[3205]=

$$-\frac{5}{2}$$

In[3206]:=

```
permutationOperatorPathDiagram[{3, 3}, {2, 1}, {3, 0}]
```

Out[3206]=



In[3201]:=

```
enumeratePossibleChangePaths[{6, 3, 3}, 4]
```

Out[3201]=

```
{{{2, 4}, {3, 0}, {3, 0}}, {{3, 3}, {2, 1}, {3, 0}},
 {{3, 3}, {3, 0}, {2, 1}}, {{4, 2}, {1, 2}, {3, 0}},
 {{4, 2}, {2, 1}, {2, 1}}, {{4, 2}, {3, 0}, {1, 2}}, {{5, 1}, {0, 3}, {3, 0}},
 {{5, 1}, {1, 2}, {2, 1}}, {{5, 1}, {2, 1}, {1, 2}}, {{5, 1}, {3, 0}, {0, 3}},
 {{6, 0}, {0, 3}, {2, 1}}, {{6, 0}, {1, 2}, {1, 2}}, {{6, 0}, {2, 1}, {0, 3}}}
```

In[3216]:=

```
Assuming[
 k < 0 && a1 > 0 && a1 > s1 && s1 > 0 && sT > 0,
 Abs[k - (2 a1 - s1)] ≤ sT // Reduce
 (*Abs[k-(2 a1-s1)]≤sT*)
]
```

Out[3216]=

```
sT ≥ Abs[-2 a1 + k + s1]
```

The condition is

$$|k-(2a-s)| \leq s_T$$

In[3224]:=

```
Assuming[a > 0 && s > 0 && sT > 0 && k ∈ Integers,
 Reduce[Abs[k - (2 a - s)] == sT, a,  Reals]
]
```

Out[3224]=

$$\left(\mathtt{sT} == 0 \;\&\&\; \mathtt{a} == \frac{\mathtt{k}+\mathtt{s}}{2}\right) \;||\; \left(\mathtt{sT} > 0 \;\&\&\; \left(\mathtt{a} == \frac{1}{2}\;(\mathtt{k}+\mathtt{s}-\mathtt{sT}) \;||\; \mathtt{a} == \frac{1}{2}\;(\mathtt{k}+\mathtt{s}+\mathtt{sT})\right)\right)$$

but we also know that

$$|k|-|(2a-s)| \leq |k-(2a-s)|$$

and so we have the simplified condition

$$|(2a-s)| \geq |k| - s_T$$

and so if $a > s/2$, we get

$$a \geq \frac{1}{2}\,(|k| - ...)$$

which holds when $a \leq \frac{1}{2}\,(s + s_T - |k|)$

In[351]:=

```
Clear[enumeratePossibleChangePaths2i, enumeratePossibleChangePaths2];
enumeratePossibleChangePaths2i[{s_}, k_] := {{{(s + k)/2, (s - k)/2}}};
enumeratePossibleChangePaths2i[{s1_Integer, s2_Integer}, k_] :=
 Table[
  {
   {a1, s1 - a1},
   {1 / 2 (k + s1 + s2) - a1, 1 / 2 (s2 - (k + s1)) + a1}
  },
  {a1,
   Max@{0, 1 / 2 (k + s1 - s2)},
   Min@{s1, 1 / 2 (k + s1 + s2)}
  }
 ]
enumeratePossibleChangePaths2i[{s1_, steps__}, k_] :=
  With[{sT = Plus[steps]},
   Catenate@
    Table[
     Prepend[{a1, s1 - a1}] /@
      enumeratePossibleChangePaths2[
       {steps},
       k - (2 a1 - s1)
      ],
     {a1,
      Max@{0, 1 / 2 (k + s1 - sT)},
      Min@{s1, 1 / 2 (k + sT + s1)}
     }
    ]
  ];
enumeratePossibleChangePaths2[{steps__}, k_] :=
 If[Mod[Plus[steps], 2] == Mod[Abs@k, 2],
  enumeratePossibleChangePaths2i[{steps}, k],
  {}
 ]
```

In[246]:=

```
enumeratePossibleChangePaths2[{1, 4, 1}, 0]
```

Out[246]=

```
{{{0, 1}, {2, 2}, {1, 0}}, {{0, 1}, {3, 1}, {0, 1}},
 {{1, 0}, {1, 3}, {1, 0}}, {{1, 0}, {2, 2}, {0, 1}}}
```

In[49]:=

```
enumeratePossibleChangePaths[{6, 3, 6, 3, 8}, 4] // Length // RepeatedTiming
```

Out[49]=

```
{0.00456637, 589}
```

In[50]:=
```
enumeratePossibleChangePaths2[{6, 3, 6, 3, 8}, 4] // Length // RepeatedTiming
```

Out[50]=
```
{0.00400309, 589}
```

```
tSqrt[n_] :=
```

In[39]:=
```
Complement[
 enumeratePossibleChangePaths[{6, 3, 3, 2}, 4],
 enumeratePossibleChangePaths2[{6, 3, 3, 2}, 4]
]
```

Out[39]=
```
{}
```

In[18]:=
```
enumeratePossibleChangePaths2[{6, 3, 3}, 4]
```

Out[18]=
```
{{{2, 4}, {3, 0}, {3, 0}}, {{3, 3}, {2, 1}, {3, 0}},
 {{3, 3}, {3, 0}, {2, 1}}, {{4, 2}, {1, 2}, {3, 0}},
 {{4, 2}, {2, 1}, {2, 1}}, {{4, 2}, {3, 0}, {1, 2}}, {{5, 1}, {0, 3}, {3, 0}},
 {{5, 1}, {1, 2}, {2, 1}}, {{5, 1}, {2, 1}, {1, 2}}, {{5, 1}, {3, 0}, {0, 3}},
 {{6, 0}, {0, 3}, {2, 1}}, {{6, 0}, {1, 2}, {1, 2}}, {{6, 0}, {2, 1}, {0, 3}}}
```

In[3280]:=
```
{{{2, 4}, {3, 0}, {3, 0}}, {{3, 3}, {2, 1}, {3, 0}},
  {{3, 3}, {3, 0}, {2, -1}}, {{4, 2}, {1, 2}, {3, 0}}, {{4, 2}, {2, 1}, {2, -1}},
  {{4, 2}, {3, 0}, {1, -2}}, {{5, 1}, {0, 3}, {3, 0}}, {{5, 1}, {1, 2}, {2, -1}},
  {{5, 1}, {2, 1}, {1, -2}}, {{5, 1}, {3, 0}, {0, -3}}, {{6, 0}, {0, 3}, {2, -1}},
  {{6, 0}, {1, 2}, {1, -2}}, {{6, 0}, {2, 1}, {0, -3}}} // Length
```

Out[3280]=
```
13
```

```
s
```

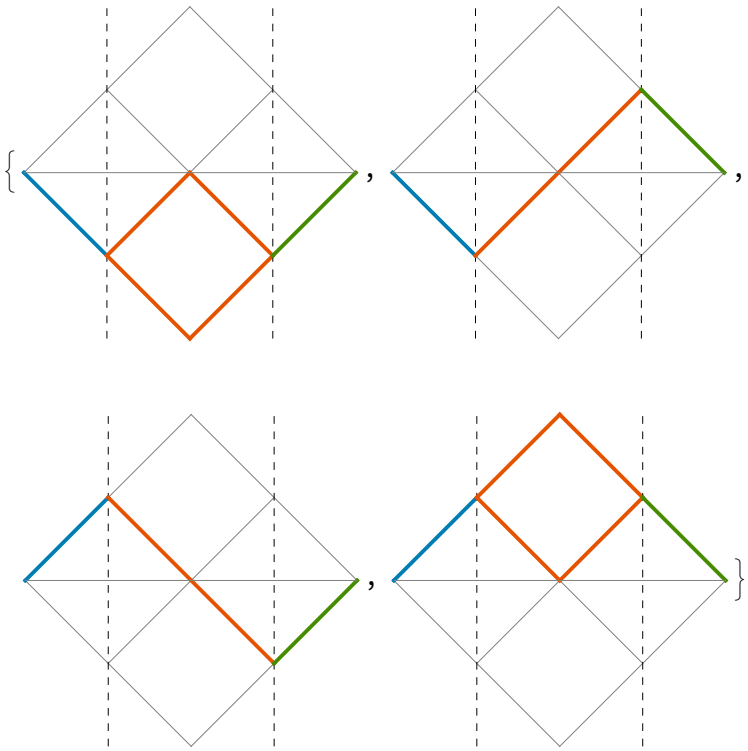
Out[3192]=
```
{{{2, 4}, {3, 0}, {3, 0}}, {{3, 3}, {2, 1}, {3, 0}},
 {{3, 3}, {3, 0}, {2, 1}}, {{4, 2}, {1, 2}, {3, 0}},
 {{4, 2}, {2, 1}, {2, 1}}, {{4, 2}, {3, 0}, {1, 2}}, {{5, 1}, {0, 3}, {3, 0}},
 {{5, 1}, {1, 2}, {2, 1}}, {{5, 1}, {2, 1}, {1, 2}}, {{5, 1}, {3, 0}, {0, 3}},
 {{6, 0}, {0, 3}, {2, 1}}, {{6, 0}, {1, 2}, {1, 2}}, {{6, 0}, {2, 1}, {0, 3}}}
```

```
permutationOperatorPathDiagram[{3, 2}, {1, 0}, {3, 4}, {3, 1}]
```

In[359]:=

```
permutationOperatorPathDiagram @@@ enumeratePossibleChangePaths2[{1, 2, 1}, 0]
```

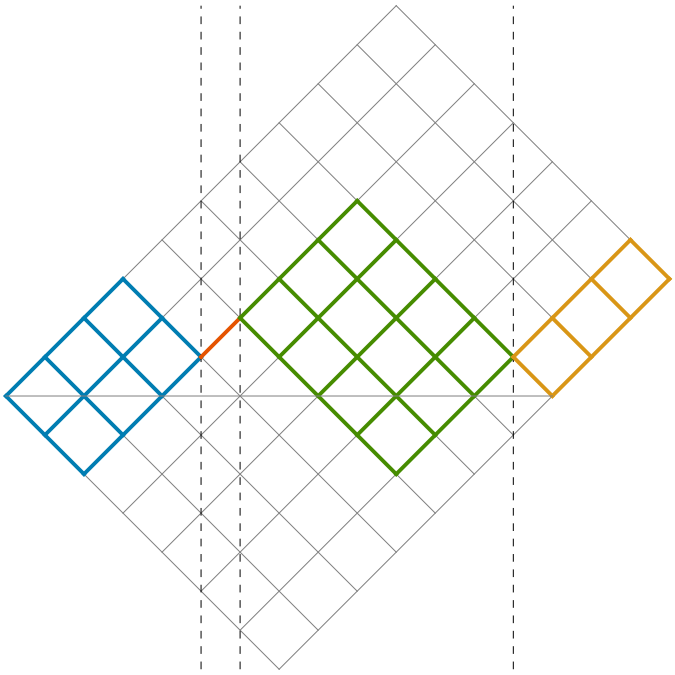
Out[359]=



In[3143]:=

```
permutationOperatorPathDiagram[{3, 2}, {1, 0}, {3, 4}, {3, 1}]
```

Out[3143]=



Inclusion of Momenta

Generic Momentum Evaluation

Mixed Products

Our final forms will be given as things like

$$\left\langle n \right| x^{s_1} p^{s_2} x^{s_3} p^{s_4} \left| n{+}k \right\rangle$$

and just like regular K-products we'll enumerate the possible state change paths and use the fact that

$$\left\langle n \right| p^s \left| n{+}k \right\rangle = (-1)^b \left\langle n \right| x^s \left| n{+}k \right\rangle$$

to quickly build the appropriate alternating series

Stupid Attempt

Maximally Permuted Perturbation Theory

Let's consider the evaluation of a transition moment in VPTN

$$\sum_{j=0}^{k-i} \left\langle n^{(i)} \right| \mu^{(k-i-j)} \left| m^{(j)} \right\rangle \Bigg)$$

$$n^{(i)} \Big| \sum_{k=0}^{N-i-j} \mu^{(k)} \left| m^{(j)} \right\rangle$$

$$\mu^{(k)} \left| m^{(0)} \right\rangle + \left\langle n^{(0)} \right| \Delta \mathrm{H}_n^{(1)} \, \Pi_n \sum_{k=0}^{N-1} \mu^{(k)} + \sum_{k=0}^{N-1} \mu^{(k)} \, \Pi_m \, \Delta \mathrm{H}_m^{(1)} \left| m^{(0)} \right\rangle$$

$$n^{(0)} \Big| \left(\Delta \mathrm{H}_n^{(2)} \, \Pi_n + \Delta \mathrm{H}_n^{(1)} \, \Pi_n \, \Delta \mathrm{H}_n^{(1)} \, \Pi_n + \right) \sum_{k=0}^{N-2} \mu^{(k)} + \sum_{k=0}^{N-2} \mu^{(k)} \left(\Pi_m \, \Delta \mathrm{H}_m^{(2)} + \Pi_m \, \Delta \mathrm{H}_m^{(1)} \, \Pi_m \, \Delta \mathrm{H}_m^{(1)} \right) + \Delta \mathrm{H}_n^{(1)} \, \Pi_n \sum_{k=0}^{N-2}$$

...

and from this it *should* be clear that we have

$$\left\langle n \right| \mu \left| m \right\rangle = \sum_{s,r} \left\langle n^{(0)} \right| \left(\prod_i \Delta \mathrm{H}_n^{(s_i)} \, \Pi_n \right) \mu^{(k)} \left(\prod_j \Pi_m \, \Delta \mathrm{H}_m^{(r_j)} \right) \left| m^{(0)} \right\rangle$$

where

$$\sum_i s_i + \sum_j r_j + k \leq N$$

which will lead to long strings of raising and lowering operations, and in particular tabulating the options at second order with the linear dipole in $\mu^{(0)}$ we get

$$\#\left|\left\langle n^{(0)}\right|\sum_{k=0}^{2}\mu^{(k)}\left|m^{(0)}\right\rangle\right| = (1,),\ (2,),\ (3,)$$

$$\#\left|\left\langle n^{(0)}\right|\Delta\mathrm{H}_n^{(1)}\,\Pi_n\sum_{k=0}^{1}\mu^{(k)}\left|m^{(0)}\right\rangle\right| = (3,\ 1),\ (3,\ 2)$$

$$\#\left|\left\langle n^{(0)}\right|\sum_{k=0}^{N-1}\mu^{(k)}\,\Pi_m\,\Delta\mathrm{H}_m^{(1)}\left|m^{(0)}\right\rangle\right| = (1,\ 3),\ (2,\ 3)$$

$$\#\left|\left\langle n^{(0)}\right|\Delta\mathrm{H}_n^{(2)}\,\Pi_n\,\mu^{(0)}\left|m^{(0)}\right\rangle\right| = (4,\ 1)$$

$$\#\left|\left\langle n^{(0)}\right|\mu^{(0)}\,\Pi_m\,\Delta\mathrm{H}_m^{(2)}\left|m^{(0)}\right\rangle\right| = (1,\ 4)$$

$$\#\left|\left\langle n^{(0)}\right|\Delta\mathrm{H}_n^{(1)}\ \Pi_n\ \Delta\mathrm{H}_n^{(1)}\,\Pi_n\,\mu^{(0)}\left|m^{(0)}\right\rangle\right| = (3,\ 3,\ 1)$$

$$\#\left|\left\langle n^{(0)}\right|\mu^{(0)}\,\Pi_m\,\Delta\mathrm{H}_m^{(1)}\,\Pi_m\,\Delta\mathrm{H}_m^{(1)}\left|m^{(0)}\right\rangle\right| = (1,\ 3,\ 3)$$

$$\#\left|\left\langle n^{(0)}\right|\Delta\mathrm{H}_n^{(1)}\,\Pi_n\,\mu^{(0)}\,\Pi_m\ \Delta\mathrm{H}_m^{(1)}\left|m^{(0)}\right\rangle\right| = (3,\ 1,\ 3)$$

but given the appropriate expansion coefficients for each operator, we can of course work this all out programmatically

At the end of the day, though, as noted we're taking every possible sequence of $H^{(i)}$ and $\mu^{(k)}$where the sum of the exponents is less than or equal to our total order (and of course we can add on corrections pretty easily)

Each of these operators now can be expanded like

$$\sum_{z\in\mathrm{idx}(i)}(V_z\pm G_z)\,abc...$$

where $abc...$ are strings of raising and lowering operations

Therefore to evaluate, say

$$\left\langle n^{(0)}\right|\Delta\mathrm{H}_n^{(1)}\,\Pi_n\,\mu^{(0)}\,\Pi_m\ \Delta\mathrm{H}_m^{(1)}\left|m^{(0)}\right\rangle$$

we need to first determine which quantum number changes we need to do to get from n to m, then we have to write out the classes of "do nothing" terms in the remaining coordinates, which if we have say 8 remaining operators looks like

$$\begin{array}{c} i^+i^-\ i^+i^-\ i^+i^-\ i^+i^- \\ i^+i^-\ i^+i^-\ i^+i^-\ j^+j^- \\ i^+i^-\ i^+i^-\ j^+j^-\ j^+j^- \\ i^+i^-\ j^+j^-\ j^+j^-\ k^+k^- \\ i^+i^-\ i^+i^-\ j^+j^-\ k^+k^- \end{array}$$

where i, j, and k can take on any value, which means we do need to consider when they are the same as one of the changed coordinates and when they are not

For each class of raising and lowering operations, we then enumerate the *subclasses* of terms, by noting the different ways we can distribute them over the corresponding operators, again, for

$$\left\langle n^{(0)} \right| \Delta \mathrm{H}_n^{(1)} \, \Pi_n \, \mu^{(0)} \, \Pi_m \, \Delta \mathrm{H}_m^{(1)} \left| m^{(0)} \right\rangle$$

the operator string $i^+ \, i^+ j^+ \, i^+ i^- \, k^+ k^-$ would have the classes

$$\begin{gathered} i^+ \, i^+ i^+ i^- \, j^+ \, k^+ k^- \\ i^+ \, i^+ i^+ i^- \, j^+ \, k^- k^+ \\ i^+ \, i^+ i^+ i^- \, k^+ \, j^+ k^- \\ i^+ \, i^+ i^+ i^- \, k^- \, j^+ k^+ \\ i^+ \, i^+ i^+ i^- \, k^+ k^- \, j^+ \\ \ldots \end{gathered}$$

but we can reduce much of this over permutations by the work done before, so we are better off characterizing this by the number of operations in each dimension in each of three buckets, so like

$$\begin{gathered} i^+ \, i^+ i^+ i^- \, j^+ \, k^+ k^- \to [3,\, 0,\, 0] \mid [1,\, 0,\, 0] \mid [0,\, 1,\, 2] \\ i^+ \, i^+ i^+ i^- \, j^+ \, k^- k^+ \to [3,\, 0,\, 0] \mid [1,\, 0,\, 0] \mid [0,\, 1,\, 2] \\ i^+ \, i^+ i^+ i^- \, k^+ \, j^+ k^- \to [3,\, 0,\, 0] \mid [1,\, 0,\, 0] \mid [0,\, 1,\, 2] \\ i^+ \, i^+ i^+ i^- \, k^- \, j^+ k^+ \to [3,\, 0,\, 0] \mid [1,\, 0,\, 0] \mid [0,\, 1,\, 2] \\ i^+ \, i^+ i^+ i^- \, k^+ k^- \, j^+ \to [3,\, 0,\, 0] \mid [1,\, 0,\, 0] \mid [0,\, 1,\, 2] \\ \ldots \end{gathered}$$

which as we can see reduces all of these different permutations to just one subclass, and indeed these subclasses are straightforwardly enumerable, first by noting that we have buckets of length 3, 1, and 3, and 4 terms in dimension i, 1 term in dimension j, and 2 in dimension k which we'll distribute recursively

First, we can start off by taking the integer partitions of $4i$ into up to 3 terms and distribute these over the different buckets, like

$$(3 \mid 1 \mid 0),\, (3 \mid 0 \mid 1),\, (1 \mid 0 \mid 3),\, (0 \mid 1 \mid 3),\, (2 \mid 0 \mid 2),\, (2 \mid 1 \mid 1),\, (1 \mid 1 \mid 2)$$

then we can take the 1 in dimension j and add that to the buckets in as many ways as are possible

$$\begin{gathered} ([3,\, 0] \mid [1,\, 0] \mid [0,\, 1]),\, ([3,\, 0] \mid [0,\, 1] \mid [1,\, 0]),\, ([3,\, 0] \mid [0,\, 0] \mid [1,\, 1]) \\ ([0,\, 1] \mid [1,\, 0] \mid [3,\, 0]),\, ([1,\, 0] \mid [0,\, 1] \mid [3,\, 0]),\, ([1,\, 1] \mid [0,\, 0] \mid [3,\, 0]) \\ ([2,\, 1] \mid [0,\, 0] \mid [2,\, 0]),\, ([2,\, 0] \mid [0,\, 1] \mid [2,\, 0]),\, ([2,\, 0] \mid [0,\, 0] \mid [2,\, 1]) \\ \ldots \end{gathered}$$

and finally we can iterate over the permutations of the integer partitions of 2 and add those in

This will give us the ways we can divide the operations up, then we can use the recursive strategy for enumerating the possible energetic contributions from there, giving us energy change paths in each dimension for each class of terms

The energy contribution then will be the direct product of the polynomials over each path with divided by the direct product of the energy changes, all multiplied by the appropriate operator expansion coefficients

Tail Recursive Expansions

Finally, if we can take these tail-recursive processes and expand them out to build up as functions of permutations we should be able to parallelize well so to that we'll provide a constructive algorithm for enumerating the classes of terms possible over a set of buckets and terms

To start, we note that given N buckets $\{s_i\}$ and a set of numbers of operations $\{o_j\}$, we can take our first operator and divide it up into its integer partitions

$$\pi_1 = \{(n_1,\ n_2,\ \ldots,\ n_N)_k \mid n_1+\ldots+n_N=o_1\ \}$$

taking then all the permutations of this, $\{\sigma_l^{(1)}\}$, dropping the ones that don't go cleanly into the $\{s_i\}$, for each permutation/partition we get a new set of buckets and we've reduced our set of operators by 1

However, we can also do this in the *opposite* direction, by building towards the $\{s_i\}$, starting from every admissible permutation partition

Why does this help? Well from this, it should be clear that we will need all of the integer partitions with maximum part no larger than the biggest s_i of all of our operators, so we can precompute those

Tree Expression for Classes of Terms

Fourier Treatment of Polynomial Coeffs

Exploration

Implementation

Recursive Polynomial PT

$$5\ P_1\ P_2\ P_3\ +\ 3\ P_1\ P_2\ P_j$$

To avoid redoing a bunch of work it would be nice to take the expressions we have

$$E_n^{(k)} = \langle n^{(0)}| \; H^{(k)} \; |n^{(0)}\rangle + \sum_{i=1}^{k-1} \langle n^{(0)}| \; H^{(k-i)} \; |n^{(i)}\rangle - E_n^{(k-i)} \langle n^{(0)} | n^{(i)}\rangle$$

$$\langle n^{(0)} | n^{(k)}\rangle = -\frac{1}{2} \sum_{i=1}^{k-1} \langle n^{(i)} | n^{(k-i)}\rangle$$

$$P_n |n^{(k)}\rangle = \Pi_n \left(\sum_{i=0}^{k-1} \left(E_n^{(k-i)} - H^{(k-i)} \right) |n^{(i)}\rangle \right)$$

to evaluate any property we need for the PT

To do so, we'll assume to start that for every $i < k$ we can evaluate

$$\langle m^{(0)}| \, n^{(i)}\rangle = W^{(i)}_{(n-m)}(n)$$

where $W^{(i)}_{\Delta}$ is a polynomial in the quantum numbers (n) encoding the ith order wave function correction for a change in quanta by Δ

This will build off of a set of Hamiltonian operator polynomials given by

$$\langle n^{(0)} + \Delta| \; H^{(k)} \; |n^{(0)}\rangle = H^{(k)}_{\Delta}(n)$$

then when we have something like

$$\langle n^{(0)}| \; H^{(k-i)} \; |n^{(i)}\rangle$$

we'll note that we can change quanta in the range $[-(2+k-i),\ 2+k-i]$ stepping by twos, and for each of those changes we'll then get the polynomial

$$\langle n^{(0)}| \; H^{(k-i)} \; |n^{(i)}\rangle = \sum_{\Delta} \left(H^{(k-i)}_{\Delta} * W^{(i)}_{-\Delta} \right)(n)$$

To consider how this polynomial is constructed, we'll note that, letting $k = 4$, if Δ changes one mode by two quanta, then

$$\langle n^{(0)} + \Delta| \; H^{(k)} \; |n^{(0)}\rangle = H^{(k)}_{\Delta}(n)$$

is a polynomial with three distinct contributions, we have

- a 1D polynomial in the changed mode coming from putting all 6 possible changes in on degree of freedom
- a 2D polynomial from putting 4 quanta in the changed mode and 2 in any arbitrary unchanged mode (where we at a later step must sum over all unchanged modes)
- a 2D polynomial from putting 2 quanta in the changed mode and 4 in any arbitrary unchanged mode
- a 3D polynomial from putting 2 quanta in the changed mode and 2 each in any pair of unchanged modes, and again at a later step we will sum over pairs

Then when we take the product with $W^{(i)}_{\Delta}$, these will be the four cases we need to consider, *however* we also need to note that the polynomial contributions to $W^{(i)}_{\Delta}$ will also have a set of

"free" modes, and the free modes from $H_{\Delta}^{(k)}(n)$ will not necessarily align with the free modes from $W_{\Delta}^{(i)}$ and therefore we must take the direct product of free modes aligning and not aligning
Mechanistically, when computing the terms available to e.g. $H^{(1)}\left|n^{(1)}\right\rangle$ we need to be a bit careful, $H^{(1)}$ can change quanta by

$$H^{(1)} \sim q_1^3,\ q_1^2 q_2,\ q_1 q_2 q_3$$

and so $\left|n^{(1)}\right\rangle$ has these changes for all possible operators plus a contribution from changing no quanta, and so we end up with

$$H^{(1)}\left|n^{(1)}\right\rangle \sim q_1^6, q_1^3 q_2^3,\ q_1^3 q_2^2 q_3,\ q_1^3 q_2 q_3 q_4,\ q_1^2 q_2^2 q_3 q_4,\ q_1^2 q_2 q_3 q_4 q_5,\ q_1 q_2 q_3 q_4 q_5 q_6$$

but difficultly, while there's only one way to get q_1^6, there are multiple ways yo get to, e.g., $q_1^3 q_2 q_3 q_4$ and so assuming initially that

$$\left\langle n^{(0)}+\Delta_1\right| n^{(i)}\rangle = W_{\Delta_1}^{(i)}(n)$$

for Δ_1=(2, 1), then

$$\left\langle n^{(0)}\right|\ H^{(k-i)}\left|n^{(i)}\right\rangle = \left(H_{\Delta}^{(k-i)} * W_{\Delta}^{(i)}\right)(n)$$

$$\left\langle n^{(0)}\right|\ H^{(k-i)}\left|n^{(i)}\right\rangle = \left(H_{\Delta}^{(k-i)} * W_{\Delta}^{(i)}\right)(n)$$

we note that

One complication with a simple polynomial approach is that when we have terms like

$$\left\langle n^{(0)}+\Delta\right|\ \Pi H^{(1)}\left|n^{(0)}\right\rangle = W_{\Delta}^{(1)}(n)$$

we need to include the energy weighting for the given paths, which are in turn multi-dimensional...eh not an issue

$$W_{\Delta}^{(1)}(n) = \frac{1}{E_{\Delta}} P_{\Delta}^{(1)}(n)$$

In[1216]:=

```
{2, 1, 0, 0, 0} + # & /@
 Permutations[
  {3, 2, 1, 0, 0}
 ]
```

Out[1216]=

```
{{5, 3, 1, 0, 0}, {5, 3, 0, 1, 0}, {5, 3, 0, 0, 1}, {5, 2, 2, 0, 0}, {5, 2, 0, 2, 0},
 {5, 2, 0, 0, 2}, {5, 1, 2, 1, 0}, {5, 1, 2, 0, 1}, {5, 1, 1, 2, 0}, {5, 1, 1, 0, 2},
 {5, 1, 0, 2, 1}, {5, 1, 0, 1, 2}, {4, 4, 1, 0, 0}, {4, 4, 0, 1, 0}, {4, 4, 0, 0, 1},
 {4, 2, 3, 0, 0}, {4, 2, 0, 3, 0}, {4, 2, 0, 0, 3}, {4, 1, 3, 1, 0}, {4, 1, 3, 0, 1},
 {4, 1, 1, 3, 0}, {4, 1, 1, 0, 3}, {4, 1, 0, 3, 1}, {4, 1, 0, 1, 3}, {3, 4, 2, 0, 0},
 {3, 4, 0, 2, 0}, {3, 4, 0, 0, 2}, {3, 3, 3, 0, 0}, {3, 3, 0, 3, 0}, {3, 3, 0, 0, 3},
 {3, 1, 3, 2, 0}, {3, 1, 3, 0, 2}, {3, 1, 2, 3, 0}, {3, 1, 2, 0, 3}, {3, 1, 0, 3, 2},
 {3, 1, 0, 2, 3}, {2, 4, 2, 1, 0}, {2, 4, 2, 0, 1}, {2, 4, 1, 2, 0}, {2, 4, 1, 0, 2},
 {2, 4, 0, 2, 1}, {2, 4, 0, 1, 2}, {2, 3, 3, 1, 0}, {2, 3, 3, 0, 1}, {2, 3, 1, 3, 0},
 {2, 3, 1, 0, 3}, {2, 3, 0, 3, 1}, {2, 3, 0, 1, 3}, {2, 2, 3, 2, 0}, {2, 2, 3, 0, 2},
 {2, 2, 2, 3, 0}, {2, 2, 2, 0, 3}, {2, 2, 0, 3, 2}, {2, 2, 0, 2, 3}, {2, 1, 3, 2, 1},
 {2, 1, 3, 1, 2}, {2, 1, 2, 3, 1}, {2, 1, 2, 1, 3}, {2, 1, 1, 3, 2}, {2, 1, 1, 2, 3}}
```

In[1223]:=

```
Map[{2, 1} + # &]@*Permutations /@ Subsets[{3, 2, 1}, {2}]
```

```
{
 {{5, 3}, {4, 4}},
 {{5, 2}, {3, 4}},
 {{4, 2}, {3, 3}}
}
```

In[1218]:=

```
{2, 1, 0, 0} +
 {1, 3, 2, 0}
```

```
{3, 4, 2, 0}
```

$$[(P_3 * P_1),\ (P_2 * P_1),\ P_2]$$

In[1219]:=

```
{2, 1, 0, 0} +
 {2, 1, 3, 0}
```

Out[1219]=

```
{4, 2, 3, 0}
```

$$[(P_2 * P_2),\ P_3,\ P_1 * P_1]$$

In[1212]:=

```
GroupBy[
  {2, 1, 0, 0, 0} + # & /@
   Permutations[
    {3, 2, 1, 0, 0}
   ],
  Reverse@*Sort
 ] // Keys
```

Out[1212]=

```
{{5, 3, 1, 0, 0}, {5, 2, 2, 0, 0}, {5, 2, 1, 1, 0},
 {4, 4, 1, 0, 0}, {4, 3, 2, 0, 0}, {4, 3, 1, 1, 0}, {3, 3, 3, 0, 0},
 {3, 3, 2, 1, 0}, {4, 2, 2, 1, 0}, {3, 2, 2, 2, 0}, {3, 2, 2, 1, 1}}
```

In[1215]:=

```
IntegerPartitions[9, 5] // Select[#[[1]] < 6 &] // Column
```

Out[1215]=

```
{5, 4}
{5, 3, 1}
{5, 2, 2}
{5, 2, 1, 1}
{5, 1, 1, 1, 1}
{4, 4, 1}
{4, 3, 2}
{4, 3, 1, 1}
{4, 2, 2, 1}
{4, 2, 1, 1, 1}
{3, 3, 3}
{3, 3, 2, 1}
{3, 3, 1, 1, 1}
{3, 2, 2, 2}
{3, 2, 2, 1, 1}
{2, 2, 2, 2, 1}
```

State Selection Rules

The number of states that are required to be accurate enough with the perturbation theory grows exponentially, as do the number of potentially relevant transitions. If we have N_{vib} vibrational modes and want to put up to M quanta of excitation into any of these modes, assuming $M<N_{\text{vib}}$ the number of states we're working with can be defined recursively for $M>1$ as

$$\rho(N_{\text{vib}},\ M)=\rho(N_{\text{vib}},\ M-1)\ +\ \rho(N_{\text{vib}}-1,\ M)$$

where the $\rho(N_{\text{vib}},\ M-1)$ contribution comes from the fact that the $M-1$ states also have fewer than M quanta and the $\rho(N_{\text{vib}}-1,\ M)$ comes from adding an extra quantum to all of those states & dealing with duplicates. I might actually be mildly over counting in this formula, but at the very least it suffices to show the exponential blow-up we expect.

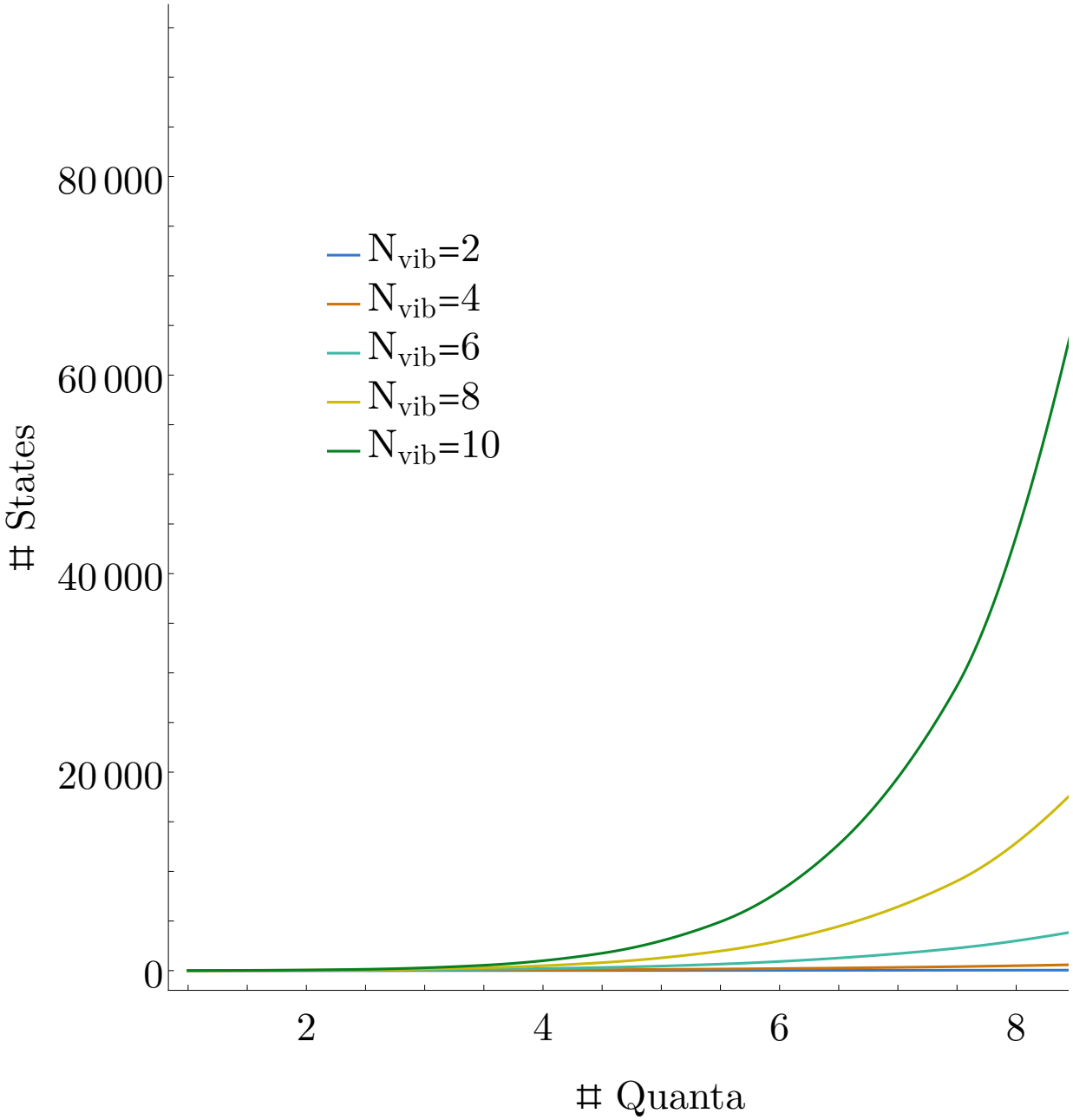
When N_{vib}=2, these will grow as the triangular numbers, defined as

$$a(n) = \binom{n+1}{2}$$

and when $\rho(N_{\text{vib}}, 1) = N_{\text{vib}}$ so this building recursively we're going to by something like a binomial expansion on the binomial numbers. I'll leave the proper analysis of this kind of thing to the number theorists, who have certainly already studied problems like this and just leave you with a plot of what this looks like

```
rhoNM // Clear
rhoNM[nvib : {__Integer}, m : {__Integer}] :=
  Block[{rhoNM},
   rhoNM[nv_Integer, 0] := 1;
   rhoNM[nv_Integer, 1] := nv + 1;
   rhoNM[2, mq_Integer] := Binomial[mq + 2, 2];
   rhoNM[nv_Integer, mq_Integer] :=
    rhoNM[nv, mq] = rhoNM[nv, mq - 1] + rhoNM[nv - 1, mq];
   SetAttributes[rhoNM, Listable];
   rhoNM[nvib, m]
  ];
rhoNM[nvib_Integer, m_Integer] := rhoNM[{nvib}, {m}][[1]];
figListLinePlot[
  Table[
   Transpose[{Range[9], rhoNM[ConstantArray[n, 9], Range[9]]}],
   {n, 2, 10, 2}
  ],
  {"# Quanta", "# States"},
  InterpolationOrder → 2,
  PlotLegends →
   Map[figStyleText@StringForm["N_vib=``", #] &, Table[n, {n, 2, 10, 2}]],
  PlotMarkers → None,
  AspectRatio → 1
 ] // insetLegend[#, Scaled[{.25, .65}], Scaled[{.5, .5}]] &
```

Out[2450]=



This is already bad for studying the entire space of highly-excited vibrations. On the other hand, we're mostly interested in vibrations that put up to two quanta in given modes, and especially up to two quanta in modes we care about.

To that effect, rather than computing our Hamiltonian elements in the total set of possibly-coupled states, we'd like to take a set of initial states of interest and generate the space of states that really do couple to those ones. To generate this space, we need to understand how our zero-order transform under application of our perturbations. We'll do this with the harmonic Hamiltonian that we're mostly to want to use, but the ideas apply to any initial basis you might care

about.

$H^{(1)}$

By assumption, $H^{(0)}$ is diagonal, so first we'll focus in on $H^{(1)}$ where we have

$$H^{(1)}=\nabla_Q G\odot pQp+\nabla_Q F\odot QQQ$$

and now if we stick any two states in there we have

$$\langle n_1,\ n_2,\ ...|\ H^{(1)}\ |m_1,\ m_2,\ ...\rangle=$$
$$\nabla_Q G\odot\langle n_1,\ n_2,\ ...|\ pQp\ |m_1,\ m_2,\ ...\rangle+\nabla_Q F\odot\langle n_1,\ n_2,\ ...|\ QQQ\ |m_1,\ m_2,\ ...\rangle$$

and now we have to deal with the terms that look like

$$\langle n_1,\ n_2,\ ...|\ p_iQ_jp_k\ |m_1,\ m_2,\ ...\rangle=\langle n_i,\ n_j,\ n_k|\ p_iQ_jp_k\ |m_i,\ m_j,\ m_k\rangle\ \langle\{n_l\}\backslash\{i,\ j,\ k\}|\{m_l\}\backslash\{i,\ j,\ k\}\rangle$$
$$\langle n_1,\ n_2,\ ...|\ Q_iQ_jQ_k\ |m_1,\ m_2,\ ...\rangle=\langle n_i,\ n_j,\ n_k|\ Q_iQ_jQ_k\ |m_i,\ m_j,\ m_k\rangle\ \langle\{n_l\}\backslash\{i,\ j,\ k\}|\{m_l\}\backslash\{i,\ j,\ k\}\rangle$$

the overlap term is just a big pile of delta functions

$$\langle\{n_l\}\backslash\{i,\ j,\ k\}|\{m_l\}\backslash\{i,\ j,\ k\}\rangle=\prod_{\alpha\notin\{i,\ j\ k\}}\delta_{n_\alpha\ m_\alpha}$$

and so we can clearly only ever change *at most* three indices at once through $H^{(1)}$.

Next we need to consider the three cases for $\{i,\ j,\ k\}$

Case 1: $i=j=k$

This is the simplest case, where we've got the same index three times. From our δ terms, this means we can only change one mode. Constructing our raising/lower representation of the operators we have

$$\begin{aligned}p_iQ_ip_i&=\left(a_i-a_i^\dagger\right)\left(a_i+a_i^\dagger\right)\left(a_i-a_i^\dagger\right)\\&=\left(a_i-a_i^\dagger\right)\left(a_i\ a_i+a_i^\dagger\ a_i-a_i^\dagger\ a_i^\dagger\right)\\&=a_i\ a_i\ a_i-a_i^\dagger\ a_i\ a_i-a_i^\dagger\ a_i^\dagger\ a_i+a_i^\dagger\ a_i^\dagger\ a_i^\dagger\\Q_iQ_iQ_i&=a_i\ a_i\ a_i+a_i^\dagger\ a_i\ a_i+a_i^\dagger\ a_i^\dagger\ a_i+a_i^\dagger\ a_i^\dagger\ a_i^\dagger\end{aligned}$$

which means we can raise/lower our quanta by one or three through this operator

Case 2: $i\neq j=k$

In this case, we can change up to two modes. Constructing our raising/lower representation of the operators we have a few different flavors, based on the different permutations we of the indices we can do, but we'll focus on

$$Q_iQ_jQ_j=\left(a_i+a_i^\dagger\right)\left(a_j+a_j^\dagger\right)\left(a_j+a_j^\dagger\right)$$

$$=\left(a_i+a_i^\dagger\right)\left(a_j\ a_j+a_j^\dagger\ a_j+a_j^\dagger\ a_j^\dagger\right)$$

and from this we can change one mode by up to 2 and another mode by one

Case 3: $i\neq j\neq k$

We're just gonna jump ahead to the punch-line and leave working it out to the reader. We can change three modes at a time and each mode can change by one quantum.

$H^{(2)}$

We'll recall

$$H^{(1)}=\nabla_{Q^2}G\odot pQQp+\nabla_{Q^2}F\odot QQQQ$$

this will allow things to change by up to 4 quanta in up to 4 modes.

Orthogonality

When we have an operator like Q_iQ_j, any modes *not* in $\{q_i,\ q_j\}$ are unaffected by its operation, so if e.g. if we have a set of modes $q_a,\ q_b,\ q_c,\ q_d$ and want to know if two states $\left|n_a,\ n_b,\ n_c,\ n_d\right\rangle$ and $\left|m_a,\ m_b,\ m_c,\ m_d\right\rangle$ are possibly coupled through the $Q_b\ Q_c$ we have

$$n_a==m_a\&\&n_d==m_d$$

but now we also know we'll need to check this for $Q_b\ Q_d$ and $Q_a\ Q_b$, etc. so if there's a way to reuse information to compute fewer orthogonality checks that would pay dividends.

To that effect, we'll do a bunch of Boolean algebra. First we'll assume we have set of states which we'll denote by $\{i\}$. Then we'll assume we have an operator that couples states $\{1,\ 2,\ 3\}$ and write whether a state can couple through this operator as

$$C(\{1,\ 2,\ 3\})=\bigwedge_{i\notin\{1,\ 2,\ 3\}}n_i==m_i$$

next if we have an operator that couples states $\{1,\ 2,\ 3,\ 4\}$ we have

$$C(\{1,\ 2,\ 3,\ 4\})=\bigwedge_{i\notin\{1,\ 2,\ 3,\ 4\}}n_i==m_i$$

and it would be nice to be able to express this in terms of $\{1,\ 2,\ 3\}$ if possible.

For that, we can write

$$\begin{aligned}C(\{1,\ 2,\ 3,\ 4\})&=\left(\bigwedge_{i\notin\{1,\ 2,\ 3\}}n_i==m_i\right)\vee\left(n_4\neq m_4\wedge\left(\bigwedge_{i\notin\{1,\ 2,\ 3,\ 4\}}n_i==m_i\right)\right)\\&=C(\{1,\ 2,\ 3\})\vee\left(n_4\neq m_4\wedge\left(\bigwedge_{i\notin\{1,\ 2,\ 3,\ 4\}}n_i==m_i\right)\right)\end{aligned}$$

which works because we've now got both $n_4 \neq m_4$ and $n_4 == m_4$ accounted for.

The problem is that this is *more* computationally expensive that just computing $C(\{1, 2, 3, 4\})$ directly, since you'll notice that it includes $C(\{1, 2, 3, 4\})$ itself...

We're going to reexpress this, though, in terms of Boolean algebra in the hope that it will give us a better route forward. To that end, we'll say

$$A = C(\{1, 2, 3, 4\})$$
$$B = C(\{1, 2, 3\})$$
$$D = n_4 \neq m_4$$

giving us

$$A = B \vee (D \wedge A)$$

and then to have expressions that work in terms of 0 and 1 we write

$$x \wedge y = xy$$
$$x \vee y = x + y - xy$$
$$\neg x = 1 - x$$

which when applied to what we had before gives us

$$A = B + DA - BDA$$
$$= B + (1 - B)\, DA$$

and we note from this that $B=1 \Rightarrow A=B$ and similarly when $B=0$ and $D=0$, $A=0$, which just means we don't need to calculate anything there. Finally, when $B=0$ and $D=1$ we get a tautology and so we need to just calculate A directly in that case. OTOH this means we've cut down the amount of work we need to do pretty dramatically, since we only need to calculate things for the intersection of $\neg B$ and D.

Tensor Operations

Tensor Addition and Scalar Tensor Multiplication

These two operations will be defined elementwise, that is to say

Given $\boldsymbol{A}$, $\boldsymbol{B}$ tensors, c a scalar,

$$(A+B)_{i_1\, i_2\, i_3\, \ldots i_n} = A_{i_1\, i_2\, i_3\, \ldots i_n} + B_{i_1\, i_2\, i_3\, \ldots i_n}$$

$$(\mathrm{cA})_{i_1\, i_2\, i_3\, \ldots i_n} = c * A_{i_1\, i_2\, i_3\, \ldots i_n}$$

Or extending this a bit, given $\boldsymbol{A}$, $\boldsymbol{B}$ tensors, c, d scalars,

$$(\mathrm{cA}+\mathrm{dB})_{i_1\, i_2\, i_3\, \ldots i_n} = \mathrm{cA}_{i_1\, i_2\, i_3\, \ldots i_n} + \mathrm{dB}_{i_1\, i_2\, i_3\, \ldots i_n}$$

This has a few important ramifications, namely

- Tensors can only be added if all of their dimensions are aligned
- Tensor scalar multiplication and tensor addition do not change the overall dimension of the tensor
- Tensor scalar multiplication distributes over tensor addition
- Tensor scalar multiplication and tensor addition can be used to form vector spaces of tensors
- All linear operators (e.g. derivatives) will distribute and act normally with the scalar multiplications

Tensor Transposition

This operations is the analog of the matrix transpose in that it changes the way indexing works in a tensor

Given a tensor $\boldsymbol{A}$ with axes i_1, i_2, ... i_N we define the tensor transpose $\boldsymbol{A}_{T(\{k_m\})}$ such that

$$A_{T(k_1,\ k_2,\ldots,k_N)} \text{ has axes } i_{k_1},\ i_{k_2},\ \ldots\ i_{k_N}$$

$$\left(A_{T(k_1,\ k_2,\ldots,k_N)}\right)_{j_1\ j_2\ \ldots,j_N} = A_{k_{j_1} k_{j_2}\ \ldots\ k_{j_N}}$$

Note that for the case where N = 2 the traditional transpose is

$$A^{\mathrm{T}} = A_{T(2,\ 1)}$$

Axes Shift

If we are simply shift one axis to a new position, we define an alternate notation for convenience

Given a tensor $\boldsymbol{A}$ with axes i_1, i_2, ... i_N, assuming WLOG that $n>m$

$$A^{n\to m} = A_{T(1,\ 2,\ \ldots,\ m-1,\ n,\ m,\ m+1,\ \ldots,\ n-1,\ n+1,\ \ldots,\ N)}$$

Axes Swap

If we are simply swap two axes, we define an alternate notation for convenience

Given a tensor $\boldsymbol{A}$ with axes i_1, i_2, ... i_N, assuming WLOG that $n>m$

$$A^{n\leftrightarrow m} = A_{T(1,\ 2,\ \ldots,\ m-1,\ n,\ m+1,\ \ldots,\ n-1,\ m,\ n+1,\ \ldots,\ N)}$$

Property 1 (Derivative Relations)

WLOG we will state this only for swaps, but the same ideas hold for standard transpositions

Scalar Derivatives

Given a swap $\boldsymbol{A}^{n\to m}$ and a single cordinate x

$$\frac{\partial}{\partial x}\boldsymbol{A}^{n\to m}=\left(\frac{\partial}{\partial x}\boldsymbol{A}\right)^{n\to m}$$

Proof

$\frac{\partial}{\partial x}$ is an elementwise operator

Vector Derivatives

Given a swap $\boldsymbol{A}^{n\to m}$ and a vector of cordinates X

$$\nabla_{X^k}A^{n\to m}=(\nabla_X\boldsymbol{A})^{n+k\to m+k}$$

Proof

∇_{X^k} increases the rank of the tensor by k but operates elementwise

Property 2 (Composition)

Given a tensor $\boldsymbol{A}$ and two transpositions $k_1, k_2,\ldots,k_N$ and $j_1, j_2,\ldots,j_N$

$$\left(\mathrm{A}_{T(k_1,\,k_2,\ldots,k_N)}\right)_{T(j_1,\,j_2,\ldots,j_N)}=\mathrm{A}_{T(k_{j_1},\,k_{j_2},\ldots,k_{j_N})}$$

Proof

By definition?

Property 3 (Indexing)

Given a transpositon $\boldsymbol{A}_{T(k_1,\,k_2,\ldots,k_N)}$ and an ordering of the k_n, $\{o_n\}$

$$\left(A_{T(k_1,\,k_2,\ldots,k_N)}\right)_{i_1\,i_2\ldots\,i_N}=A_{i_{o_1}\,i_{o_2}\ldots\mathrm{i}_{o_N}}$$

(By ordering we mean a sequence such that $\forall_n\ k_{o_n}< k_{o_{n+1}}$)

Proof

We won't do a full proof, which would basically have to be done by induction by showing that it works for a single axis swap and then showing that if it works for n swaps it works for n+1

Assume that $\boldsymbol{A}$ has axes $\{a_n\}\Rightarrow A_{T(k_1,\,k_2,\ldots,k_N)}$ has axes $\{a_{k_n}\}$

When we ask for element $i_1\,i_2\ldots\,i_N$ we are asking, therefore, for element i_m of axis a_{k_m}

E.g. suppose $m = 5$, i_5 will be taken from a_{k_5} then suppose $k_5 = 2$

$\Rightarrow$ we want element i_5 from axis 2 in the original

$\Rightarrow$ we need to find an *ordering* of the $i_1\ i_2 \ldots\ i_N$ such that $o_2 = 5$

$\therefore$ we use the ordering of the $\{k_n\}$

Corr (Shift Indexing)

Given *a* shift $\boldsymbol{A}^{n\to m}$ WLOG assume $n < m$

$$(A^{n\to m})_{i_1,\ldots,\ i_n,\ldots,\ i_m,\ \ldots\ i_N} = A_{i_1,\ldots,\ i_m,\ i_n,\ \ldots,\ i_{m-1},\ i_{m+1},\ \ldots,\ i_N}$$

or in more prosaic terms, the indices shift in the opposite direction that the axes shift

Property 4 (Shift Compositions)

Given *a* double shift $(\boldsymbol{A}^{n\to m})^{l\to k}$

if $m{=}l$, $(\boldsymbol{A}^{n\to m})^{l\to k} = \boldsymbol{A}^{n\to k}$, otherwise there is no reduction to be had

Table of N=3 cases

Proof

WLOG we will assume $\boldsymbol{A}$ has dimension $N = \max\{n,\ m,\ l,\ k\} - \min\{n,\ m,\ l,\ k\}$

Case 1 : n<m<l< k

If $n{<}m{<}l{<}k$, we have two distinct uncomposable shift operations, $n{\to}m$ and $l{\to}k$

Note that in this case and only in this case (*I* think) the operations commute

Case 2: n<m=l<k

WLOG suppose $n{=}1$ and $k{=}N$

$$\begin{aligned}\left(A^{1\to m}\right)^{m\to N} &= \left(A_{T(2,\ \ldots,\ m-1,\ 1,\ m,\ \ldots,\ N)}\right)_{T(1,\ \ldots,\ m-1,\ m+1,\ \ldots,\ N,\ m)}\\ &= A_{T(2,\ \ldots,\ N,\ 1)}\\ &= A^{1\to N}\end{aligned}$$

Case 3: n<m & l<k & n=l

WLOG suppose $n{=}1$ and $k{=}N$

$$\begin{aligned}\left(A^{1\to m}\right)^{1\to N} &= \left(A_{T(2,\ \ldots,\ m-1,\ 1,\ m,\ \ldots,\ N)}\right)_{T(2,\ \ldots,\ N,\ 1)}\\ &= A_{T(3,\ \ldots,\ m-1,\ 1,\ m,\ \ \ldots,\ N,\ 2)}\end{aligned}$$

$\therefore$ Uncomposable in general

Case 4: n<m & l<k & m=k

WLOG suppose n=1 and k=N

$$\left(A^{1\to N}\right)^{l\to N}=\left(A_{T(2,\ \ldots,\ N,\ 1)}\right)_{T(1,\ \ldots,\ k-1,\ k+1,\ \ldots,\ N,\ k)}$$
$$=A_{T(2,\ \ldots,\ N,\ 1,\ k)}$$

∴ Uncomposable in general

Case 5: n>m & l<k & n=k

WLOG suppose m=1 and k=N

$$\left(A^{n\to 1}\right)^{l\to N}=A_{T(N,\ 1,\ \ldots,\ l-1,\ l+1,\ \ldots,\ N-1,\ l)}$$

∴ Uncomposable in general

Tensor Derivatives

Given a tensor $\mathbf{A}$, and some variable x upon which the elements of $\mathbf{A}$ may depend we'll define the derivative element-wise such that

$$\left(\frac{\partial}{\partial x}A\right)_{i_1\ i_2\ldots\ i_n}=\frac{\partial}{\partial x}A_{i_1\ i_2\ldots\ i_n}$$

We will also define derivative tensors such that given a vector of coordinates $\boldsymbol{x}$ and a scalar function f,

$$\nabla_x f=\begin{pmatrix}\frac{\partial}{\partial x_1}f & \frac{\partial}{\partial x_2}f & \cdots & \frac{\partial}{\partial x_n}f\end{pmatrix}$$

if instead of scalar f we have a tensor $\mathbf{A}$ everything is basically the same but we get the tensor of derivatives

Higher dimensional tensor derivative operators may also be defined, e.g. given another vector of coordinates $\mathbf{y}$ we can have

$$\nabla_{x,y}f=\begin{pmatrix}\frac{\partial^2}{\partial x_1\,\partial y_1}f & \frac{\partial^2}{\partial x_1\,\partial y_2}f & \cdots & \frac{\partial^2}{\partial x_1\,\partial y_m}f\\ \frac{\partial^2}{\partial x_2\,\partial y_1}f & & & \vdots\\ \vdots & & \ddots & \\ \frac{\partial^2}{\partial x_n\,\partial y_1}f & & \cdots & \frac{\partial^2}{\partial x_n\,\partial y_m}f\end{pmatrix}$$

Chain Rule

We'll define the chain rule over a tensor of derivatives in the standard kind of way by supplying a Jacobian for the transformation, e.g. given a vector of coordinates $\boldsymbol{x}$ and a function f and a set of coordinates upon which f depends, $\mathbf{y}$,

$$\nabla_x f = \nabla_x y \, \nabla_y f$$

The same basic form holds for higher dimensional derivative tensors, e.g. given a vector valued function $\mathbf{g}$

$$\nabla_x g = \nabla_x y \, \nabla_y g$$

And for a general derivative tensor for a tensor valued function $\mathbf{A}$

$$\nabla_x A = \nabla_x y \odot \nabla_y A$$

where $\odot$ is the tensor dot operation talked about later

We'll also note that this still works in a slightly different form with univariate derivatives

$$\frac{\partial}{\partial x_i} f = (\nabla_x y)_i \, \nabla_y f$$

Proof

Case 1:

$$\begin{aligned}
(\nabla_x f)_i &= \frac{\partial}{\partial x_i} f \\
&= \sum_{j=1}^{N} \frac{\partial y_j}{\partial x_i} \frac{\partial}{\partial y_j} f \\
&= (\nabla_x y)_i \cdot \nabla_y f
\end{aligned}$$

Case 2:

$$\begin{aligned}
(\nabla_x g)_{\mathrm{ij}} &= \left(\frac{\partial}{\partial x_i} g \right)_j \\
&= \frac{\partial}{\partial x_i} g_j \\
&= \sum_k \frac{\partial y_k}{\partial x_i} \frac{\partial}{\partial y_k} g_j \\
&= (\nabla_x y)_i \cdot \left(\nabla_y g_j \right)
\end{aligned}$$

Case 3:

Left to reader

Tensor Dot

Definition

We'll define the tensor dot $\odot$ to be a binary tensor operation that generalizes the dot product. It will operate recursively such that:

- Given a vector $\mathbf{a}$ of length N and tensor $\mathbf{B}$ with primary dimension N we'll define this such that

$$a \bigodot B = \sum_{i=1}^{N} a_i \, B_i$$

- Given a tensor $\mathbf{A}$ with primary dimension N and innermost dimension M and a tensor $\mathbf{B}$ with primary dimension M we'll define this such that

$$A \bigodot B = (A_1 \odot B \;\; A_2 \odot B \;\; \cdots \;\; A_N \odot B)$$

Property 1 (associativity)

Given $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ tensors,

$$(A \odot B) \odot C = A \odot (B \odot C)$$

Proof

LTR

Property 2 (dot product)

Given vectors $\boldsymbol{a}$, $\boldsymbol{b}$ of length N

$$a \odot b \;\; = \;\; a \cdot b$$

where $\cdot$ is the classic dot product

Proof

Basically just by definition

$$a \odot b = \sum_{i=1}^{N} a_i \, b_i = a \cdot b$$

Property 3 (vector matrix)

Given a vector $\boldsymbol{a}$ of length N and matrix $\boldsymbol{B}_{\mathrm{NxM}}$

$$a \odot B \ = \mathrm{aB}$$

where **aB** is interpreted in the normal sense of a matrix product

Proof

We start with

$$a \odot B = \sum_{i=1}^{N} a_i \, B_i$$

Given $1 \leq i \leq N$

$$a_i \, B_i = (a_i \, B_{\mathrm{i1}} \ \ a_i \, B_{\mathrm{i2}} \ \cdots \ \ a_i \, B_{\mathrm{iM}})$$

then

$$\sum_{i=1}^{N} a_i \, B_i = \left(\sum_{i=1}^{N} a_i \, B_{\mathrm{i1}} \ \ \sum_{i=1}^{N} a_i \, B_{\mathrm{i2}} \ \cdots \ \ \sum_{i=1}^{N} a_i \, B_{\mathrm{iM}} \right)$$
$$= (a \cdot B_{:1} \ \ a \cdot B_{:2} \ \cdots \ \ a \cdot B_{:M})$$

which is of course the classic vector matrix product

Property 4 (matrix product)

Given matrices $\boldsymbol{A}_{\mathrm{NxM}}$ and $\boldsymbol{B}_{\mathrm{MxL}}$

$$A \odot B \ = \mathrm{AB}$$

Proof

$$A \odot B = (A_1 \odot B \ \ A_2 \odot B \ \cdots \ \ A_N \odot B)$$

Given $1 \leq i \leq M$, then, by Property 2,

$$(A_1 \odot B \ \ A_2 \odot B \ \cdots \ \ A_N \odot B) = (A_1 \, B \ \ A_2 \, B \ \cdots \ \ A_N \, B)$$
$$= \begin{pmatrix} A_1 \cdot B_{:1} & A_1 \cdot B_{:2} & \cdots & A_1 \cdot B_{:L} \\ A_2 \cdot B_{:1} & & & \\ \vdots & & \ddots & \vdots \\ A_N \cdot B_{:1} & & \cdots & A_N \cdot B_{:L} \end{pmatrix}$$

which is how **AB** is defined

Property 5 (indexing)

Given $\boldsymbol{A}$, $\boldsymbol{B}$ tensors of dimension N and M, suppose k>0, we have

$$(A \odot B)_{i_1\, i_2 \ldots\, i_{N-1}\, j_2 \ldots j_M} = A_{i_1\, i_2 \ldots\, i_{N-1}} \cdot B_{:j_2 \ldots j_M} \tag{11}$$

$$(A \odot B)_{i_1\, i_2 \ldots\, i_{N-k}} = A_{i_1\, i_2 \ldots\, i_{N-1}} \odot B \tag{12}$$

$$(A \odot B)_{i_1\, i_2 \ldots\, i_{N-1}\, j_2 \ldots j_{M-k}} = A_{i_1\, i_2 \ldots\, i_{N-1}} \odot B_{:j_2 \ldots j_{M-k}:\ldots} \tag{13}$$

where : means to index every element along that axis

Intuitively this means that we simply distribute the indices from left to right until we have run out

Proof

(I proved this for an older statement first so the proof doesn't really directly correspond to what I stated, but is equivalent)

(1)

LTR

(2)

We will show this first for the case that k = N-1

$$\begin{aligned}(A \odot B)_i &= (A_1 \odot B \;\; A_2 \odot B \;\cdots\; A_N \odot B)_i \\ &= A_i \odot B\end{aligned}$$

Then we'll do this by induction, à la Property 6

Base Case

Suppose N=2, this reduces to (1), therefore it holds

Induction on N:

Suppose this holds for all $\boldsymbol{A}'$ with dimension $N-1$. Next as we have

$$A \odot B = (A_1 \odot B \;\; A_2 \odot B \;\cdots\; A_N \odot B)$$

and $\forall_i$ $\dim(A_i) = N-1$

$$\begin{aligned}(A \odot B)_{i_1\, i_2 \ldots\, i_{N-1}} &= (A_{i_1} \odot B)_{i_2 \ldots\, i_{N-1}} \\ &= (A_{i_1})_{i_2 \ldots\, i_{N-1}} \odot B \\ &= A_{i_1\, i_2 \ldots\, i_{N-1}} \odot B\end{aligned}$$

(3)

By (1) we have:

$$(A \odot B)_{i_1\, i_2 \ldots\, i_{N-1}\, j_2 \ldots j_M} = (A_{i_1\, i_2 \ldots\, i_{N-1}} \odot B)_{j_2 \ldots j_M}$$

Next note that $\dim(A_{i_1\, i_2 \ldots\, i_{N-1}}) = 1$, so

$$\begin{aligned}
(A_{i_1\, i_2 \ldots\, i_{N-1}} \odot B)_{j_2 \ldots j_M} &= \left(\sum_{k=1}^{N} (A_{i_1\, i_2 \ldots\, i_{N-1}})_k \, B_k \right)_{j_2 \ldots j_M} \\
&= \sum_{k=1}^{N} (A_{i_1\, i_2 \ldots\, i_{N-1} k} \, B_k)_{j_2 \ldots j_M} \text{ (by distribution of indexing)} \\
&= \sum_{k=1}^{N} A_{i_1\, i_2 \ldots\, i_{N-1} k} \, B_{k\, j_2 \ldots j_M} \text{(by commutation of indexing and scalar mul.)} \\
&= A_{i_1\, i_2 \ldots\, i_{N-1}} \cdot B_{:\, j_2 \ldots j_M}
\end{aligned}$$

Property 6 (dimension)

Given $\boldsymbol{A}$ and $\boldsymbol{B}$ tensors,

$$\dim(A \odot B) = \dim(A) + \dim(B) - 2$$

Proof

Lemma:

If $\dim(A) = 1$,

$$\dim(A \odot B) = \dim(B) - 1$$

Lemma Proof:

$$A \bigodot B = \sum_{i=1}^{N} A_i \, B_i$$

Given $\boldsymbol{A}'$, $\boldsymbol{B}'$ tensors with the same dimensions, we'll recall that

$$\dim(A' + B') = \dim(A') = \dim(B')$$

therefore

$$\begin{aligned}
\dim\left(A \bigodot B\right) &= \dim\left(\sum_{i=1}^{N} A_i \, B_i \right) \\
&= \dim(B_i) \\
&= \dim(B) - 1
\end{aligned}$$

QED

Returning to the main problem, we'll use induction on dim(A), taking the Lemma as our base case

Suppose that $\forall A'$ with $\dim(A')=n-1$, $\dim(A'\odot B)=n-1+\dim(B)-2$

Given A with $\dim(A) = n$,

$$A \bigodot B = (A_1\odot B \ A_2\odot B \ \cdots \ A_N\odot B)$$

Given $1\leq i\leq N$,

$$\begin{aligned}\dim(A_i\odot B) &= n-1+\dim(B)-2\\ &=n+\dim(B)-3\\ &=\dim(A)+\dim(B)-3\end{aligned}$$

Then as

$$A \bigodot B = (A_1\odot B \ A_2\odot B \ \cdots \ A_N\odot B)$$

since we have added a new outer dimension, we have

$$\begin{aligned}\forall_i \ \dim(A\odot B)&=\dim(A_i\odot B) + 1\\ &=\dim(A)+\dim(B)-3+1\\ &=\dim(A)+\dim(B)-2\end{aligned}$$

And then by induction we have this for all n

Property 7 (scalar multiplication)

Given $\boldsymbol{A}$, $\boldsymbol{B}$, tensors, c a scalar,

$$A\odot(\mathrm{cB})=c(A\odot B)$$

Proof

The general strategy for this proof would be the same as for Property 6, where we show it holds for tensors of dimension one and then generalize upwards via induction on the dimension of **A** or **C**. In the interest of time, we will simply show that it operates for dim(A)=1

Suppose $\dim(A) = 1$,

$$\begin{aligned}A \bigodot(\mathrm{c}B)&=\sum_{i=1}^{N} A_i(\mathrm{cB}_i)\\ &=c\sum_{i=1}^{N} A_i \, B_i\\ &=c(A\odot B)\end{aligned}$$

Property 8 (distributivity)

Given $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ tensors,

$$A \odot (B + C) = A \odot B + A \odot C$$

and

$$(A + B) \odot C = A \odot C + B \odot C$$

Proof

As in Property 7, we will simply show that it operates for tensors of dimension one.

Suppose $\dim(A) = 1$,

$$\begin{aligned} A \odot (B + C) &= \sum_{i=1}^{N} A_i * (B + C)_i \\ &= \sum_{i=1}^{N} A_i * (B_i + C_i) \\ &= \sum_{i=1}^{N} A_i \, B_i + A_i \, C_i \\ &= \sum_{i=1}^{N} A_i \, B_i + \sum_{i=1}^{N} A_i \, C_i \\ &= A \odot B + A \odot C \end{aligned}$$

Further, now suppose $\dim(B) = 1$,

$$\begin{aligned} (A + B) \odot C &= \sum_{i=1}^{N} (A + B)_i \, C_i \\ &= \sum_{i=1}^{N} (A_i + B_i) \, C_i \\ &= \sum_{i=1}^{N} A_i \, C_i + \sum_{i=1}^{N} B_i \, C_i \\ &= A \odot C + B \odot C \end{aligned}$$

Property 9 (product rule)

9.a: scalar derivatives

Given $\boldsymbol{A}$, $\boldsymbol{B}$ tensors, x some variable upon which the elements of $\boldsymbol{A}$ and $\boldsymbol{B}$ may depend

$$\frac{\partial}{\partial x} A \odot B = \left(\frac{\partial}{\partial x} A\right) \odot B + A \odot \left(\frac{\partial}{\partial x} B\right)$$

Proof

As in Property 7, we will simply show that it operates for tensors of dimension one.

Suppose $\dim(A) = 1$,

$$\begin{aligned}\frac{\partial}{\partial x} A \odot B &= \frac{\partial}{\partial x} \sum_{i=1}^{N} A_i \, B_i \\ &= \sum_{i=1}^{N} \frac{\partial}{\partial x} (A_i \, B_i)\end{aligned}$$

then as the derivative will operate element−wise on the tensor—

is this obvious?—we will have

$$\begin{aligned}\frac{\partial}{\partial x} (A_i \, B_i)_{j_1 \, j_2 \, \ldots \mathrm{j}_n} &= \frac{\partial}{\partial x} A_i (B_i)_{j_1 \, j_2 \, \ldots \mathrm{j}_n} \\ &= \left(\frac{\partial}{\partial x} A_i\right) (B_i)_{j_1 \, j_2 \, \ldots \mathrm{j}_n} + A_i \left(\frac{\partial}{\partial x} (B_i)_{j_1 \, j_2 \, \ldots \mathrm{j}_n}\right)\end{aligned}$$

therefore

$$\begin{aligned}\frac{\partial}{\partial x} A \odot B &= \sum_{i=1}^{N} \frac{\partial}{\partial x} (A_i \, B_i) \\ &= \sum_{i=1}^{N} \left(\frac{\partial}{\partial x} A_i\right) B_i + A_i \left(\frac{\partial}{\partial x} B_i\right) \\ &= \sum_{i=1}^{N} \left(\frac{\partial}{\partial x} A_i\right) B_i + \sum_{i=1}^{N} A_i \left(\frac{\partial}{\partial x} B_i\right) \\ &= \left(\frac{\partial}{\partial x} A\right) \odot B + A \odot \left(\frac{\partial}{\partial x} B\right)\end{aligned}$$

9.b: vector derivatives

For vector derivatives this has *a* slightly different form

$$\nabla_X (A \odot B) = (\nabla_X A) \odot B + (A \odot (\nabla_{X_i} B))_{i=1\ldots}$$

or, using transposition notation, if A is of rank a and B is of rank b

$$\nabla_X (A \odot B) = (\nabla_X A) \odot B + \left(A \odot (\nabla_X B)^{2 \to 1}\right)^{a \to 1}$$

Proof

Property 10 (transposition)

Given $\boldsymbol{A}$ with axes $\{a_n\}_{n=1}^{N+1}$ $\boldsymbol{B}$ with asxes $\{b_m\}_{m=1}^{M+1}$ and some transposition $\{k_l\}$,

if $\forall_i\ i \leq N \Rightarrow k_i \leq N$ and $\forall_j\ j > N \Rightarrow k_j > N$,

$$(A \odot B)_{T(\{k_l\})} = A_{T(\{k_l | l \leq N\})} \odot B_{T(\{k_l - N | l > N\})}$$

if $\forall_i\ i > M \Rightarrow k_i \leq M$ and $\forall_j\ j \leq M \Rightarrow k_j > M$,

$$(A \odot B)_{T(\{k_l\})} = B_{T(\{k_l | l \leq M\})} \odot A_{T(\{k_l - M | l > M\})}$$

otherwise $\nexists$ $\{i_l\}$, $\{j_l\}$ such that

$$(A \odot B)_{T(\{k_l\})} = A_{T(\{i_l\})} \odot B_{T(\{j_l\})} \text{ or } (A \odot B)_{T(\{k_l\})} = B_{T(\{j_l\})} \odot A_{T(\{i_l\})}$$

Relation to Einstein Summation Notation (einsum)

Basically a tensor dot is a simpler version of einsum where we only contract along the inner dimensions, e.g.

Given tensors $\boldsymbol{A}_{\text{ijkl}}$ and $\boldsymbol{B}_{\text{l}\alpha\beta\gamma}$ we have

$$A \odot B = A_{\text{ijkl}}\, B_{\text{l}\alpha\beta\gamma} = (\text{AB})_{\text{ijk}\alpha\beta\gamma}$$

Note that when written like this the validity of Properties 2-6 are evident "by notation"

Notations

Since these derivatives involve a large number of $\nabla \cdot$, $x^{\dot{\rightarrow}\cdot}$, and $\cdot \odot \cdot$ operations we'll introduce a convenient shorthand where

$$\nabla_x A \equiv A_x$$
$$A^{a \to b} \equiv A^{a:b}$$
$$A \odot B \equiv AB$$

This means the product rule for tensor derivatives can be expressed as

$$\nabla_x (A \odot B) = A_x\, B + \left(A (B_x)^{2:1}\right)^{a:1}$$

Contraction Notation

We'll introduce a convenient notation to cut down on the number of parens we need where we define

$$A \langle n \rangle B = AB^{n:1}$$

to be the tensor dot operation applied along the n^{th} axis of B. Notably

$$A\langle 1\rangle B=\text{AB}$$

we'll also define

$$A\langle m,\ n\rangle B=A^{m:N_A}\ B^{n:1}$$

to be the contraction of A and B along axes m and n.

To cut down on parentheses, these will be further defined to be right-associative

Code

Potential Derivatives

$$\nabla_q Q = \nabla_q R \nabla_R Q$$
$$\nabla_{q^2} Q = \nabla_q R \nabla_R Q$$

With these rules in place we can write code to do this differentiation automatically, but to show the procedure out we'll do up to the fourth derivatives by hand

Derivatives to Target

Basically we'll want

$$\nabla_Q V = \left(\frac{\partial}{\partial Q_1} \cdots \frac{\partial}{\partial Q_m}\right) V$$
$$\nabla_{Q^2} V = \begin{pmatrix} \frac{\partial^2}{\partial Q_1 Q_1} V & \cdots & \frac{\partial^2}{\partial Q_1 Q_m} V \\ \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial Q_1 Q_m} V & \cdots & \frac{\partial^2}{\partial Q_m Q_m} V \end{pmatrix}$$

and higher up to order 4.

We don't actually have these directly, though, so we'll need to build them using the chain rule a bunch from the components we do have.

As a simple initial example, we want

$$\nabla_Q V = \left(\frac{\partial}{\partial Q_1} \cdots \frac{\partial}{\partial Q_m}\right) V$$

and we can look at this element wise as

$$\left(\nabla_Q V\right)_i = \frac{\partial}{\partial Q_i} V = \sum_j \frac{\partial x_j}{\partial Q_i} \frac{\partial}{\partial x_j} V$$

General Layout

We have basically two types of terms here:

$$\nabla_Q x = \begin{pmatrix} \frac{\partial x_1}{\partial Q_1} & \cdots & \frac{\partial x_n}{\partial Q_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_1}{\partial Q_m} & \cdots & \frac{\partial x_n}{\partial Q_m} \end{pmatrix}$$

which is the Jacobian going from Cartesian coordinates to internal coordinate normal modes. We will also have higher order Jacobian type things.

$$\nabla_x V = \begin{pmatrix} \frac{\partial}{\partial x_1} & \cdots & \frac{\partial}{\partial x_n} \end{pmatrix} V$$

which is the gradient of V with respect to the Cartesian coordinates. We'll also have higher order derivatives here.

I will denote higher derivatives by adding an exponent to the dependent variable, e.g.

$$\nabla_{Q^2} x = \begin{pmatrix} \frac{\partial}{\partial Q_1} \begin{pmatrix} \frac{\partial x_1}{\partial Q_1} & \cdots & \frac{\partial x_n}{\partial Q_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_1}{\partial Q_m} & \cdots & \frac{\partial x_n}{\partial Q_m} \end{pmatrix} & \cdots & \frac{\partial}{\partial Q_m} \begin{pmatrix} \frac{\partial x_1}{\partial Q_1} & \cdots & \frac{\partial x_n}{\partial Q_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_1}{\partial Q_m} & \cdots & \frac{\partial x_n}{\partial Q_m} \end{pmatrix} \end{pmatrix} =$$

$$\begin{pmatrix} \begin{pmatrix} \frac{\partial^2 x_1}{\partial Q_1 Q_1} & \cdots & \frac{\partial^2 x_n}{\partial Q_1 Q_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 x_1}{\partial Q_1 Q_m} & \cdots & \frac{\partial^2 x_n}{\partial Q_1 Q_m} \end{pmatrix} & \cdots & \begin{pmatrix} \frac{\partial^2 x_1}{\partial Q_m Q_1} & \cdots & \frac{\partial^2 x_n}{\partial Q_m Q_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 x_1}{\partial Q_m Q_m} & \cdots & \frac{\partial^2 x_n}{\partial Q_m Q_m} \end{pmatrix} \end{pmatrix}$$

is the 3D tensor of second derivatives of Cartesian coordinates with respect to pairs of internal coordinate normal modes.

Derivatives

A worthwhile thing to keep in mind: the number of terms in an expansion grows exponentially as the Bell Numbers. It's a fun little graph proof, if you want to see it. This also shows that we should be able to write a general combinatoric algorithm for this expansion.

Firsts

These are unchanged

$$\nabla_Q V = \nabla_Q X \, \nabla_X V$$

Seconds

These have two transpositions removed and rewritten as tensor contractions

$$\begin{aligned}\nabla_{Q^2} V &= \nabla_{Q^2} X \, \nabla_X V + \left(\nabla_Q X \, \langle 2 \rangle \, \nabla_Q X \, \nabla_{X^2} V\right)^{2:1} \\ &= \nabla_{Q^2} X \, \nabla_X V + \nabla_Q X \, \langle 2 \rangle \, \nabla_Q X \, \langle 2 \rangle \, \nabla_{X^2} V\end{aligned}$$

Thirds

These still get gross to start, since we've got 5 elements

$$\begin{aligned}\nabla_{Q^3} V = \nabla_{Q^3} X \, \nabla_X V &+ \left(\nabla_{Q^2} X \, \langle 2 \rangle \, \nabla_Q X \, \nabla_{X^2} V\right)^{3:1} \\ &+ \nabla_{X^2} Q \, \langle 2 \rangle \, \nabla_Q X \, \langle 2 \rangle \, \nabla_{X^2} V \\ &+ \left(\nabla_Q X \, \langle 3 \rangle \, \nabla_{Q^2} X \, \langle 2 \rangle \, \nabla_{X^2} V\right)^{2:1} + \left(\nabla_Q X \, \langle 3 \rangle \left(\nabla_Q X \, \langle 3 \rangle \, \nabla_Q X \, \nabla_{X^2} V\right)^{2:1}\right)^{2:1}\end{aligned}$$

we can simplify this by noting that, since $\nabla_{X^2} V$ is symmetric

$$\begin{aligned}\left(\nabla_{Q^2} X \, \langle 2 \rangle \, \nabla_Q X \, \nabla_{X^2} V\right)^{3:1} &= \left(\nabla_{Q^2} X \, \langle 2 \rangle \, \nabla_Q X \, \langle 2 \rangle \, \nabla_{X^2} V\right)^{3:1} \\ \left(\nabla_Q X \, \langle 3 \rangle \, \nabla_{Q^2} X \, \langle 2 \rangle \, \nabla_{X^2} V\right)^{2:1} &= \left(\nabla_{Q^2} X \, \langle 2 \rangle \, \nabla_Q X \, \langle 2 \rangle \, \nabla_{X^2} V\right)^{3:2}\end{aligned}$$

and so we really only have one matrix multiplication to do

$$\nabla_{Q^2, Q} \nabla_{x^2} V = \nabla_{Q^2} X \, \langle 2 \rangle \, \nabla_Q X \, \langle 2 \rangle \, \nabla_{X^2} V$$

giving us

$$\begin{aligned}\nabla_{Q^3} V = \nabla_{Q^3} X \, \nabla_X V &+ \nabla_{Q^2, Q} \nabla_{x^2} V \\ &+ \left(\nabla_{Q^2, Q} \nabla_{x^2} V\right)^{3:1} \\ &+ \left(\nabla_{Q^2, Q} \nabla_{x^2} V\right)^{3:2} + \nabla_Q X \, \langle 3 \rangle \, \nabla_Q X \, \langle 3 \rangle \, \nabla_Q X \, \langle 3 \rangle \, \nabla_{X^3} V\end{aligned}$$

Fourths

We need 15 elements here, but 9 of them are going to come from the $\nabla_{Q^2, Q} \nabla_{x^2} V$ term, so we'll do that first

$$\nabla_Q \left(\nabla_{Q^2, Q} \nabla_{x^2} V\right) = \nabla_Q \left(\nabla_{Q^2} X \, \langle 2 \rangle \, \nabla_Q X \, \langle 2 \rangle \, \nabla_{X^2} V\right)$$

$$
\begin{aligned}
&= \nabla_{Q^3} X \langle 2 \rangle \nabla_Q X \langle 2 \rangle \nabla_{X^2} V \\
&\quad + \left(\nabla_{Q^2} X \langle 3 \rangle \nabla_{Q^2} X \langle 2 \rangle \nabla_{X^2} V \right)^{3:1} \\
&\quad + \left(\nabla_{Q^2} X \langle 3 \rangle \left(\nabla_Q X \langle 3 \rangle \nabla_Q X \nabla_{X^3} V \right)^{2:1} \right)^{3:1} \\
&= \nabla_{Q^3} X \langle 2 \rangle \nabla_Q X \langle 2 \rangle \nabla_{X^2} V \\
&\quad + \left(\nabla_{Q^2} X \langle 3 \rangle \nabla_{Q^2} X \langle 2 \rangle \nabla_{X^2} V \right)^{3:1} \\
&\quad + \left(\nabla_{Q^2} X \langle 3 \rangle \nabla_Q X \langle 2 \rangle \nabla_Q X \langle 3 \rangle \nabla_{X^3} V \right)^{3:1} \\
&= \nabla_{Q^3} X \langle 2 \rangle \nabla_Q X \langle 2 \rangle \nabla_{X^2} V \\
&\quad + \left(\nabla_{Q^2} X \langle 3 \rangle \nabla_{Q^2} X \langle 2 \rangle \nabla_{X^2} V \right)^{3:1} \\
&\quad + \left(\nabla_{Q^2} X \langle 3 \rangle \nabla_Q X \langle 2 \rangle \nabla_Q X \langle 3 \rangle \nabla_{X^3} V \right)^{3:1}
\end{aligned}
$$

then when we plug in our transpositions from before we get

$$
\nabla_Q \left(\nabla_{Q^2,Q} \nabla_{x^2} V + \left(\nabla_{Q^2,Q} \nabla_{x^2} V \right)^{3:1} + \left(\nabla_{Q^2,Q} \nabla_{x^2} V \right)^{3:2} \right) =
$$

$$
\nabla_Q \left(\nabla_{Q^2,Q} \nabla_{x^2} V \right) + \nabla_Q \left(\nabla_{Q^2,Q} \nabla_{x^2} V \right)^{4:2} + \nabla_Q \left(\nabla_{Q^2,Q} \nabla_{x^2} V \right)^{4:3}
$$

which gives us three terms & three transpositions, resulting in nine terms overall

Then we can easily do the first term

$$
\nabla_Q \left(\nabla_{Q^3} X \nabla_X V \right) = \nabla_{Q^4} X \nabla_X V + \nabla_Q X \langle 4 \rangle \nabla_{Q^3} X \langle 2 \rangle \nabla_{X^2} V
$$

and with a bit more annoyance we can get the final term

$$
\begin{aligned}
\nabla_Q \left(\nabla_Q X \langle 3 \rangle \nabla_Q X \langle 3 \rangle \nabla_Q X \langle 3 \rangle \nabla_{X^3} V \right) = {} & \nabla_{Q^2} X \langle 3 \rangle \nabla_Q X \langle 3 \rangle \nabla_Q X \langle 3 \rangle \nabla_{X^3} V \\
& + \nabla_Q X \langle 4 \rangle \left(\nabla_{Q^2} X \langle 3 \rangle \nabla_Q X \langle 3 \rangle \nabla_{X^3} V \right)^{2:1} \\
& + \nabla_Q X \langle 4 \rangle \left(\nabla_Q X \langle 4 \rangle \nabla_{Q^2} X \langle 3 \rangle \nabla_{X^3} V \right)^{2:1} \\
& + \nabla_Q X \langle 4 \rangle \left(\nabla_Q X \langle 4 \rangle \left(\nabla_Q X \langle 4 \rangle \nabla_Q X \nabla_{X^4} V \right)^{2:1} \right)^{2:1}
\end{aligned}
$$

we'll do some total term gathering in a bit, but we'll first note that by some rearrangements and symmetry in $\nabla_{X^4} V$ we can write

$$
\nabla_Q X \langle 4 \rangle \left(\nabla_Q X \langle 4 \rangle \left(\nabla_Q X \langle 4 \rangle \nabla_Q X \nabla_{X^4} V \right)^{2:1} \right)^{2:1} = \nabla_Q X \langle 4 \rangle \nabla_Q X \langle 4 \rangle \nabla_Q X \langle 4 \rangle \nabla_Q X \langle 4 \rangle \nabla_{X^4} V
$$

Finally, we'll gather all the terms we have here, since we only have five distinct types

$$
\nabla_{Q^4} \nabla_X V = \nabla_{Q^4} X \nabla_X V
$$

$$
\begin{aligned}
\nabla_{Q^2\ Q^2} \nabla_{X^2} V = {} & \left(\nabla_{Q^2} X \langle 3 \rangle \nabla_{Q^2} X \langle 2 \rangle \nabla_{X^2} V \right)^{3:1} \\
& + \left(\nabla_{Q^2} X \langle 3 \rangle \nabla_{Q^2} X \langle 2 \rangle \nabla_{X^2} V \right)^{3:1\,4:2} \\
& + \left(\nabla_{Q^2} X \langle 3 \rangle \nabla_{Q^2} X \langle 2 \rangle \nabla_{X^2} V \right)^{3:1\,4:3}
\end{aligned}
$$

$$\begin{aligned}\nabla_{Q^3\,Q}\nabla_{X^2}V =& \nabla_Q X\langle 4\rangle\,\nabla_{Q^3}X\langle 2\rangle\,\nabla_{X^2}V\\ &+\nabla_{Q^3}X\langle 2\rangle\,\nabla_Q X\langle 2\rangle\,\nabla_{X^2}V\\ &+\left(\nabla_{Q^3}X\langle 2\rangle\,\nabla_Q X\langle 2\rangle\,\nabla_{X^2}V\right)^{4:2}\\ &+\left(\nabla_{Q^3}X\langle 2\rangle\,\nabla_Q X\langle 2\rangle\,\nabla_{X^2}V\right)^{4:3}\\ =&\nabla_{Q^3}X\langle 2\rangle\,\nabla_Q X\langle 2\rangle\,\nabla_{X^2}V\\ &+\left(\nabla_{Q^3}X\langle 2\rangle\,\nabla_Q X\langle 2\rangle\,\nabla_{X^2}V\right)^{4:1}\\ &+\left(\nabla_{Q^3}X\langle 2\rangle\,\nabla_Q X\langle 2\rangle\,\nabla_{X^2}V\right)^{4:2}\\ &+\left(\nabla_{Q^3}X\langle 2\rangle\,\nabla_Q X\langle 2\rangle\,\nabla_{X^2}V\right)^{4:3}\end{aligned}$$

$$\begin{aligned}\nabla_{Q^2\,QQ}\nabla_{X^3}V =& \nabla_{Q^2}X\langle 3\rangle\,\nabla_Q X\langle 3\rangle\,\nabla_Q X\langle 3\rangle\,\nabla_{X^3}V\\ &+\left(\nabla_{Q^2}X\langle 3\rangle\,\nabla_Q X\langle 3\rangle\,\nabla_Q X\langle 3\rangle\,\nabla_{X^3}V\right)^{3:1}\\ &+\left(\nabla_{Q^2}X\langle 3\rangle\,\nabla_Q X\langle 3\rangle\,\nabla_Q X\langle 3\rangle\,\nabla_{X^3}V\right)^{3:1\,4:2}\\ &+\left(\nabla_{Q^2}X\langle 3\rangle\,\nabla_Q X\langle 3\rangle\,\nabla_Q X\langle 3\rangle\,\nabla_{X^3}V\right)^{3:1\,4:3}\\ &+\left(\nabla_{Q^2}X\langle 3\rangle\,\nabla_Q X\langle 3\rangle\,\nabla_Q X\langle 3\rangle\,\nabla_{X^3}V\right)^{1:3}\\ &+\left(\nabla_{Q^2}X\langle 3\rangle\,\nabla_Q X\langle 3\rangle\,\nabla_Q X\langle 3\rangle\,\nabla_{X^3}V\right)^{1:3\,1:3}\end{aligned}$$

$$\nabla_{QQQQ}\nabla_{X^4}V=\nabla_Q X\langle 4\rangle\,\nabla_Q X\langle 4\rangle\,\nabla_Q X\langle 4\rangle\,\nabla_Q X\langle 4\rangle\,\nabla_{X^4}V$$

Mixed Derivatives

We don't actually start out with stuff like $\nabla_{X^4}V$, instead having things like $\nabla_{q^2}\nabla_{x^2}V$, and we need to account for that in the derivs. This will enter only in the cubic and quartic force terms, giving us stuff like

$$\begin{aligned}\nabla_Q\nabla_{X^2}V=&\nabla_Q q\,\nabla_q\nabla_{X^2}V\\ \nabla_{Q^2}\nabla_{X^2}V=&\nabla_{Q^2}q\,\nabla_q\nabla_{X^2}V+\nabla_Q q\langle 2\rangle\,\nabla_Q q\langle 2\rangle\,\nabla_{q^2}\nabla_{X^2}V\end{aligned}$$

then we need to back-convert all of our $\nabla_{X^3}V$-like terms into ones involving the mixed derivatives we start with

$$\begin{aligned}\nabla_Q\nabla_{X^2}V=&\nabla_Q X\,\nabla_{X^3}V\\ \nabla_{Q^2}\nabla_{X^2}V=&\nabla_{Q^2}X\,\nabla_{X^3}V+\nabla_Q X\langle 2\rangle\,\nabla_Q X\,\nabla_{X^4}V\end{aligned}$$

meaning some of our terms disappear/simplify out

Alternately if we're trying to translate unmixed derivatives into mixed ones we have

$$\nabla_{Q^2}\nabla_{X^2}V=\nabla_Q X\langle 2\rangle\,\nabla_Q X\langle 1\rangle\,\nabla_{X^4}V+\nabla_{Q^2}X\,\nabla_{X^3}V$$

Firsts & Seconds

Thirds

Here we just need to change up one term, giving us

$$\begin{aligned}\nabla_{Q^3} V = \nabla_{Q^3} X \, \nabla_X V &+ \nabla_{Q^2,Q} \nabla_{x^2} V \\ &+ \left(\nabla_{Q^2,Q} \nabla_{x^2} V\right)^{3:1} \\ &+ \left(\nabla_{Q^2,Q} \nabla_{x^2} V\right)^{3:2} + \nabla_Q X \, \langle 3 \rangle \, \nabla_Q X \, \langle 3 \rangle \, \nabla_Q \nabla_{X^2} V\end{aligned}$$

Fourths

Diagonal Elements

Original Vs. New Stuff

General Case

We don't want to write more terms out by hand, so we note that in general we have

$$\nabla_Q\left(\nabla_{Q^n} X \, \langle i,j \rangle \, \nabla_{X^m} V\right) = \nabla_{Q^{n+1}} X \, \langle i+1,j \rangle \, \nabla_{X^m} V + \nabla_Q X \, \langle 2,\ n \rangle \left(\nabla_{Q^n} X \, \langle i,j+1 \rangle \, \nabla_{X^{m+1}} V\right)$$

which we can see by considering that $\langle i,\ j \rangle$ can be written as a matrix multiplication where axis i is shifted to the end of $\nabla_{Q^n} X$ and axis j is shifted to the start of $\nabla_{X^{m+1}} V$ and the two are then reshaped down to and matrix multiplied together and so the regular product rule properties apply, but every time we add an axis the contracted axes shift by one. The general form for this is given by

$$\nabla_Q(A \, \langle \alpha,\ \beta \rangle \, B) = \nabla_Q A \, \langle \alpha+1,\ \beta \rangle \, B + \left(A \, \langle \alpha,\ \beta+1 \rangle \, \nabla_Q B\right)^{n \to 1}$$

but it's almost easier to understand in the concrete context

$$\nabla_Q\left(\ \nabla_{Q^n} X \, \langle i,j \rangle \, \nabla_{X^m} V\right) = \nabla_Q\left(\nabla_{Q^n} X\right) \langle i+1,\ j \rangle \, \nabla_{X^m} V + \left(\nabla_{Q^n} X \, \langle i,\ j+1 \rangle \, \nabla_Q(\nabla_{X^m} V)\right)^{n \to 1}$$

then we know elementwise that

$$\begin{aligned}\nabla_Q\left(\nabla_{Q^n} X\right) &= \nabla_{Q^{n+1}} X \\ \nabla_Q(\nabla_{X^m} V) &= \nabla_Q X \, \nabla_{X^{m+1}} V\end{aligned}$$

meaning

$$\nabla_Q\left(\ \nabla_{Q^n} X \, \langle i,j \rangle \, \nabla_{X^m} V\right) = \nabla_Q\left(\nabla_{Q^n} X\right) \langle i+1,\ j \rangle \, \nabla_{X^m} V + \left(\nabla_{Q^n} X \, \langle i,\ j+1 \rangle \, \left(\nabla_Q X \, \nabla_{X^{m+1}} V\right)\right)^{n \to 1}$$

and then we can use the fact that tensor contractions along different axes commute up to a transpose (although associativity is still certainly not a thing) so

$$\nabla_{Q^n} X \, \langle i,\ j+1 \rangle \, \left(\nabla_Q X \, \nabla_{X^{m+1}} V\right) = \left(\nabla_Q X \, \langle 2,\ n \rangle \, \nabla_{Q^n} X \, \langle i,\ j+1 \rangle \, \nabla_{X^{m+1}} V\right)^{1 \to n}$$

and finally the axis shifts cancel, giving us the expected result.

This is useful as it allows us to write simple code to handle terms as long as we don't worry too much about efficiency. The efficiency can be recovered later by reducing the computation graph over unique contractions. The final complication is dealing with the axis shifts we'll pick up as we expand this out further. The key here will be to note that in general we have, assuming $i<j$

$$A\,\langle\alpha,\,\beta\rangle\,B^{i\to j} = \begin{cases} (A\,\langle\alpha,\,\beta\rangle\,B)^{\alpha-2+i\to\alpha-2+j} & \beta<i \\ (A\,\langle\alpha,\,\beta+1\rangle\,B)^{\alpha-1+i\to\alpha-2+j} & \beta<j \\ A\,\langle\alpha,\,i\rangle\,B & \beta=j \\ (A\,\langle\alpha,\,\beta\rangle\,B)^{\alpha-1+i\to\alpha-1+j} & \beta>j \end{cases}$$

or the other way around

$$A^{i\to j}\,\langle\alpha,\,\beta\rangle\,B = \begin{cases} (A\,\langle\alpha,\,\beta\rangle\,B)^{i\to j} & \alpha>j \\ A\,\langle i,\,\beta\rangle\,B & \alpha=j \\ (A\,\langle\alpha+1,\,\beta\rangle\,B)^{i\to j-1} & \alpha\geq i \\ (A\,\langle\alpha,\,\beta\rangle\,B)^{i-1\to j-1} & \alpha<i \end{cases}$$

Next we can note that for $\left(A^{i\to j}\right)^{k\to l}$, assuming $i>j$ and $k>l$, we can rewrite this in a canonically ordered form as

$$\left(A^{i\to j}\right)^{k\to l} = \begin{cases} A & j=k \text{ and } l=i \\ A^{i\to l} & j=k \\ \left(A^{k\to l}\right)^{i\to j} & k<i \text{ and } l<i \\ \left(A^{k\to l}\right)^{i-1\to j} & k<i \text{ and } l<j \\ \left(A^{k\to l}\right)^{i-1\to j-1} & k<i \text{ and } l>j \\ \left(A^{i\to j}\right)^{k\to l} & \text{else} \end{cases}$$

where we can also use the fact that if we have $\left(A^{i\to j}\right)^{i\to j i\to j\ldots}$ where we iterate $j-i+1$ times we get the identity. Similarly, given $j<i$ we have $A^{i\to j} = \left(A^{j\to i}\right)^{j\to i j\to i\ldots}$ where we iterate exactly $j-i$ times (you can see how these two come together nicely). With canonical ordering this mean we're hopefully going to get fully simplified terms.

Mixed Term

The most important term to be able to simplify out for are expandable as

$$\nabla_Q \nabla_{X^2} V = \nabla_Q X\,\nabla_{X^3} V$$
$$\nabla_{Q^2} \nabla_{X^2} V = \nabla_{Q^2} X\,\nabla_{X^2} V + \nabla_Q X\,\langle 2\rangle\,\nabla_Q X\,\nabla_{X^4} V$$

and so we can factor these terms out

$$\nabla_{Q^2} \nabla_{X^2} V = \nabla_{Q^2} \nabla_{X^2} V$$

G-Matrix Derivatives

Helpful Notes

When reading *Wilson, Decius, & Cross* (WDC) the authors introduce some odd terminology at least according to modern notation. First off, they define their internal coordinates by S_i which, like, fine. I'm gonna use r_i, because that's more in line with that we do.

Then given a reference structure which I'll call $\boldsymbol{S}_e$ and whose existence they allow to be implicit, they approximate the internals up to first order by Cartesians by an equation that they write as

$$S_t = \sum_{i=1}^{N} B_{ti}\,\xi_i \;,\; t = 1,\, 2,\, \ldots,\, M$$

where N and M are the number of Cartesian coordinates and number of respective internal coordinates

Here we have ξ_i being the Cartesian displacement. I'd usually call it Δx_i. And the B term is just the derivative of the internal with respect to the Cartesians. This can be much stated more cleanly by saying that we'll *represent* S_t as

$$S_t\!: \left(\frac{\partial S_t}{\partial x_1}\,\Delta x_1,\; \frac{\partial S_t}{\partial x_2}\,\Delta x_2,\; \ldots,\; \frac{\partial S_t}{\partial x_{3N}}\,\Delta x_{3N}\right) = \left(\frac{\partial S_t}{\partial x_i}\,\Delta x_i\right)_{i=1,\ldots}$$

This is saying that we'll linearly approximate our internal as a Cartesian displacement. We won't actually use this definition to evaluate the our internals, but these derivatives are important as this is how we get the B matrix that is implicitly referred to in the text

$$B = \begin{pmatrix} \dfrac{\partial S_1}{\partial x_1} & \dfrac{\partial S_1}{\partial x_2} & \cdots & \dfrac{\partial S_1}{\partial x_N} \\ \dfrac{\partial S_2}{\partial x_1} & \dfrac{\partial S_2}{\partial x_2} & & \\ \vdots & & \ddots & \vdots \\ \dfrac{\partial S_M}{\partial x_1} & & \cdots & \dfrac{\partial S_M}{\partial x_N} \end{pmatrix}$$

For anyone who's taken intro calc or ever transformed from Cartesian to polar coordinates to do an integral, this is just the Jacobian matrix and I'm not sure why *WDC* don't refer to it as such.

Returning to the notation that I prefer (and introducing a slightly odd gradient notation), we have

$$\nabla_{\mathrm{X}}\mathrm{R}=\begin{pmatrix} \frac{\partial r_1}{\partial x_1} & \frac{\partial r_1}{\partial x_2} & \cdots & \frac{\partial r_1}{\partial x_N} \\ \frac{\partial r_2}{\partial x_1} & \frac{\partial r_2}{\partial x_2} & & \\ \vdots & & \ddots & \vdots \\ \frac{\partial r_M}{\partial x_1} & & \cdots & \frac{\partial r_M}{\partial x_N} \end{pmatrix}$$

Next, in this notation we're given that the G-matrix is defined as

$$G_{ij}=\sum_{\alpha=1}^{N}\frac{1}{m_\alpha}\,\frac{\partial}{\partial x_\alpha}r_i\,\frac{\partial}{\partial x_\alpha}r_j$$

This is all well and good, but we can write this in a more intuitive way. Letting

$$\nabla_{\mathrm{X}}r_i=\left(\frac{\partial r_i}{\partial x_\alpha}\,\Delta x_\alpha\right)_{\alpha=1,\ldots,3N}$$

we'll define a mass-weighted form of this by

$$\nabla_{\mathrm{Y}}r_i=\left(\frac{1}{\sqrt{m_\alpha}}\,\frac{\partial r_i}{\partial x_\alpha}\,\Delta x_\alpha\right)_{\alpha=1,\ldots,3N}$$

the notation for which is motivated by the mass-weighted Coordinate transformation

$$y_\alpha=\sqrt{m_\alpha}\;x_\alpha$$

taking all of this together we get

$$G_{ij}=\nabla_{\mathrm{Y}}r_i\cdot\nabla_{\mathrm{Y}}r_j$$

This type of definition of a matrix element $\left(\text{vector}_i\cdot\text{vector}_j\right)$ tells us that our matrix can be constructed by a matrix multiplication. In this case we have

$$G=(\nabla_{\mathrm{Y}}\mathrm{R})^{\mathrm{T}}\,(\nabla_{\mathrm{Y}}\mathrm{R})$$

where we can build $\nabla_{\mathrm{Y}}\mathrm{R}$ by

$$\nabla_{\mathrm{Y}}\mathrm{R}=(\nabla_{\mathrm{X}}\mathrm{R})\,\mathrm{M}^{-\frac{1}{2}}$$

with

$$\mathrm{M}=\begin{pmatrix} m_1 & & & \\ & m_2 & & \\ & & \ddots & \\ & & & m_{3N} \end{pmatrix}$$

Sample Transform if G initially in internals

$$
\begin{aligned}
G_{ij} &= \sum_{\alpha=1}^{N} \frac{1}{m_\alpha} \frac{\partial}{\partial x_\alpha} q_i \frac{\partial}{\partial x_\alpha} q_j \\
&= \sum_{\alpha=1}^{N} \frac{1}{m_\alpha} \frac{\partial}{\partial x_\alpha} \left(\sum_k c_k^{(i)} r_k \right) \frac{\partial}{\partial x_\alpha} \left(\sum_l c_l^{(j)} r_l \right) \\
&= \sum_{\alpha=1}^{N} \frac{1}{m_\alpha} \left(\sum_k c_k^{(i)} \frac{\partial}{\partial x_\alpha} r_k \right) \left(\sum_l c_l^{(j)} \frac{\partial}{\partial x_\alpha} r_l \right) \\
&= \sum_k \sum_l c_k^{(i)} \, c_l^{(j)} \left(\sum_{\alpha=1}^{N} \frac{1}{m_\alpha} \frac{\partial}{\partial x_\alpha} r_k \frac{\partial}{\partial x_\alpha} r_l \right) \\
&= \sum_k \sum_l \left(c_k^{(i)} \, c_l^{(j)} \cdot G_{kl}^{(r)} \right)
\end{aligned}
$$

GF Analysis

one fun thing to note is that in our usual construction, we find normal modes and frequencies by finding the eigenvalue and eigenvectors of GF, where F is the Hessian. Since both F and G and real-symmetric, this is equivalent to finding the *generalized* eigenvectors of F and G^{-1}, i.e. solving

$$\mathrm{F}\, v = \lambda \mathrm{G}^{-1}\, v$$

which can be done nicely by `scipy.linalg.eigh`

Note that this means that

$$\mathrm{G}\mathrm{V}^{-1}\, \mathrm{F}\mathrm{V} = \Lambda$$

where V are the normal modes

Derivatives

Since we use it so often, we let the mass-weighted Jacobian be

$$\mathrm{J} = \nabla_{\mathrm{Y}} \mathrm{R}$$

With this, the G matrix is given

$$\mathrm{G} = \mathrm{J}^{\mathrm{T}}\, \mathrm{J}$$

Next let's say we want the derivatives of the G matrix with respect to a new set of coordinates, $\{Q_n\}$.

Going back to prior notes on tensor derivates, we had

$$\nabla_{\mathrm{Q}} \mathrm{V} = \nabla_{\mathrm{Q}\mathrm{x}}\, \nabla_{\mathrm{x}} \mathrm{V}$$

where V was our potential function in Cartesian coordinates, Q was our internal coordinate set, and $\odot$ represents a generalized matrix multiplication to tensors (implemented in `numpy` via

`tensordot`).

Correspondingly since J is defined in our R coordinates, this means that

$$\nabla_{\mathrm{Q}}\mathrm{J}=\nabla_{\mathrm{Q}}\mathrm{R}\,\nabla_{\mathrm{R}}\mathrm{J}$$

the $\nabla_{\mathrm{Q}}\mathrm{R}$ term is the Jacobian for going from the Q coordinate set to the R coordinate set. In the special case the Q_i are defined as linear combinations of the R coordinates defined by some transformation matrix L, this reduces to being simply

$$\nabla_{\mathrm{Q}}\mathrm{R} = \mathrm{L}^{-1}$$

Next we have the $\nabla_{\mathrm{R}}\mathrm{J}$ term, a three dimensional tensor whose elements look like

$$(\nabla_{\mathrm{R}}\mathrm{J})_{ijk}=\frac{\partial}{\partial r_i}\left(\frac{\partial r_j}{\partial x_k}\right)=\frac{\partial {r_j}^2}{\partial r_i\,\partial x_k}=\frac{\partial}{\partial x_k}\left(\frac{\partial r_j}{\partial r_i}\right)=\frac{\partial r_j}{\partial x_k}\,\delta_{ij}$$

which means this is just a nice block-diagonal tensor. This fact can be convenient when implementing this.

G-matrix Derivatives

First

We'll try to rearrange

$$\begin{aligned}\nabla_{\mathrm{Q}}(J^{\mathrm{T}}\,J)&=\left(\nabla_{\mathrm{Q}}J^{\mathrm{T}}\right)\mathrm{J}+\left(J^{\mathrm{T}}\left(\nabla_{\mathrm{Q}}\mathrm{J}\right)^{2:1}\right)^{2:1}\\&=\nabla_{\mathrm{Q}}\mathrm{Y}\,\nabla_{\mathrm{Y}}\mathrm{J}^{\mathrm{T}}\,\mathrm{J}+\nabla_{\mathrm{Q}}\mathrm{Y}\,\langle 3\rangle\,\mathrm{J}^{\mathrm{T}}\,\nabla_{\mathrm{Y}}\mathrm{J}\end{aligned}$$

and just expanding to find the most stable way to make these we expand that first term

$$\begin{aligned}\nabla_{\mathrm{Y}}\mathrm{J}&=\nabla_{\mathrm{Y}}(\nabla_{\mathrm{Y}}\mathrm{Q})\\&=\nabla_{\mathrm{Y}^2}\mathrm{Q}\\&=\nabla_{Y}(\nabla_{Y}R\,\nabla_{R}Q)\\&=\nabla_{Y^2}R\,\nabla_{R}Q+\nabla_{Y}R\,\nabla_{Y}R\,\nabla_{R^2}Q\\&=\nabla_{Y^2}R\,\nabla_{R}Q\\\nabla_{R}Q&=\nabla_{R}Y\,\nabla_{Y}Q\\\nabla_{\mathrm{Q}}\mathrm{Y}\,\nabla_{\mathrm{Y}}\mathrm{J}^{\mathrm{T}}\,\mathrm{J}&=\nabla_{\mathrm{Q}}\mathrm{Y}\,\langle 2\rangle\,(\nabla_{\mathrm{Y}^2}\mathrm{R}\,\nabla_{R}\mathrm{Y}\,\nabla_{\mathrm{Y}}Q)\,\nabla_{\mathrm{Y}}Q\end{aligned}$$

and since we never make use of something that looks like $\nabla_{\mathrm{Y}}\mathrm{Q}\,\nabla_{\mathrm{Q}}\mathrm{Y}$ we should be fine. For the next term we have

$$\nabla_{\mathrm{Q}}\mathrm{Y}\,\langle 3\rangle\,\mathrm{J}^{\mathrm{T}}\,\nabla_{\mathrm{Y}}\mathrm{J}=\nabla_{\mathrm{Q}}\mathrm{Y}\,\langle 3\rangle\,\nabla_{\mathrm{Y}}\mathrm{Q}\,(\nabla_{\mathrm{Y}^2}\mathrm{R}\,\nabla_{R}\mathrm{Y}\,\nabla_{\mathrm{Y}}\mathrm{Q})$$

and again this also should be fine

Second

Then for the second derivatives

$$\begin{aligned}
\nabla_{\mathrm{Q}}\left(\nabla_{\mathrm{Q}} Y\, \nabla_{\mathrm{Y}} \mathrm{J}^{\mathrm{T}}\, J + \nabla_{\mathrm{Q}} \mathrm{Y}\, \langle 3\rangle\, \mathrm{J}^{\mathrm{T}}\, \nabla_{\mathrm{Y}} J\right) &= \nabla_{\mathrm{Q}}\left(\nabla_{\mathrm{Q}} Y\, \nabla_{\mathrm{Y}} J^{\mathrm{T}}\, J\right) + \nabla_{\mathrm{Q}}\left(\nabla_{\mathrm{Q}} \mathrm{Y}\, \langle 3\rangle\, \mathrm{J}^{\mathrm{T}}\, \nabla_{\mathrm{Y}} J\right) \\
&= \nabla_{\mathrm{Q}^2} Y\, \nabla_{\mathrm{Y}} J^{\mathrm{T}}\, J + \nabla_{\mathrm{Q}} Y\, \langle 2\rangle\, \nabla_{\mathrm{Q}} Y\, \nabla_{\mathrm{Y}^2} J^{\mathrm{T}}\, J + \nabla_{\mathrm{Q}} Y\, \langle 2\rangle\, \nabla_{\mathrm{Q}} Y\, \langle 3\rangle\, \nabla_{\mathrm{Y}} J^{\mathrm{T}}\, \nabla_{\mathrm{Y}} \\
&+ \nabla_{\mathrm{Q}^2} \mathrm{Y}\, \langle 3\rangle\, \mathrm{J}^{\mathrm{T}}\, \nabla_{\mathrm{Y}} J + \nabla_{\mathrm{Q}} \mathrm{Y}\, \langle 2\rangle\, \nabla_{\mathrm{Q}} \mathrm{Y}\, \langle 3\rangle\, \nabla_{\mathrm{Y}} \mathrm{J}^{\mathrm{T}}\, \nabla_{\mathrm{Y}} J + \nabla_{\mathrm{Q}} \mathrm{Y}\, \langle 3\rangle\, \nabla_{\mathrm{Q}} \mathrm{Y}\, \langle 3\rangle\, \mathrm{J}^{\mathrm{T}}\, \nabla
\end{aligned}$$

most things are straightforward, but worth noting that

$$\begin{aligned}
\nabla_{\mathrm{Q}}\, Y &= \nabla_Q R\, \nabla_R\, Y \\
\nabla_{Q^2}\, Y &= \nabla_{Q^2} R\, \nabla_R\, Y + \nabla_Q R\, \langle 2\rangle\, \nabla_Q R\, \nabla_{R^2}\, Y \\
&= \nabla_Q R\, \langle 2\rangle\, \nabla_Q R\, \nabla_{R^2}\, Y \\
\nabla_{\mathrm{Y}^2} J &= \nabla_Y (\nabla_Y J) \\
&= \nabla_Y (\nabla_{Y^2} R\, \nabla_R Q) \\
&= \nabla_{Y^3} R\, \nabla_R Q + \nabla_Y R\, \nabla_{R^2} Q \\
&= \nabla_{Y^3} R\, \nabla_R Q
\end{aligned}$$

i.e.

$$\frac{\partial y_j}{\partial Q_i} = \sum_\alpha \frac{\partial R_\alpha}{\partial Q_i}\, \frac{\partial R_\alpha}{\partial Q_i}$$

G-Matrix with coordinate transformations

Assuming we've calculate the G matrix for some set of coordinates, R, as

$$G_R = J^T\, J$$

if we want to translate this to a different coordinate system, Q, we need to recognize that

$$\nabla_X Q = \nabla_X R\, \nabla_R Q$$

giving us

$$\begin{aligned}
G_Q &= (\nabla_X Q)^T \nabla_X Q = (\nabla_X R\, \nabla_R Q)^T \nabla_X R\, \nabla_R Q \\
&= \nabla_R Q^T (\nabla_X R)^T \nabla_X R\, \nabla_R Q \\
&= \nabla_R Q^T\, J^T J\, \nabla_R Q \\
&= \nabla_R Q^T\, G_R\, \nabla_R Q
\end{aligned}$$

Alternately if we're calculating G^{-1} by default we start with

$$G^{-1} = J^{-1} \left(J^{-1}\right)^T$$

and we can work from there in a similar way

Dipole Derivatives

We need to transform from Cartesian coords to internals again...but this time we've got different mixed derivatives to deal with.

The first derivatives are straightforward (being wholly in Cartesian coordinates)

$$\nabla_Q \mu_\alpha = \nabla_Q X \, \nabla_X \mu_\alpha$$

The second derivatives are not bad, even though we start with $\nabla_q \nabla_X \mu_\alpha$

$$\nabla_Q \nabla_X \mu_\alpha = \nabla_Q q \, \nabla_q \nabla_X \mu_\alpha$$
$$\nabla_{Q^2} \mu_\alpha = \nabla_Q X \, \langle 2 \rangle \, \nabla_Q \nabla_X \mu_\alpha$$

note that this can be easily proved by the fact that $\nabla_Q \nabla_X \mu_\alpha = \left(\nabla_X \nabla_Q \mu_\alpha\right)^{\mathrm{T}}$ *by Clairaut's theorem or whatever*

The third derivatives require a bit of care, as we start with $\nabla_{q^2} \nabla_X \mu_\alpha$ and so we have

$$\begin{aligned} \nabla_{Q^2} \nabla_X \mu_\alpha &= \nabla_Q \left(\nabla_Q q \, \nabla_q \nabla_X \mu_\alpha \right) \\ &= \nabla_{Q^2} q \, \nabla_q \nabla_X \mu_\alpha + \nabla_Q q \, \langle 2 \rangle \, \nabla_Q q \, \nabla_{q^2} \nabla_X \mu_\alpha \end{aligned}$$

and then finally

$$\nabla_{Q^3} \mu_\alpha = \nabla_Q X \, \langle 3 \rangle \, \nabla_{Q^2} \nabla_X \mu_\alpha$$

Coriolis Zeta Constants

To be able to compare to pure Cartesian coordinate calculations, it is useful to be able to add the effects of Coriolis coupling to our calculations, even though we'd prefer to work purely in internal coordinates. To make this work, we note that the according to Barone and friends, the Coriolis term comes in at second order

$$H^{(2)} = \frac{1}{24} \, \nabla_{Q^4} V \; \mathrm{QQQQ} + \sum_{\tau \in \{x \; y, \; z\}} B_e^\tau \, \zeta_{ij}^\tau \, \zeta_{kl}^\tau \left(\frac{\omega_j \, \omega_l}{\omega_i \, \omega_k} \right) q_i \, p_j \, q_k \, p_l$$

$$\hat{H}^{(2)} = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} \sum_{l=1}^{N} \frac{1}{24} k_{ijkl} q_i q_j q_k q_l$$

$$+ \sum_{\tau=x,y,z} B_e^\tau \zeta_{ij}^\tau \zeta_{kl}^\tau \left(\frac{\omega_j \omega_l}{\omega_i \omega_k} \right) q_i \hat{p}_j q_k \hat{p}_l ,$$

and then the question is simply how to evaluate the ζ^{τ}_{ij} terms. Those are defined as

$$\zeta^{z}_{ij}=\sum_{n}\frac{\partial x_n}{\partial Q_i}\,\frac{\partial y_n}{\partial Q_j}-\frac{\partial x_n}{\partial Q_j}\,\frac{\partial y_n}{\partial Q_i}$$

which can be rewritten as

$$\zeta^{z}_{ij}=\sum_{n}\left(\frac{\partial x_n}{\partial Q_i}+\frac{\partial y_n}{\partial Q_i}\right)\left(-\frac{\partial x_n}{\partial Q_j}+\frac{\partial y_n}{\partial Q_j}\right)-\left(\frac{\partial y_n}{\partial Q_i}\,\frac{\partial y_n}{\partial Q_j}-\frac{\partial x_n}{\partial Q_i}\,\frac{\partial x_n}{\partial Q_j}\right)$$

Why is this nice? Well it makes the operator form of this clearer. Letting $\tau\in\{a,b,c\}$ and defining two transpositions of the Jacobian like

$$J_{ni\tau}=\frac{\partial \tau_n}{\partial_{Q_i}}$$

$$J^{\mathrm{T}}{}_{n\tau i}=\frac{\partial \tau_n}{\partial_{Q_i}}$$

we can write

$$\zeta^{\tau}_{ij}=\sum_{n}J_{ni}\,\varepsilon^{(\alpha\beta)}\,J^{\mathrm{T}}{}_{jn}$$

and $\varepsilon^{(\alpha\beta)}$ is the 2D Levi-Civita tensor (antisymmetric tensor) defined for $\alpha,\beta\in\{\mathrm{a,\ b,\ c}\}\backslash\{\tau\}$. And so then we can build the entirety of ζ in one go by

$$\zeta=\sum_{n}J_n\,\varepsilon\,J^{\mathrm{T}}{}_n$$

where ε is now the proper 3D Levi-Civita tensor.

Unit-wise, things are kinda annoying. We note that we expect J to be the matrix of derivatives of the mass-weighted Cartesian coordinates with respect to the dimensionless normal modes. If this is the case, we know that

$$J^{-1\,T}\,J^{-1}=G$$

and so all of these products should have units of energy. If everything is internally consistent and in atomic units, specifically this energy unit is E_h. That means we expect ζ to have units of $1/E_h$...but that doesn't work with the expression

$$\sum_{\tau}B^{\tau}_{e}\,\zeta^{\tau}_{ij}\,\zeta^{\tau}_{kl}\left(\frac{\omega_j\,\omega_l}{\omega_i\,\omega_k}\right)p_i\,q_j\,p_l\,q_k$$

which is supposed to be the Coriolis correction in $H^{(2)}$...because B^{τ}_{e} should *also* have units of energy and this would end up giving something with units of $1/E_h$.

Potential-Like Terms

Come out of the how the inertia tensor transforms in internal coordinates. Add a brief overview.

The Watson Term

Add some notes about how you need the U term to get a ZPE that agrees with Gaussian in Cartesian displacements (linearized internals).

The V' Term

Old Origin

Origins

From Meyer we have

$$T_v = -\frac{\hbar^2}{2}\sum_{i,j}\left(\frac{\partial}{\partial r_i}+\frac{1}{4d}\frac{\partial d}{\partial r_i}\right)G_{ij}\left(\frac{\partial}{\partial r_j}-\frac{1}{4d}\frac{\partial d}{\partial r_j}\right)$$

where

$$d = \begin{vmatrix} M & & \\ & I_0 & \Gamma \\ & \Gamma & G^{-1} \end{vmatrix}$$

which in the PAS reduces to

$$d = \begin{vmatrix} M & & \\ & I_0 & \\ & & G^{-1} \end{vmatrix} = |I_0|/|G|$$

then we can foil out terms to get

$$\left(\frac{\partial}{\partial r_i}+\frac{1}{4d}\frac{\partial d}{\partial r_i}\right)G_{ij}\left(\frac{\partial}{\partial r_j}-\frac{1}{4d}\frac{\partial d}{\partial r_j}\right) =$$

$$\left(\frac{\partial}{\partial r_i}G_{ij}\frac{\partial d}{\partial r_j}\right)+\frac{1}{4d}\left(\frac{\partial d}{\partial r_i}G_{ij}\frac{\partial}{\partial r_j}-\frac{\partial}{\partial r_i}G_{ij}\frac{\partial d}{\partial r_j}-\frac{1}{4d}G_{ij}\frac{\partial d}{\partial r_i}\frac{\partial d}{\partial r_j}\right)$$

applying commutator rules we have

$$\frac{\partial}{\partial r_i}G_{ij} = G_{ij}\frac{\partial}{\partial r_i}+\frac{\partial G_{ij}}{\partial r_i}$$

$$\frac{\partial}{\partial r_i}\frac{\partial d}{\partial r_j}=\frac{\partial d}{\partial r_j}\frac{\partial}{\partial r_i}+\frac{\partial^2 d}{\partial r_i\partial r_j}$$

$$\frac{\partial d}{\partial r_i}G_{ij}\frac{\partial}{\partial r_j}-\frac{\partial}{\partial r_i}G_{ij}\frac{\partial d}{\partial r_j}=\frac{\partial d}{\partial r_i}G_{ij}\frac{\partial}{\partial r_j}-G_{ij}\frac{\partial}{\partial r_i}\frac{\partial d}{\partial r_j}-\frac{\partial d}{\partial r_j}\frac{\partial G_{ij}}{\partial r_i}$$

$$=\frac{\partial d}{\partial r_i}G_{ij}\frac{\partial}{\partial r_j}-\frac{\partial d}{\partial r_j}G_{ij}\frac{\partial}{\partial r_i}-\frac{\partial d}{\partial r_j}\frac{\partial G_{ij}}{\partial r_i}-G_{ij}\frac{\partial^2 d}{\partial r_i\partial r_j}$$

and we add over all pairs of i and j so

$$\sum_{i,j}\frac{\partial d}{\partial r_i}G_{ij}\frac{\partial}{\partial r_j}-\frac{\partial d}{\partial r_j}G_{ij}\frac{\partial}{\partial r_i}=0$$

giving us

$$-G_{ij}\frac{\partial d}{\partial r_j}+\frac{\partial d}{\partial r_j}G_{ij}\frac{\partial}{\partial r_i}-\frac{\partial}{\partial r_i}G_{ij}\frac{\partial d}{\partial r_i}=\frac{\partial d}{\partial r_i}G_{ij}\frac{\partial}{\partial r_j}-\frac{\partial d}{\partial r_j}G_{ij}\frac{\partial}{\partial r_i}-\frac{\partial d}{\partial r_j}\frac{\partial G_{ij}}{\partial r_i}+\frac{\partial d}{\partial r_j}G_{ij}\frac{\partial}{\partial r_i}-\frac{\partial d}{\partial r_i}G_{ij}$$

$$=-\frac{\partial d}{\partial r_j}\frac{\partial G_{ij}}{\partial r_i}-\frac{\partial d}{\partial r_i}\frac{\partial G_{ij}}{\partial r_j}$$

so

$$\sum_{i,j}\left(\frac{\partial}{\partial r_i}+\frac{1}{4d}\frac{\partial d}{\partial r_i}\right)G_{ij}\left(\frac{\partial}{\partial r_j}-\frac{1}{4d}\frac{\partial d}{\partial r_j}\right)=$$

$$\sum_{i,j}\left(\frac{\partial}{\partial r_i}G_{ij}\frac{\partial d}{\partial r_j}\right)-\frac{1}{4d}\sum_{i,j}\frac{\partial d}{\partial r_j}\frac{\partial G_{ij}}{\partial r_i}+G_{ij}\left(\frac{\partial^2 d}{\partial r_i\partial r_j}+\frac{1}{4d}\frac{\partial d}{\partial r_i}\frac{\partial d}{\partial r_j}\right)$$

this gives us then

$$T_v=-\frac{\hbar^2}{2}\sum_{i,j}\left(\frac{\partial}{\partial r_i}+\frac{1}{4d}\frac{\partial d}{\partial r_i}\right)G_{ij}\left(\frac{\partial}{\partial r_j}-\frac{1}{4d}\frac{\partial d}{\partial r_j}\right)$$

$$=-\frac{\hbar^2}{2}\sum_{i,j}\left(\frac{\partial}{\partial r_i}G_{ij}\frac{\partial d}{\partial r_j}\right)+\frac{\hbar^2}{8}\sum_{i,j}\frac{1}{d}\left(\frac{\partial d}{\partial r_j}\frac{\partial G_{ij}}{\partial r_i}+G_{ij}\left(\frac{\partial^2 d}{\partial r_i\partial r_j}+\frac{1}{4d}\frac{\partial d}{\partial r_i}\frac{\partial d}{\partial r_j}\right)\right)$$

which gives

$$V'=\frac{\hbar^2}{8}\sum_{i,j}\frac{1}{d}\left(\frac{\partial d}{\partial r_j}\frac{\partial G_{ij}}{\partial r_i}+G_{ij}\left(\frac{\partial^2 d}{\partial r_i\partial r_j}+\frac{1}{4d}\frac{\partial d}{\partial r_i}\frac{\partial d}{\partial r_j}\right)\right)$$

Implementation

From Pickett, we have

$$V' = \frac{\hbar^2}{8} \sum_{ij} \frac{\partial G_{ij}}{\partial q_i} \frac{\partial \gamma}{\partial q_j} + G_{ij} \left(\frac{\partial^2 \gamma}{\partial q_i \, q_j} + \frac{1}{4} \frac{\partial \gamma}{\partial q_i} \frac{\partial \gamma}{\partial q_j} \right)$$

where, calling the mass-weighted coordinates Y

$$G = \nabla_Q Y \left(\nabla_Q Y \right)^T$$

$$\gamma = \ln \frac{|I_0|}{|G|}$$

where I^0 is the moment of inertia tensor. This γ term is the natural log of the determinant of the inverse vibration-rotation kinetic energy tensor. Ignoring the difficulty of getting the γ derivatives for the moment (we'll come back to it), we'll note that we can partially factorize this as

$$\frac{\partial G_{ij}}{\partial q_i} \frac{\partial \gamma}{\partial q_j} + G_{ij} \left(\frac{\partial^2 \gamma}{\partial q_i \, q_j} + \frac{1}{4} \frac{\partial \gamma}{\partial q_i} \frac{\partial \gamma}{\partial q_j} \right) = \left(\frac{\partial G_{ij}}{\partial q_i} + \frac{1}{4} G_{ij} \frac{\partial \gamma}{\partial q_i} \right) \frac{\partial \gamma}{\partial q_j} + G_{ij} \frac{\partial^2 \gamma}{\partial q_i \, q_j}$$

which is different from the product-rule type factorization that we could also try

$$\frac{\partial G_{ij}}{\partial q_i} \frac{\partial \gamma}{\partial q_j} + G_{ij} \left(\frac{\partial^2 \gamma}{\partial q_i \, q_j} + \frac{1}{4} \frac{\partial \gamma}{\partial q_i} \frac{\partial \gamma}{\partial q_j} \right) = \frac{\partial}{\partial q_i} \left(G_{ij} \frac{\partial \gamma}{\partial q_j} \right) = \frac{\partial G_{ij}}{\partial q_i} \frac{\partial \gamma}{\partial q_j} + G_{ij} \frac{1}{4} \frac{\partial \gamma}{\partial q_i} \frac{\partial \gamma}{\partial q_j}$$

on the other hand, that first form is potentially more useful, though, because assuming we have the tensor $\nabla_Q G$ (which we need for the perturbation theory anyway) we can write the total sum as

$$\sum_{ij} \frac{\partial G_{ij}}{\partial q_i} \frac{\partial \gamma}{\partial q_j} + G_{ij} \left(\frac{\partial^2 \gamma}{\partial q_i \, q_j} + \frac{1}{4} \frac{\partial \gamma}{\partial q_i} \frac{\partial \gamma}{\partial q_j} \right) = \sum_i \left(\nabla_Q G \right)_{ii} \nabla_Q \gamma + G \odot_2 \nabla_{Q^2} \gamma + \frac{1}{4} \left(G \nabla_Q \gamma \right)^{\mathrm{T}} \nabla_Q \gamma$$

and now all that's left is figuring out how to evaluate

$$\left(\nabla_Q \gamma \right)_i = \frac{\partial}{\partial q_i} \ln \frac{|I_0|}{|G|}$$

$$\left(\nabla_{Q^2} \gamma \right)_{ij} = \frac{\partial^2}{\partial q_i \, \partial q_j} \ln \frac{|I_0|}{|G|}$$

and now the Jacobi formula will be super useful.

$$\begin{aligned} \frac{\partial}{\partial q_i} \ln \frac{|I_0|}{|G|} &= \frac{\partial}{\partial q_i} \ln |I_0| - \frac{\partial}{\partial q_i} \ln |G| \\ &= \frac{1}{|I_0|} \frac{\partial}{\partial q_i} |I_0| - \frac{1}{|G|} \frac{\partial}{\partial q_i} |G| \end{aligned}$$

$$\frac{\partial^2}{\partial q_j \, \partial q_i} \ln \frac{|I_0|}{|G|} = \frac{\partial}{\partial q_j} \left(\frac{1}{|I_0|} \frac{\partial}{\partial q_i} |I_0| - \frac{1}{|G|} \frac{\partial}{\partial q_i} |G| \right)$$

$$=-\frac{1}{|I_0|^2}\,\frac{\partial|I_0|}{\partial q_i}\,\frac{\partial|I_0|}{\partial q_j}+\frac{1}{|I_0|}\,\frac{\partial^2|I_0|}{\partial q_j\,\partial q_i}-\left(-\frac{1}{|G|^2}\,\frac{\partial|G|}{\partial q_i}\,\frac{\partial|G|}{\partial q_j}+\frac{1}{|G|}\,\frac{\partial^2|G|}{\partial q_j\,\partial q_i}\right)$$

We'll focus on the inertia tensor for now, but note that everything will work the same for G. For the first derivatives we have

$$\frac{\partial}{\partial q_i}\left|I^0\right|=\mathrm{Tr}\left(\mathrm{Adj}(I_0)\,\frac{\partial I_0}{\partial q_i}\right)$$

where Adj is the *adjugate matrix*. In general, it's the transpose of the *cofactor matrix*, but since I^0 is invertible we have

$$\mathrm{Adj}(I_0)=I_0^{-1}\,|I_0|$$
$$\nabla_Q\mathrm{Adj}(I_0)=|I_0|\,\nabla_Q I_0^{-1}\,+\,\nabla_Q\,|I_0|\otimes I_0^{-1}$$

and looking towards the second derivatives we note that

$$\frac{\partial}{\partial q_j}\,\mathrm{Tr}(A)=\frac{\partial}{\partial q_j}\sum_k A_{kk}$$
$$=\sum_k\frac{\partial}{\partial q_j}\,A_{kk}$$

where just for generality it's worth noting that

$$\nabla_Q\mathrm{Tr}(A)\,=\,\mathrm{Tr}\left(\nabla_Q A,\,2{,}3\right)$$

where the 2,3 means we sum over terms diagonal in axes 2 and 3 or equivalently we can write

$$\nabla_Q\mathrm{Tr}(A)\,=\,\mathrm{Tr}\left(\left(\nabla_Q A\right)^{1:3}\right)$$

giving us

$$\nabla_{Q^2}\mathrm{Tr}(A)\,=\,\mathrm{Tr}\left(\left(\nabla_Q\left(\nabla_Q A\right)^{1:3}\right)^{1:3}\right)$$
$$=\,\mathrm{Tr}\left(\nabla_{Q^2}A^{2:4^{1:3}}\right)$$

and so we get

$$\frac{\partial^2}{\partial q_j\,\partial q_i}|I_0|=\sum_k\frac{\partial}{\partial q_j}\left(\mathrm{Adj}(I_0)\,\frac{\partial I_0}{\partial q_i}\right)_{kk}$$
$$=\sum_k\frac{\partial}{\partial q_j}\left(\mathrm{Adj}(I_0)_k\,\frac{\partial(I_0)_{:k}}{\partial q_i}\right)$$
$$=\sum_k\frac{\partial\mathrm{Adj}(I_0)_k}{\partial q_j}\,\frac{\partial(I_0)_{:k}}{\partial q_i}+\mathrm{Adj}\left(I^0\right)_k\,\frac{\partial^2(I_0)_{:k}}{\partial q_i\,\partial q_j}$$
$$=\left(\nabla_Q I_0\right)_i\odot_2\left(\nabla_Q\mathrm{Adj}(I_0)\right)_j+\mathrm{Adj}\left(I^0\right)\odot_2\left(\nabla_{Q^2}I_0\right)_{ij}$$

then

$$\begin{aligned}\frac{\partial \mathrm{Adj}(I_0)_k}{\partial q_j} &= \frac{\partial}{\partial q_j}\left(I_0^{-1}\right)_k |I_0| \\ &= |I_0| \frac{\partial}{\partial q_j}\left(I_0^{-1}\right)_k + \left(I_0^{-1}\right)_k \frac{\partial}{\partial q_j}|I_0| \\ &= |I_0| \left(\nabla_Q I_0^{-1}\right)_{jk} + \left(\nabla_Q |I_0|\right)_j \left(I_0^{-1}\right)_k \\ \nabla_Q \mathrm{Adj}(I_0) &= |I_0| \nabla_Q I_0^{-1} + \left(\nabla_Q |I_0|\right) I_0^{-1}\end{aligned}$$

(where we made use of the fact that since I_0 is symmetric its inverse is too). Putting these together because of masochism we have

$$\frac{\partial^2}{\partial q_j \partial q_i}|I_0| = \left(\nabla_Q I_0\right)_i \odot_2 \left(|I_0| \nabla_Q I_0^{-1}\right)_j + \left(\nabla_Q I_0\right)_i \odot_2 \left(\left(\nabla_Q |I_0|\right) I_0^{-1}\right)_j + |I_0| I_0^{-1} \odot_2 \left(\nabla_{Q^2} I_0\right)_{ij}$$

Finally when worrying about how to do the inertia tensor derivatives we have

$$\begin{aligned}\frac{\partial I_0}{\partial q_i} &= \left(\nabla_Q Y\right)_i \nabla_Y I_0 \\ \frac{\partial^2 I_0}{\partial q_i \partial q_j} &= \nabla_Q \left(\left(\nabla_Q Y\right)_j \nabla_Y I_0\right)_i \\ &= \nabla_Q Y \left(\left(\nabla_Q Y\right)_j \nabla_Y I_0\right) \\ &= \left(\nabla_{Q^2} Y\right)_{ij} \nabla_Y I_0 + \left(\nabla_Q Y\right)_j \left(\nabla_Q Y\right)_i \nabla_{Y^2} I_0\end{aligned}$$

then getting the Cartesian derivatives of the inertia tensor, for every mass-weighted atomic position vector we get a derivative tensor like

$$\begin{aligned}\frac{\partial}{\partial y_n} I_0 &= \frac{\partial}{\partial y_n} \sum_m \mathbb{I}_3 (y_m \cdot y_m) - y_m \otimes y_m \\ &= \frac{\partial}{\partial y_n} (\mathbb{I}_3\ y_n \cdot y_n - y_n \otimes y_n) \\ &= 2\ y_n (\mathbb{I}_3 \otimes \mathbb{I}_3) - (y_n \otimes \mathbb{I}_3 + \mathbb{I}_3 \otimes y_n)\end{aligned}$$

where we've got one non-standard notation. The term $y_n \otimes \mathbb{I}_3$ should be read to mean the vector $(y_n \otimes (\mathbb{I}_3)_i)_i$. I chose to write it this way because this is easy to implement in modern numerical programming languages. Your milage may vary on this, though. Worth also noting that if your language supports high-dimensional transpositions, we have

$$y_n \otimes \mathbb{I}_3 = \mathbb{I}_3 \otimes y_n^{(1,\ 3,\ 2)}$$

where the superscripted (1, 3, 2) indicates that we transposed the final two axes of the tensor.

With this out of the way, the second derivative tensor is actually pretty straightforward and actually has no dependence on the coordinates themselves (this because the products & moments

of inertia are quadratic terms)

$$\frac{\partial}{\partial y_n \partial y_m} I_0 = \left(2\,(\mathbb{I}_3 \otimes \mathbb{I}_3) - \left(\,(\mathbb{I}_3 \otimes (\mathbb{I}_3)_i)_i + \,(\mathbb{I}_3 \otimes (\mathbb{I}_3)_i)_i{}^{(1,2,4,3)}\right)\right) \delta_{nm}$$

where again, totally weird notation, but $(\mathbb{I}_3 \otimes (\mathbb{I}_3)_i)_i$ is the same kind of vector of outer products as before and the (1,2,4,3) exponent just means transpose the final two axes in the tensor. Not gonna deny that it's not intuitive notation, but it's easy to implement and that's what really matters. Importantly, this also means that the third derivative of I_0 is identically zero

The final component that we need to think about how to handle is

$$\frac{\partial}{\partial q_j}\left(\left(I_0{}^{-1}\right)\right)_j = \left(\nabla_Q \left(I_0{}^{-1}\right)\right)_{jj}$$

Happily, using the formula for the derivative of a matrix inverse from The Matrix Cookbook we can actually formulate this without too much trouble. They give use

$$\frac{\partial Y^{-1}}{\partial x} = -\left(Y^{-1}\right) \frac{\partial Y}{\partial x}\; Y^{-1}$$

or explicitly

$$\frac{\partial I_0{}^{-1}}{\partial y_n} = -\left(I_0{}^{-1}\right) \frac{\partial I_0}{\partial y_n}\; I_0{}^{-1}$$

and so generalizing to the gradient with respect to the normal modes we have

$$\nabla_Q \left(I_0{}^{-1}\right) = -\left(I_0{}^{-1}\right) \langle 2 \rangle\, \nabla_Q I_0\, I_0{}^{-1}$$

which justifies all of the effort that we put into writing down those expressions for $\frac{\partial}{\partial y_n} I_0$ and friends

Tests

A Simpler Take on V'

Towards Analytic Expressions

State Indexing

We need to be able to do efficient indexing of states, which can be expressed as vectors of quanta of excitation in our different modes. The basic tack we'll take is to order by number of quanta of excitation, then lexicographically based on the partition of the total number of quanta, and

finally lexicographically based for the permutation of that partition.

This has many nice features, for one, the ground state is always the first state, the states are ordered in terms of quanta of excitation, and then states are ordered by how few modes they touch. On the other hand, these numbers will still grow quite quickly.

For a given number of total quanta of excitation, there will be some number of integer partitionings. For each of those partitionings, the number of distinct permutations is given by

$$\text{perms}(p) = \frac{D!}{(D\text{-}|p|)! \prod_i \#p_i}$$

where $\#p_i$ gives the number of times p_i appears in the partition.

Permutation Indexing

Partition Indexing

We can get nice recurrences to give us ways to count the number of partitions of an integer with given properties. We'll let it be stated without proof that

$$p(n,\ M,\ L) = p(n,\ M,\ L{-}1) + p(n{-}L,\ M{-}1,\ L)$$

is the number of partitions of n that have at most L elements where each element is $\leq M$. We will also find it useful to note that we can write the number of terms with *exactly* L elements and max-size M as

$$\begin{aligned} q(n,\ M,\ L) &= p(n,\ M,\ L) - p(n,\ M,\ L{-}1) \\ &= p(n,\ M,\ L{-}1) + p(n{-}L,\ M{-}1,\ L) - p(n,\ M,\ L{-}1) \\ &= p(n{-}L,\ M{-}1,\ L) \end{aligned}$$

and then furthermore we'll also often want the number of terms with *exactly* L elements with max-size *greater* than M and *less than* m, which is given by

$$\begin{aligned} g(n,\ m,\ M,\ L) &= q(n,\ m,\ L) - q(n,\ M,\ L) \\ &= p(n{-}L,\ m{-}1,\ L) - p(n{-}L,\ M{-}1,\ L) \end{aligned}$$

Next, if we order our partitions in reverse lexicographic order, prioritizing shorter partitions, we know that the number of partitions *before* a given partition σ of n with length L and max-size M is given by

$$\text{idx}(\sigma,\ n,\ M,\ L) = p(n,\ n,\ L{-}1) + \text{idxL}(\sigma,\ n,\ M,\ L)$$

WARNING: There are some fucked up indices here

where idxL counts the number of partitions with *exactly* length L that precede σ and is recur-

sively defined as

$$\begin{aligned}\text{idxL}(\sigma,\, n,\, M,\, L) &= g(n,\, \sigma[0],\, M,\, L) + \text{idxL}(\sigma[1{:}],\, n - M,\, \sigma[1],\, L{-}1) \\ &= g(n,\, n,\, M,\, L) + \sum_{i=0}^{L} g\left(n{-}\sum_{j=0}^{i-1}\sigma[j],\, \sigma[i{-}1],\, \sigma[i],\, L{-}i\right)\end{aligned}$$

we can recognize, though, that $n = \sum_{j=0}^{L-1}\sigma_j$, therefore we can say

$$s_i(\sigma) = n{-}\sum_{j=0}^{i-1}\sigma_j = \sum_{j=i}^{L-1}\sigma_j$$

which gives

$$\begin{aligned}\text{idxL}(\sigma,\, n,\, M,\, L) &= \sum_{i=0}^{L} g(s_i(\sigma),\, \sigma_i,\, L{-}i) \\ &= \sum_{i=0}^{L} p(s_i(\sigma){-}(L{-}i),\, s_i(\sigma) - 1,\, L{-}i) - p(s_i(\sigma){-}(L{-}i),\, \sigma_i - 1,\, L{-}i)\end{aligned}$$

so

$$p(n,\, n,\, L{-}1) + p(n{-}L,\, n{-}1,\, L) - p(n,\, \sigma_0{-}1,\, L) + \sum_{i=1}^{L} p(s_i(\sigma){-}(L{-}i),\, \sigma_{i-1}{-}1,\, L{-}i) - p(s_i(\sigma){-}(L{-}i)$$

$$p(n,\, n,\, L) - p(n,\, \sigma_0{-}1,\, L) + \sum_{i=1}^{L} p(s_i(\sigma){-}(L{-}i),\, \sigma_{i-1}{-}1,\, L{-}i) - p(s_i(\sigma){-}(L{-}i),\, \sigma_i{-}1,\, L{-}i)$$

$$p(n,\, n,\, L) + \sum_{i=1}^{L} p(s_i(\sigma){-}(L{-}i),\, \sigma_{i-1}{-}1,\, L{-}i) - \sum_{i=0}^{L} p(s_i(\sigma){-}(L{-}i),\, \sigma_i{-}1,\, L{-}i)$$

$$p(n,\, n,\, L) - p(s_L(\sigma){-}(L{-}L),\, \sigma_L{-}1,\, L{-}L) + \sum_{i=1}^{L-1} p(s_i(\sigma){-}(L{-}i),\, \sigma_{i-1}{-}1,\, L{-}i) - p(s_{i-1}(\sigma){-}(L{-}(i{-}1)),$$

$$p(n,\, n,\, L) + \sum_{i=1}^{L-1} p(s_i(\sigma){-}(L{-}i),\, \sigma_{i-1}{-}1,\, L{-}i) - p(s_i(\sigma){-}(L{-}i) + \sigma_{i-1}{-}1,\, \sigma_{i-1}{-}1,\, L{-}i{+}1)$$

which can be written cleanly as a direct loop without any need for recursion (or even the partial summations)...although I'm not sure this final expression is any nicer than just using the original g form

---

Young Diagrams and Permutation/Partition Representations

We'll note that quantum numbers for every state for a multidimensional H.O. with N quanta of excitation can be written like

$$\left|n_1,\ n_2,\ ...,\ n_D\right\rangle \sim \left(\pi_D,\ \lambda_D^N\right)$$

where π_D is a permutation of length D and λ_D^N is a partition of the integer N over D elements.

We'll also note that we can map every operator $H^{(i)}$ to a similar representation

$$H^{(i)} \sim \left(\mathcal{P}_D,\ \Lambda_D^M\right)$$

where $H^{(i)}$ is just the direct-sum of terms looking like

$$q{=}\pi_D\left(\prod_{j=1}^{D} q_j^{\left(\lambda_D^M\right)_j}\right)$$

so we have an expansion over partitions and permutations.

When we look at how one of these terms operates on a H.O. function, we get restricted numbers of terms, which will come from our selection rule

$$\begin{aligned} s(q){=}\pi_D\left(\bigotimes_{j=1}^{D} s\left(q_j^{\left(\lambda_D^M\right)_j}\right)\right) \\ {=}\pi_D\left(\bigotimes_{j=1}^{D} (-1,\ 1)^{\left(\lambda_D^M\right)_j}\right) \end{aligned}$$

where that exponentiation just means iterated direct summation

$$\begin{aligned} (-1,\ 1)^1&{=}(-1,\ 1) \\ (-1,\ 1)^2&{=}(-2,\ 0,\ 2) \\ (-1,\ 1)^3&{=}(-3,\ -1,\ 1,\ 3) \\ (-1,\ 1)^4&{=}(-4,\ -2,\ 0,\ 2,\ 4) \\ &... \end{aligned}$$

each of these, clearly, can just be written as

$$(-1,\ 1)^m{=}\{-m\ +\ 2\ i\ :\ i{\in}\mathbb{N}_{2m}\}$$

or can be mapped onto a binary sequence representing the actual way this was calculated

$$(-1,\ 1)^m{=}\left\{\sum_i (-1)^{l_i}\ :\ l{\in}\{\mathrm{bin}(2^n{-}1,\ m)\ :\ n{\in}\mathbb{N}_{2m}\}\ \right\}$$

in any case, this shows we can map the set of selection rules for a given operator which maps to $\left(\pi_D,\ \lambda_D^M\right)$ as

$$s(q){=}\pi_D\left(\bigcup_{j=1}^{D}\left\{\ r_{\xi_j}^{\left(\lambda_D^M\right)_j}\right\}\right)$$

where $r_{\xi_j}^{(\lambda_D^M)_j}$ is just the ξ_j^{th} element of $(-1,\ 1)^{(\lambda_D^M)_j}$ and we're taking the union over every possible combo of indices, i.e.

$$\xi \in \bigotimes_{i=1}^{D} \mathbb{N}_{2(\lambda_D^M)_j} - 1 = \Xi\left(\lambda_D^M\right)$$

this then allows us to map every element of $s(q)$ as

$$\left|\Delta s_1,\ \Delta s_2,\ ...,\ \Delta s_D\right\rangle \sim \left(\pi_D,\ \lambda_D^M,\ \xi\left(\lambda_D^M\right)\right)$$

so then when we do

$$\left|n_1,\ n_2,\ ...,\ n_D\right\rangle + \left|\Delta s_1,\ \Delta s_2,\ ...,\ \Delta s_D\right\rangle = \left|m_1,\ m_2,\ ...,\ m_D\right\rangle$$

we end up mapping to

$$\left|m_1,\ m_2,\ ...,\ m_D\right\rangle \sim \left(\left({\pi^{(1)}}_D,\ {\lambda^{(1)}}_D^N\right),\ \left(\pi_D^{(2)},\ {\lambda^{(2)}}_D^M,\ \xi\left({\lambda^{(2)}}_D^M\right)\right)\right)$$

and then the question is how do we remap this back to a form like

$$\left(\pi_D,\ \lambda_D^L\right)$$

where L is somehow related to M and N and this new π_D is related to π_{1D}, π_{2D} , and ξ...

Starting out with the absolute basics, we know for a partition λ_{2D}^M, the total number of quanta changed is in $[-M,\ M]$, corresponding to $\xi = (0,\ 0,\ 0...\ )$ and $\xi = 2\,\lambda_{2D}^M - 1$ respectively, for the H.O. This means we have $L \in [N-M,\ N+M]$ but of course the question is *which* specifically.

For this, we'll think algorithmically about what's happening. First off, thinking about the sequence of changes we have, excluding the permutation, we have

$$\Delta s_i = -\left(\lambda_D^M\right)_i + 2\ \xi\left(\lambda_D^M\right)_i$$

which means the change in quanta is given by

$$\begin{aligned}\Delta M &= \sum_i \Delta s_i \\ &= \sum_i -\left(\lambda_D^M\right)_i + 2\ \xi\left(\lambda_D^M\right)_i \\ &= -M + 2\sum_i \xi\left(\lambda_D^M\right)_i\end{aligned}$$

and since $\xi\left(\lambda_{2D}^M\right)_i$ is restricted to integer values, the possible changes in quanta are restricted to $[-M,\ -M+2,\ ...,\ M-2,\ M]$

That, at least, is straightforward! Obviously which case depends on which ξ we pull.

More annoying is to figure out which λ_D^L we'll end up with. For that, we note that when we write

$$\left|n_1,\ n_2,\ ...,\ n_D\right\rangle \sim \left({\pi^{(1)}}_D,\ {\lambda^{(1)}}_D^N\right)$$

we are defining this relation by

$$n_i = \left(\lambda^{(1)}{}^N_D\right)_j$$
$$j = \left(\pi^{(1)}{}_D\right)_i$$

similarly, when we write

$$\left|\Delta s_1,\, \Delta s_2,\, ...,\, \Delta s_D\right\rangle \sim \left(\pi^{(2)}_D,\, \lambda^{(2)}{}^M_D,\, \xi\left(\lambda^{(2)}{}^M_D\right)\right)$$

we are saying

$$\Delta s_i = -\left(\lambda^{(2)}{}^M_D\right)_j + 2\, \xi\left(\lambda^{(2)}{}^M_D\right)_j$$
$$j = \left(\pi^{(2)}{}_D\right)_i$$

meaning

$$n_i + \Delta s_i = \left(\lambda^{(1)}{}^N_D\right)_j - \left(\lambda^{(2)}{}^M_D\right)_k + 2\, \xi\left(\lambda^{(2)}{}^M_D\right)_k$$
$$j = \left(\pi^{(1)}{}_D\right)_i$$
$$k = \left(\pi^{(2)}{}_D\right)_i$$

unfortunately, this doesn't seem very promising. However we can imagine some *second* permutation $\pi^{(3)}{}_D$ such that

$$\pi^{(3)}{}_D\left(\pi^{(2)}{}_D\right) = \pi^{(1)}{}_D$$

under this mapping then, writing

$$\lambda^{(3)}{}^M_D = \pi^{(3)}{}_D\left(\lambda^{(2)}{}^M_D\right)$$
$$\xi\left(\lambda^{(3)}{}^M_D\right) = \pi^{(3)}{}_D\left(\xi\left(\lambda^{(2)}{}^M_D\right)\right)$$

and under this mapping

$$n_i + \Delta s_i = \left(\lambda^{(1)}{}^N_D\right)_j - \left(\lambda^{(3)}{}^M_D\right)_j + 2\, \xi\left(\lambda^{(3)}{}^M_D\right)_j$$
$$j = \left(\pi^{(1)}{}_D\right)_i$$

the permutation $\pi^{(3)}$, of course, is just

$$\pi^{(3)} = \pi^{(1)}{}_D \circ \pi^{(2)}{}_D{}^{-1}$$

under this relation we can now again ignore $\pi^{(1)}{}_D$ for a bit. In particular, we'll note that the sequence

$$\lambda^{(1)}{}^N_D - \lambda^{(3)}{}^M_D + 2\, \xi\left(\lambda^{(3)}{}^M_D\right)$$

is not necessarily an element of Λ^L_D. But it *can* be written as

$$\lambda^{(1)}{}^N_D - \lambda^{(3)}{}^M_D + 2\, \xi\left(\lambda^{(3)}{}^M_D\right) = \pi^{(4)}{}_D\left(\lambda^{(4)}{}^L_D\right)$$

where $\pi^{(4)}{}_D$ just puts the sequence in canonical order. It is unfortunately kinda hard to know what the form of $\pi^{(4)}{}_D$ looks like, since it depends on the relative sizes of $\lambda^{(3)}{}^M_D$, $\lambda^{(1)}{}^N_D$, and

$\xi\left(\lambda^{(3)M}_{D}\right)$, but it is conceivable that a fast algorithm could be constructed for $\lambda^{(1)N}_{D}-\lambda^{(3)M}_{D}$ and iteratively updated for each element of $2\ \xi\left(\lambda^{(3)M}_{D}\right)$.

Perhaps more productive is to look at the set of *all* selection rules coming from

$$H^{(i)} \sim \left(\mathcal{P}_D,\ \Lambda^{M}_{D}\right)$$

or more particularly, from the subset of $H^{(i)}$ that maps as $\left(\mathcal{P}_D,\ \lambda^{M}_{D}\right)$ for some partition λ. The benefit of this is now we don't need to worry about a specific

$$\pi^{(3)}=\pi^{(1)}{}_{D}\circ\pi^{(2)}{}_{D}{}^{-1}$$

but instead about the action of the entire space

$$\pi^{(1)}{}_{D}(\mathcal{P}_D)^{-1}=\mathcal{P}_D$$

that is, we get to write the space of coming from $\left(\mathcal{P}_D,\ \lambda^{M}_{D}\right)$ as

$$s(q)\circ\left|n_1,\ n_2,\ ...,\ n_D\right\rangle = \left\{\lambda^{(1)N}_{D}-\pi_D\,\lambda^{M}_{D}+2\ \xi\left(\pi_D\,\lambda^{M}_{D}\right)\right\}$$

for every π_D and ξ which means we can also write it as

$$s(q)\circ\left|n_1,\ n_2,\ ...,\ n_D\right\rangle = \left\{\lambda^{(1)N}_{D}+\pi_D\,\lambda^{M}_{D}-2\ \xi\left(\pi_D\,\lambda^{M}_{D}\right)\right\}$$

The way this can benefit us is by writing $\lambda^{(1)N}_{D}$ and λ^{M}_{D} as expansions over *classes* of elements with their multiplicities, i.e. we'll have

$$\lambda^{M}_{D} \sim (\chi_i,\ m_i)$$

this allows us to split our partition into "subspaces" from $[m_i,\ m_{i+1})$

Then just focusing on the term $\lambda^{(1)N}_{D}+\pi_D\,\lambda^{M}_{D}$ for every π_D, this gives us a way to efficiently resolve the set of *unique* terms under the various π_D applications. To outline the algorithm we'll start with

$$\lambda^{m}_{D} \sim (a_i,\ m_i)$$
$$\lambda^{(1)N}_{D} \sim \left(b_j,\ c_j\right)$$

then we have two sets of bins $[m_i,\ m_{i+1})$ and $\left[c_j,\ c_{j+1}\right)$ and the question is how many *unique* ways can we distribute the $[m_i,\ m_{i+1})$ across the $\left[c_j,\ c_{j+1}\right)$

We can do this in recursive fashion by recasting it as a balls/buckets type problem where we say I have m_i balls and I want to distribute them over the D buckets of size $\left\{c_j\right\}$. We'll note that in the case that the c_j are arbitrarily large, for each m_i we'd have the number of ways to distribute that m_i given by

$$\#\,(m_i,\ D) = \binom{D+m_i-1}{m_i}$$

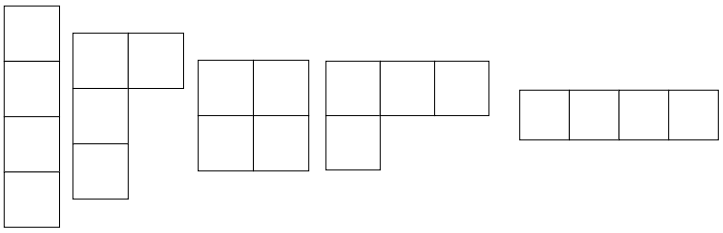
where $\binom{D+m_i-1}{m_i}$ is the binomial coefficient. This gives an upper bound on our partitions.

Tableau Examples

In[60]:=
```
showTableau[perm_] :=
 With[{v = ReverseSort[perm]},
  Graphics[{
    EdgeForm[Black],
    FaceForm[White],
    Table[
     Table[
      Rectangle[{i - 1, 1 - j}, {i, -j}],
      {j, v[[i]]}
     ],
     {i, Length[v]}
    ]
   },
   ImageSize → 30 * {Length[v], Max@v},
   AspectRatio → Automatic
  ]
 ]
```

In[61]:=
```
Grid@{showTableau /@ IntegerPartitions[4]}
```

Out[61]=



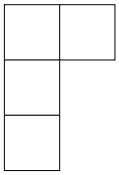
In[157]:=

```
directSum[v_, r_] :=
  Block[{
    totalLen = Length[v] + Length[r],
    padV,
    padR
   },
   padV = PadRight[v, totalLen];
   padR = PadRight[r, totalLen];
   DeleteDuplicates@Table[
     DeleteCases[ReverseSort[padV + p], 0],
     {p, Permutations[padR]}
    ]
  ];
conjPart[v_] :=
  Table[
   Total@UnitStep[v - i],
   {i, Max[v]}
  ];
tabIndex[v_] :=
  With[{ip = IntegerPartitions[Total[v]]},
   First@FirstPosition[ip, v] - (Length[ip] + 1) / 2
  ];
permDiffIdx[v1_, v2_] :=
 Block[{perm = v2 - PadRight[v1, Length[v2]]},
  First@
   FirstPosition[Permutations[ReverseSort[perm]], perm]
 ]
```

In[62]:=

```
showTableau@IntegerPartitions[4][[2]]
```

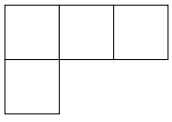
Out[62]=



In[83]:=

```
showTableau@conjPart@IntegerPartitions[4][[2]]
```

Out[83]=

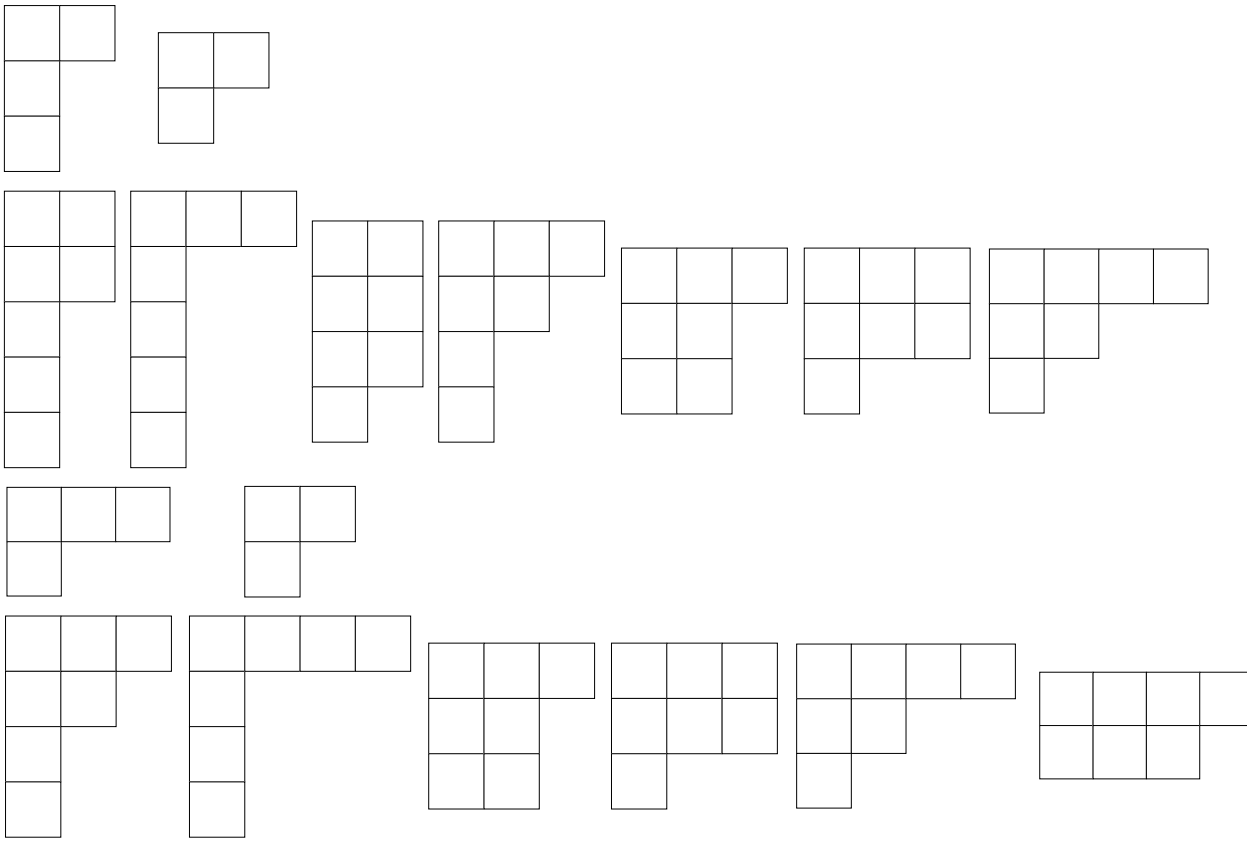


```
permDiffIdx
```

In[111]:=

```
Column@
 Table[
  With[{p = IntegerPartitions[3][[2]], n = 4},
   With[{ip = IntegerPartitions[Total[p] + n]},
    Grid@Map[
      showTableau,
      {
       {IntegerPartitions[n][[i]], p},
       SortBy[
        directSum[IntegerPartitions[n][[i]], p],
        FirstPosition[ip, #] &
       ]
      },
      {2}
     ]
   ]
  ],
  {i, {2, 4}}
 ]
```

Out[111]=



In[163]:=

```
Grid[
 Join @@
  With[{m = 3, n = 4},
   With[{p1 = IntegerPartitions[m],
     p2 = IntegerPartitions[n], ip = IntegerPartitions[m + n]},
    Table[
     {
      tabIndex@p2[[i]], tabIndex@p1[[j]],
      tabIndex /@
       directSum[p2[[i]], p1[[j]]],
      Column@{
        Row[showTableau /@ {p2[[i]], p1[[j]]}],
        Row[showTableau /@
          directSum[p2[[i]], p1[[j]]]]
       }
     },
     {i, Length[p2]},
     {j, Length[p1]}
    ]
   ]
  ],
 Alignment → {{Right, Right, Left, Left}, Center}
]
```

Character Classes

In[388]:=

```
ClearAll[fmtCharClasses, contractCharacterClasses, expandCharacterClasses];
fmtCharClasses[v_Association] :=
  Row[KeyValueMap[Superscript, v]];
fmtCharClasses[v_List] :=
  fmtCharClasses[contractCharacterClasses[v]];
contractCharacterClasses[v_List] :=
  Counts[v];
expandCharacterClasses[a_Association] :=
 Join @@ KeyValueMap[ConstantArray, a]
```

In[401]:=

```
fmtCharClasses /@
 {IntegerPartitions[10][[4]], {2, 1, 1}}
```

Out[401]=

$$\left\{8^1 1^2,\ 2^1 1^2\right\}$$

In[404]:=
```
directSum[
  IntegerPartitions[10][[4]],
  expandCharacterClasses[<|1 → 2, 2 → 1|>]
 ] // Map[fmtCharClasses]
```
Out[404]=
$\{9^1 3^1 2^1,\ 9^1 2^2 1^1,\ 9^1 3^1 1^2,\ 9^1 2^1 1^3,\ 10^1 2^2,\ 10^1 2^1 1^2,\ 10^1 1^4,\ 8^1 2^3,\ 8^1 3^1 2^1 1^1,\ 8^1 2^2 1^2,\ 8^1 3^1 1^3,\ 8^1 2^1 1^4\}$

In[395]:=
```
Permutations[IntegerPartitions[10][[-9]]] // Length
```
Out[395]=
```
60
```

In[54]:=
```
Permutations[{ 2, 1, 1}] // Length
```
Out[54]=
```
3
```

In[83]:=
```
IntegerPartitions[10][[2]]
```
Out[83]=
```
{9, 1}
```

In[85]:=
```
IntegerPartitions[10][[4]]
```
Out[85]=
```
{8, 1, 1}
```

In[86]:=
```
Table[
 directSum[
   IntegerPartitions[10][[2]],
   expandCharacterClasses[<|1 → i|>]
  ] // Length,
 {i, 10}
]
```
Out[86]=
```
{3, 4, 4, 4, 4, 4, 4, 4, 4, 4}
```

In[78]:=
```
fmtCharClasses[IntegerPartitions[10][[-9]]]
```
Out[78]=
$3^1 2^2 1^3$

In[113]:=
```
hmmm =
  Table[
   Table[
    Table[
     directSum[
       IntegerPartitions[10][[n]],
       expandCharacterClasses[<|1 → i, 2 → j|>]
      ] // Length,
     {j, i, 5}
    ],
    {i, 5}
   ],
   {n, Length[IntegerPartitions[10]]}
  ];
```

In[110]:=
```
makeLTMat[lists_] :=
  Developer`ToPackedArray@
   Map[PadLeft[#, Length[lists]] &, lists];
```

In[112]:=
```
MatrixForm[makeLTMat[#]] & /@ %92 // Column
```

In[77]:=
```
Table[
 Table[
  directSum[
    IntegerPartitions[10][[-9]],
    expandCharacterClasses[<|1 → i, 2 → j|>]
   ] // Length,
  {j, 4}
 ],
 {i, 4}
]
```

Out[77]=
```
{{13, 25, 37, 46}, {25, 43, 59, 70}, {37, 59, 77, 88}, {46, 70, 88, 99}}
```

Operator-Based Filtering

We can get a significant benefit by noting that we usually care about properties, which if we're doing the PT up to order N we do like

$$\langle g| A |n\rangle = \sum_{k=0}^{N}\sum_{i=0}^{k}\sum_{j=0}^{k-i} \langle g^{(i)}| A^{(k-i-j)} |n^{(j)}\rangle$$

or equivalently

$$\begin{aligned}\langle g| A |n\rangle &= \sum_{k=0}^{N}\sum_{j=0}^{k}\sum_{i=0}^{k-i} \langle g^{(i)}| A^{(k-i-j)} |n^{(j)}\rangle \\ &= \sum_{k=0}^{N}\sum_{j=0}^{k}\left(\sum_{i=0}^{k-j} \langle g^{(i)}| A^{(k-i-j)}\right)|n^{(j)}\rangle\end{aligned}$$

then if the expansion of A supports selection rules as well this provides a secondary filter which is especially useful if the number of initial states in g is less than the number of final states requested

We need to also make sure that the energies can be calculated correctly, i.e. that we have the necessary terms for

$$E_n^{(k)} = \langle n^{(0)}| H^{(k)} |n^{(0)}\rangle + \sum_{i=1}^{k-1}\langle n^{(0)}| H^{(k-i)} |n^{(i)}\rangle - E_n{}^{(k-i)} \langle n^{(0)} | n^{(i)}\rangle$$

but that reduces to just having the terms necessary to evaluate

$$\sum_{i=0}^{k-1}\langle n^{(0)}| H^{(k-i)} |n^{(i)}\rangle$$

where the terms connected to $\langle n^{(0)}| H^{(k-i)}$ are in fact just a subset of the terms required to evaluate $\langle n^{(k-i)}|$

Therefore the entire state space that $|n^{(j)}\rangle$ could to connect to for non-zero elements (which can include states $|n^{(j)}\rangle$ doesn't connect to) can be given by

$$\varphi\left(|n^{(j)}\rangle\right) = \bigcup_{k=0}^{N} \Phi\left(\langle n^{(0)}| H^{(k-j)}\right) \cup \left(\bigcup_{i=0}^{k-j} \Phi\left(\langle g^{(i)}| A^{(k-i-j)}\right)\right)$$

where Φ is just the total connected space for that term, given by

$$\Phi\left(|n^{(j)}\rangle\right) = \bigcup_{i=0}^{j-1} \Phi\left(H^{(j-i)} |n^{(i)}\rangle\right) \cup \left\{|n^{(0)}\rangle\right\}$$

and for convenience we'll write

$$\Gamma^{(k)} = \left(\bigcup_{i=0}^{k} \Phi\left(\langle g^{(i)}| A^{(k-i)}\right)\right)$$

so

$$\varphi\left(|n^{(j)}\rangle\right) = \bigcup_{k=0}^{N} \Phi\left(\langle n^{(0)}| H^{(k-j)}\right) \cup \Gamma^{(k-j)}$$

One thing to note is that we need to be sure that we're calculating enough elements of $\langle g^{(i)}|$, in particular we have

$$\varphi\left(\left|n^{(0)}\right\rangle\right) = \bigcup_{k=0}^{N} \Phi\left(\left\langle n^{(0)}\right| H^{(k)}\right) \cup \Gamma^{(k)}$$
$$\Phi\left(\left|n^{(0)}\right\rangle\right) = \left|n^{(0)}\right\rangle$$

and so we'll need for all k

Looking at the most common case explicitly, for N=2 then we have

$$\begin{aligned}
\Phi\left(\left|n^{(1)}\right\rangle\right) &= \Phi\left(H^{(1)}\left|n^{(0)}\right\rangle\right) \cup \left\{\left|n^{(0)}\right\rangle\right\} \\
\varphi\left(\left|n^{(1)}\right\rangle\right) &= \Phi\left(\left\langle n^{(0)}\right| H^{(1)}\right) \cup \Phi\left(\left\langle n^{(0)}\right| H^{(0)}\right) \cup \Gamma^{(1)} \\
&= \Phi\left(H^{(1)}\left|n^{(0)}\right\rangle\right) \cup \Phi\left(\left\{\left|n^{(0)}\right\rangle\right\}\right) \cup \Gamma^{(1)} \\
\Phi\left(\left|n^{(2)}\right\rangle\right) &= \Phi\left(H^{(2)}\left|n^{(0)}\right\rangle\right) \cup \Phi\left(H^{(1)}\left|n^{(1)}\right\rangle\right) \cup \left\{\left|n^{(0)}\right\rangle\right\} \\
\varphi\left(\left|n^{(2)}\right\rangle\right) &= \left\{\left|n^{(0)}\right\rangle\right\} \cup \Gamma^{(0)}
\end{aligned}$$

then from this we can note that we need the entirety of as

$$\Phi\left(\left|n^{(1)}\right\rangle\right) \cap \varphi\left(\left|n^{(1)}\right\rangle\right) = \Phi\left(\left|n^{(1)}\right\rangle\right)$$

but for the second order corrections we only need

$$\Phi\left(H^{(1)}\left|n^{(1)}\right\rangle\right) \cap \Gamma^{(0)} \cap \left\{\left|n^{(0)}\right\rangle\right\}$$

and by Hermiticity we don't need to calculate the terms from ∩ as these are already calculated for since the matrix elements of $H^{(1)}$ that connect to are the complex conjugates of the ones that connect to

Therefore the only remaining terms we might need to calculate come from

which if the selection rules of $H^{(1)}$ are symmetric (i.e. any ways they can raise they can lower) we can also write as

$$\Phi\left(H^{(1)}\left|n^{(1)}\right\rangle\right) \cap \Gamma^{(0)} = \Phi\left(\left|n^{(1)}\right\rangle\right) \cap \Phi\left(H^{(1)}\,\Gamma^{(0)}\right)$$

and then if we note we don't need to calculate any more terms. If it is not, we need to calculate .

The same logic applies to , giving us

$$\Phi\left(H^{(2)}\left|n^{(0)}\right\rangle\right) \cap \Gamma^{(0)} = \Phi\left(\left|n^{(0)}\right\rangle\right) \cap \Phi\left(H^{(2)}\,\Gamma^{(0)}\right)$$

Old Too Hard Approach

Paper Write-Up

It is generally the case that we have a property or set of properties that we are interested in evaluating. If that is the case we can potentially reduce the number of terms we need to evaluate and store. To understand how this works, we can note that the evaluation of properties between some initial state, g and a given other state, n, in perturbation theory looks like

$$\langle g| A |n\rangle = \sum_{k=0}^{N} \left(\langle g| A |n\rangle\right)^{(k)}$$

$$\left(\langle g| A |n\rangle\right)^{(k)} = \sum_{i=0}^{k}\sum_{j=0}^{k-i} \langle g^{(i)}| A^{(k-i-j)} |n^{(j)}\rangle$$

and to evaluate a given term in this expansion we only to support terms connected to through the action of $A^{(k-i-j)}$. Focusing initially on the case that $k{=}N$, we have

$$\left(\langle g| A |n\rangle\right)^{(N)} = \sum_{i=0}^{N}\sum_{j=0}^{N-i} \langle g^{(i)}| A^{(N-i-j)} |n^{(j)}\rangle$$

for each of these terms, then, the most efficient evaluation strategy will be to consider whether $i > j$ and then supposing $i > j$ to look at the state space connected to and consider its intersection with the state space for . It is, however, hard to implement this transformation completely generally without reduplicating much of the work already required in the perturbation theory calculation. Most of the benefit, too, will come from finding a more efficient strategy to evaluate and , and so we will focus on that. This filter, then, can be constructed most easily by looking at and . The state space touched by need only be intersected with and similarly that touched by need only intersect with .

A naive implementation of this filter would still require constructing the total state space for , which will become increasingly expensive as the system size increases. To mitigate this effect we can pre-filter the state space of by looking at the total numbers of quanta in the accessed states. For example, let's assume that $A^{(0)}$ can change the total number of quanta by ±1. Then let's suppose that has 1 total quantum of excitation and has 2. This means can have 0 or 2 quanta of excitation and therefore the state spaces accessed in need only have that many quanta as well. This can occur through many paths, but the straightforward approach is to note that we have terms that look like

$$\sum_i H^{(N-i)} \left| n^{(i)} \right\rangle$$

and so when looking at, e.g., , we only need to consider when one of the selection rules for $H^{(1)}$ will take one of the elements of to a state with 0 or 2 quanta of excitation.

As a final note, by assuming $A^{(0)}$ and the $H^{(i)}$ are Hermitian, we can make use of symmetry to avoid needing to calculate most of these elements. This can be implemented as a filter as well, which provides even more significant speed ups. As a concrete example,

---

Morse Expansions

Δr

The basic model is a stretch with reduced mass μ and displacement from the equilibrium length Δr. At the harmonic level, we have

$$H = \frac{{p_r}^2}{2\mu} + \frac{1}{2}\mu\omega^2 \Delta r^2$$

In[409]:=

```
dumpTeXString@PreviousCell[]
```

Out[409]=

```
H=\frac{p_r^2}{2 \mu }+\frac{1}{2} \mu  \omega ^2 \Delta  r^2
```

which can be put in dimensionless coordinates by defining

$$\Delta r = \sqrt{\hbar/\mu\omega}\ q$$
$$p_r = \sqrt{\hbar\mu\omega}\ p$$

In[410]:=

```
dumpTeXString@PreviousCell[]
```

Out[410]=

```
\left(
\begin{array}{c}
 \Delta  r=\sqrt{\frac{\hbar  \omega }{\mu }} q \\
 p_r=\sqrt{\hbar  \mu  \omega } p \\
\end{array}
\right)
```

which gives the form

$$H = \frac{\hbar\omega}{2}\left(p^2 + q^2\right)$$

We can then do the same to a series expansion of the Morse oscillator, with the Hamiltonian

$$H = \frac{{p_r}^2}{2\mu} + D_e\left(1-e^{-\alpha \Delta r}\right)^2$$

where D_e is the dissociation energy and α is the Morse parameter. We can expand the potential as

$$D_e\left(1-e^{-\alpha \Delta \mathrm{r}}\right)^2 = \sum_{i=2}^{\infty} \frac{(-1)^i}{i!} \left(2^i-2\right) D_e\, a^i\, \Delta r^i$$

In[411]:=

```
dumpTeXString@PreviousCell[]
```

Out[411]=

```
D_e\left(1-e^{-\alpha \, \text{$\Delta $r}}\right){}^2 = \sum
  _{i=2}^{\infty }  \frac{(-1)^i}{i\, !}\left(2^i-2\right)D_e\, a^i\Delta r^i
```

and putting this all into dimensionless coordinates we have

$$H = \frac{\hbar\omega}{2}\left(p^2 + q^2\right) + \sum_{i=3}^{\infty} (-1)^i \frac{\left(2^i-2\right)}{i!} \hbar^{(i/2)} D_e\left(\sqrt{a^2/\mu\omega}\right)^i q^i$$

In[412]:=

```
dumpTeXString@PreviousCell[]
```

Out[412]=

```
H = \frac{\hbar \omega }{2}\left(\, p^2 + q^2\right) + \sum
  _{i=3}^{\infty }  (-1)^i\frac{\left(2^i-2\right)}{i\, !}\hbar ^{(i/\,
  2)}D_e\, \left(\sqrt{\left.a^2\right/\mu \, \omega }\right){}^iq^i
```

The potential can also be expressed in terms of ω and ωx through the relations $D_e = \hbar\omega^2/4\,\omega x$, $\alpha = \sqrt{2\,\mu\omega x/\hbar}$ as

$$V(q;\, \omega,\, \omega x) = \hbar\omega \sum_{i=2}^{\infty} \frac{(-1)^i}{i!} \left(2^{i-1}-1\right)\left(\frac{2\,\omega x}{\omega}\right)^{(i/2-1)} q^i$$

but it is worth noting that by introducing ωx we have lost the explicit $\hbar^{(i/2)}$ form and so [[Equation ??]] will be used to expand the Hamiltonian as

y

We start with

$$H = \frac{p_R^2}{2\mu} + D_e\left(1-e^{-\alpha \Delta \mathrm{R}}\right)^2$$

then we let

$$y = \frac{1-e^{-\alpha \Delta \mathrm{R}}}{\alpha}$$

performing this substitution

$$\frac{d}{dR} = \frac{dy}{dR}\,\frac{d}{dy}$$
$$= e^{-\alpha \Delta R}\,\frac{d}{dy}$$
$$= (1-\alpha y)\,\frac{d}{dy}$$
$$\Rightarrow g_{yy} = \frac{(1-\alpha y)^2}{\mu}$$

and then we pick up the V' term from

$$V' = \frac{\hbar^2}{8}\left(\frac{dg_{yy}}{dy}\,\frac{d\ln\left(1/g_{yy}\right)}{dy} + g_{yy}\left(\frac{d^2\ln\left(1/g_{yy}\right)}{dy\,dy} + \frac{1}{4}\,\frac{d\ln\left(1/g_{yy}\right)}{dy}\,\frac{d\ln\left(1/g_{yy}\right)}{dy}\right)\right)$$
$$= -\frac{\hbar^2}{8}\,\frac{\alpha^2}{\mu}$$

and recognizing that $\omega = \sqrt{2\,D_e\,\alpha^2/\mu}$ we have

$$H = \frac{1}{2\mu}\,p_y(1-\alpha y)^2\,p_y - \frac{\hbar^2}{8}\,\frac{\alpha^2}{\mu} + \frac{\mu\omega^2}{2}\,y^2$$

In[526]:=

```
dumpTeXString@PreviousCell[]
```

Out[526]=

```
H=\frac{p_y(1-\alpha  y){}^2 p_y}{2 \mu }-\frac{\hbar ^2
  \alpha ^2}{8 \mu }+\frac{1}{2} \left(\mu  \omega ^2\right) y^2
```

then writing $y = \sqrt{\hbar\,/\,\mu\omega}\;q$, $p_y = \sqrt{\hbar\mu\omega}$, and expanding

$$H = \frac{\hbar\omega}{2}\left(p\left(1-\sqrt{\hbar}\,\sqrt{\frac{\alpha^2}{\mu\omega}}\;q\right)^2 p + q^2\right) - \frac{\hbar^2}{8}\,\frac{\alpha^2}{\mu}$$
$$= \frac{\hbar\omega}{2}\left(p^2+q^2\right) - \hbar^{(3/2)}\,\frac{1}{2}\,\sqrt{\frac{\alpha^2\,\omega}{\mu}}\;pqp + \hbar^2\,\frac{\alpha^2}{\mu}\,pqqp - \frac{\hbar^2}{8}\,\frac{\alpha^2}{\mu}$$
$$H = \frac{\hbar\omega}{2}\left(p^2+q^2\right) - \frac{\hbar^{(3/2)}}{2}\,\sqrt{\frac{\alpha^2\,\omega}{\mu}}\;pqp + \frac{\hbar^2}{2}\,\frac{\alpha^2}{\mu}\,pqqp - \frac{\hbar^2}{8}\,\frac{\alpha^2}{\mu}$$

In[527]:=

```
dumpTeXString@PreviousCell[]
```

Out[527]=

```
H=\frac{1}{2} \hbar \omega  \left(p^2+q^2\right)-\frac{1}{2} \hbar
  ^{3/2} \sqrt{\frac{\alpha ^2 \omega }{\mu }} p q p+\frac{\hbar
  ^2 \alpha ^2 p q q p}{\mu }-\frac{\hbar ^2 \alpha ^2}{8 \mu }
```

2n+1 Stuff

The goal is to find a compact way to rexpress

$$E_n^{(k)}=\left\langle n^{(0)}\right| H^{(k)} \left|n^{(0)}\right\rangle + \sum_{i=1}^{k-1}\left\langle n^{(0)}\right| \Delta \mathrm{H}_n^{(k-i)} \left|n^{(i)}\right\rangle$$

through recursions to relate

$$\left\langle n^{(0)}\right| \Delta \mathrm{H}_n^{(k-i)} \left|n^{(i)}\right\rangle$$

to an expression of the form

$$\left\langle n^{(j)}\right| H^{(k-i)} \left|n^{(i-j)}\right\rangle$$

Löwdin's trick is to rearrange our perturbation expressions so we'll see what happens... We start from the top and just to make life nicer we're going to always focus on one state so we can write

$$\Delta(0) \left|k\right\rangle=-\sum_{i=0}^{k-1}\Delta(k-i) \left|i\right\rangle$$

where $\left|k\right\rangle=\left|n^{(k)}\right\rangle$ and $\Delta(i)=\Delta \mathrm{H}_n^{(i)}$

Then apply on the left with an arbitrary $\left\langle j+1\right|$

$$\left\langle j+1\right| \Delta(0) \left|k\right\rangle = -\sum_{i=0}^{k-1}\left\langle j+1\right| \Delta(k-i) \left|i\right\rangle$$

$$\left\langle k\right| \Delta(0) \left|j+1\right\rangle = -\left\langle k\right| \Delta(1) \left|j\right\rangle -\sum_{i=0}^{j-1}\left\langle k\right| \Delta(j+1-i) \left|i\right\rangle$$

but by Hermiticity we have

$$\left\langle j\right| \Delta(1) \left|k\right\rangle = \left\langle j+1\right| \Delta(1) \left|k-1\right\rangle + \sum_{i=0}^{k-2}\left\langle j+1\right| \Delta(k-i) \left|i\right\rangle - \sum_{i=0}^{j-1}\left\langle k\right| \Delta(j+1-i) \left|i\right\rangle$$

Then we can take this and try to bootstrap, like

$$\left\langle 0\right| \Delta(1) \left|k\right\rangle = \left\langle 1\right| \Delta(1) \left|k-1\right\rangle + \sum_{i=0}^{k-2}\left\langle 1\right| \Delta(k-i) \left|i\right\rangle$$

$$\langle 1|\,\Delta\,(1)\,|k{-}1\rangle = \langle 2|\,\Delta\,(1)\,|k{-}2\rangle + \sum_{i=0}^{k-3}\langle 2|\,\Delta(k{-}1{-}i)\,|i\rangle - \langle k{-}1|\,\Delta(2)\,|0\rangle$$

$$\langle 2|\,\Delta\,(1)\,|k{-}2\rangle = \langle 3|\,\Delta(1)\,|k{-}3\rangle + \ldots$$

so then

$$1)\,|k\rangle = \langle p|\,\Delta(1)\,|k{-}p\rangle + \sum_{a=1}^{p}\left(\sum_{i=0}^{k-1-a}\langle a|\,\Delta(k{+}1{-}a{-}i)\,|i\rangle\right) - \sum_{b=2}^{p}\sum_{i=0}^{b-2}\langle k{+}1{-}b|\,\Delta(b{-}i)\,|i\rangle$$

$$= \langle p|\,\Delta(1)\,|k{-}p\rangle + \sum_{a=1}^{p}\left(\sum_{i=0}^{k-1-a}\langle a|\,\Delta(k{+}1{-}a{-}i)\,|i\rangle\right) - \left(\sum_{b=2}^{p}\sum_{i=1}^{b-2}\langle k{+}1{-}b|\,\Delta(b{-}i)\,|i\rangle\right) - \sum_{b=2}^{p}\langle k{+}1{-}b|\,\Delta(b)\,|($$

$$= \langle p|\,\Delta(1)\,|k{-}p\rangle + \sum_{a=1}^{p}\left(\sum_{i=0}^{k-1-a}\langle a|\,\Delta(k{+}1{-}a{-}i)\,|i\rangle\right) - \left(\sum_{b=2}^{p}\sum_{i=1}^{b-2}\langle k{+}1{-}b|\,\Delta(b{-}i)\,|i\rangle\right) - \sum_{i=k+1-p}^{k-1}\langle 0|\,\Delta(k{+}1{-}$$

and therefore

$$\langle 0|\,\Delta(1)\,|2n\rangle = \langle n|\,\Delta(1)\,|n\rangle + \left(\sum_{a=1}^{n}\sum_{i=0}^{2\,n-1-a}\langle a|\,\Delta(2n{+}1{-}a{-}i)\,|i\rangle\right) -$$

$$\left(\sum_{b=2}^{n}\sum_{i=1}^{b-2}\langle 2n{+}1{-}b|\,\Delta(b{-}i)\,|i\rangle\right) - \sum_{i=n+1}^{2n-1}\langle 0|\,\Delta(2n{+}1{-}i)\,|i\rangle$$

and we can just plug this in, but first we might be able to simplify a bit earlier. We start by reversing a summation

$$\left(\sum_{a=1}^{p}\sum_{i=0}^{k-1-a}\langle a|\,\Delta(k{+}1{-}a{-}i)\,|i\rangle\right) = \left(\sum_{b=2}^{p}\sum_{i=1}^{b-2}\langle k{+}1{-}b|\,\Delta(b{-}i)\,|i\rangle\right)$$

2k+1

Derivation

$$\sum_{i=1}^{2n}\langle 0|\,\Delta(2n{+}1{-}i)\,|i\rangle$$

$$\langle 0|\,\Delta(1)\,|2n\rangle + \sum_{i=1}^{2n-1}\langle 0|\,\Delta(2n{+}1{-}i)\,|i\rangle$$

$$\langle n|\,\Delta(1)\,|n\rangle + \left(\sum_{a=1}^{n}\sum_{i=0}^{2\,n-1-a}\langle a|\,\Delta(2n{+}1{-}a{-}i)\,|i\rangle\right) - \left(\sum_{b=2}^{n}\sum_{i=1}^{b-2}\langle 2n{+}1{-}b|\,\Delta(b{-}i)\,|i\rangle\right) + \sum_{i=1}^{2n-1}\langle 0|\,\Delta(2n{+}1{-}i)\,|i\rangle$$

$$\langle n|\,\Delta(1)\,|n\rangle + \sum_{i=1}^{n}\langle 0|\,\Delta(2n{+}1{-}i)\,|i\rangle + \left(\sum_{a=1}^{n}\sum_{i=0}^{2\,n-1-a}\langle a|\,\Delta(2n{+}1{-}a{-}i)\,|i\rangle\right) - \left(\sum_{b=2}^{n}\sum_{i=1}^{b-2}\langle 2n{+}1{-}b|\,\Delta(b{-}i)\,|i\rangle\right)$$

Now the issue is that we still have

$$\Delta(2n+1-a-i)\left|i\right\rangle\Big) - \left(\sum_{b=2}^{n}\sum_{i=1}^{b-2}\left\langle 2n+1-b\right|\Delta(b-i)\left|i\right\rangle\right) = \left(\sum_{a=1}^{n}\sum_{i=0}^{2n-1-a}\left\langle a\right|\Delta(2n+1-a-i)\left|i\right\rangle\right) - \left(\sum_{b=0}^{n-2}\sum_{i=1}^{b}\left\langle 2n-1-b\right|\right.$$
$$= \left(\sum_{a=1}^{n}\sum_{i=0}^{2n-1-a}\left\langle a\right|\Delta(2n+1-a-i)\left|i\right\rangle\right) - \left(\sum_{b=0}^{n-2}\sum_{i=1}^{b}\left\langle 2n-1-b\right|\right.$$

However it should be clear that we will have at least some cancellation, which we can check numerically. In particular, any term involving terms beyond should disappear, so we'll isolate those terms in the LHS sum (adding one one extra term to keep the sums clean)

$$1)\left|n\right\rangle + \left(\sum_{a=1}^{n}\sum_{i=0}^{2n-1-a}\left\langle a\right|\Delta(2n+1-a-i)\left|i\right\rangle\right) = \left(\sum_{a=1}^{n}\sum_{i=0}^{n}\left\langle a\right|\Delta(2n+1-a-i)\left|i\right\rangle\right) + \left(\sum_{a=1}^{n-2}\sum_{i=n+1}^{2n-1-a}\left\langle a\right|\Delta(2n+1-a-i\right.$$
$$= \left(\sum_{a=1}^{n}\sum_{i=0}^{n}\left\langle a\right|\Delta(2n+1-a-i)\left|i\right\rangle\right) + \left(\sum_{a=1}^{n-2}\sum_{i=0}^{n-2-a}\left\langle n+1+i\right|\Delta(n-a-i\right.$$

and then we have

$$\sum_{a=1}^{n}\sum_{i=0}^{n-2-a}\left\langle n+1+i\right|\Delta(n-a-i)\left|a\right\rangle = \sum_{b=0}^{n-2}\sum_{i=1}^{b}\left\langle 2n-1-b\right|\Delta(b+2-i)\left|i\right\rangle$$
$$= \sum_{b=0}^{n-2}\sum_{j=1}^{n-2-b}\left\langle n+1+b\right|\Delta(n-b-j)\left|j\right\rangle \text{ [reversing sum]}$$

and so the two are equivalent, giving us

Expression

$$E^{(2k+1)} = \left\langle n^{(0)}\right| H^{(2k+1)}\left|n^{(0)}\right\rangle + \sum_{i=1}^{k}\left\langle n^{(0)}\right|\Delta\mathrm{H}_{n}^{(2k+1-i)}\left|n^{(i)}\right\rangle + \sum_{a=1}^{k}\sum_{i=0}^{k}\left\langle n^{(a)}\right|\Delta\mathrm{H}_{n}^{(2\,k+1-a-i)}\left|n^{(i)}\right\rangle$$

2k

Derivation

$$\left\langle 0\right| H^{(2n)}\left|0\right\rangle + \sum_{i=1}^{2n-1}\left\langle 0\right|\Delta(2n-i)\left|i\right\rangle$$

$$\left\langle 0\right| H^{(2n)}\left|0\right\rangle + \left\langle 0\right|\Delta(1)\left|2n-1\right\rangle + \sum_{i=1}^{2n-2}\left\langle 0\right|\Delta(2n-i)\left|i\right\rangle$$

$$\left\langle 0\right| H^{(2n)}\left|0\right\rangle + \left\langle n\right|\Delta(1)\left|n-1\right\rangle + \sum_{i=1}^{n-1}\left\langle 0\right|\Delta(2n-i)\left|i\right\rangle + \sum_{a=1}^{n}\left(\sum_{i=0}^{2n-2-a}\left\langle a\right|\Delta(2n-a-i)\left|i\right\rangle\right) - \left(\sum_{b=2}^{n}\sum_{i=1}^{b-2}\left\langle 2n-b\right|\Delta(b-\right.$$

and again we can simplify by splitting out the problematic terms

$$(2n-a-i)\left|i\right\rangle\Big) = \sum_{a=1}^{n-2}\Big(\sum_{i=0}^{2n-2-a}\left\langle a\right|\Delta(2n-a-i)\left|i\right\rangle\Big) + \sum_{a=n-1}^{n}\Big(\sum_{i=0}^{2n-2-a}\left\langle a\right|\Delta(2n-a-i)\left|i\right\rangle\Big)$$

$$= \sum_{a=1}^{n-2}\Big(\sum_{i=n+1}^{2n-2-a}\left\langle a\right|\Delta(2n-a-i)\left|i\right\rangle\Big) + \sum_{a=1}^{n-2}\Big(\sum_{i=0}^{n}\left\langle a\right|\Delta(2n-a-i)\left|i\right\rangle\Big) + \sum_{a=n-1}^{n}\Big(\sum_{i=0}^{2n-2-a}\left\langle a\right|\Delta(2n-a-i$$

$$= \sum_{a=1}^{n-2}\Big(\sum_{i=0}^{n-3-a}\left\langle a\right|\Delta(n-1-a-i)\left|n+1+i\right\rangle\Big) + \sum_{a=1}^{n-2}\Big(\sum_{i=0}^{n}\left\langle a\right|\Delta(2n-a-i)\left|i\right\rangle\Big) + \sum_{a=0}^{1}\Big(\sum_{i=0}^{n-1-a}\left\langle n-1+a\right|\Delta$$

and this time around we will also want to split

$$\Big(\sum_{b=2}^{n}\sum_{i=1}^{b-2}\left\langle 2n-b\right|\Delta(b-i)\left|i\right\rangle\Big) = \Big(\sum_{b=2}^{n-1}\sum_{i=1}^{b-2}\left\langle 2n-b\right|\Delta(b-i)\left|i\right\rangle\Big) + \sum_{i=1}^{n-2}\left\langle n\right|\Delta(n-i)\left|i\right\rangle$$

since we have

$$\sum_{a=0}^{1}\Big(\sum_{i=0}^{n-1-a}\left\langle n-1+a\right|\Delta(n+1-a-i)\left|i\right\rangle\Big) = \sum_{i=0}^{n-1}\left\langle n-1\right|\Delta(n+1-i)\left|i\right\rangle + \sum_{i=0}^{n-2}\left\langle n\right|\Delta(n-i)\left|i\right\rangle$$

giving us

$$2n-a-i)\left|i\right\rangle\Big) - \Big(\sum_{b=2}^{n}\sum_{i=1}^{b-2}\left\langle 2n-b\right|\Delta(b-i)\left|i\right\rangle\Big) =$$

$$\left|\Delta(n-1-a-i)\left|n+1+i\right\rangle\Big) + \sum_{a=1}^{n-2}\Big(\sum_{i=0}^{n}\left\langle a\right|\Delta(2n-a-i)\left|i\right\rangle\Big) + \sum_{i=0}^{n-1}\left\langle n-1\right|\Delta(n+1-i)\left|i\right\rangle + \left\langle n\right|\Delta(n)\left|0\right\rangle - \Big(\sum_{b=2}^{n-1}\sum_{i=1}^{b-2}\left\langle 2\right.$$

finally, again, we'll hopefully be able to make a 1-1 map to show (note necessary transpose)

$$\sum_{a=1}^{n-2}\sum_{i=0}^{n-3-a}\left\langle n+1+i\right|\Delta(n-1-a-i)\left|a\right\rangle = \sum_{b=2}^{n-1}\sum_{i=1}^{b-2}\left\langle 2n-b\right|\Delta(b-i)\left|i\right\rangle$$

This works numerically and I assume can be shown by the same approach as before.

Therefore

$$\sum_{a=1}^{n}\Big(\sum_{i=0}^{2n-2-a}\left\langle a\right|\Delta(2n-a-i)\left|i\right\rangle\Big) - \Big(\sum_{b=2}^{n}\sum_{i=1}^{b-2}\left\langle 2n-b\right|\Delta(b-i)\left|i\right\rangle\Big) =$$

$$\sum_{a=1}^{n-2}\Big(\sum_{i=0}^{n}\left\langle a\right|\Delta(2n-a-i)\left|i\right\rangle\Big) + \sum_{i=0}^{n-1}\left\langle n-1\right|\Delta(n+1-i)\left|i\right\rangle + \left\langle n\right|\Delta(n)\left|0\right\rangle$$

Numerical Tests

Expression

$$H^{(2k)}\left|n^{(0)}\right\rangle + \left\langle n^{(k)}\right|\Delta \mathrm{H}_n^{(1)}\left|n^{(k-1)}\right\rangle + \sum_{i=1}^{k}\left\langle n^{(0)}\right|\Delta \mathrm{H}_n^{(2k-i)}\left|n^{(i)}\right\rangle + \sum_{a=1}^{k-2}\left(\sum_{i=0}^{k}\left\langle n^{(a)}\right|\Delta \mathrm{H}_n^{(2k-a-i)}\left|n^{(i)}\right\rangle\right) + \sum_{i=0}^{k-1}\left\langle n^{(k-1)}\right|\Delta \mathrm{H}$$

$$H^{(2k)}\left|n^{(0)}\right\rangle + \sum_{i=1}^{k}\left\langle n^{(0)}\right|\Delta \mathrm{H}_n^{(2k-i)}\left|n^{(i)}\right\rangle + \sum_{a=1}^{k-1}\sum_{i=0}^{k}\left\langle n^{(a)}\right|\Delta \mathrm{H}_n^{(2k-a-i)}\left|n^{(i)}\right\rangle$$

$$E^{(2k+1)} = \left\langle n^{(0)}\right| H^{(2k+1)}\left|n^{(0)}\right\rangle + \sum_{i=1}^{k}\left\langle n^{(0)}\right|\Delta \mathrm{H}_n^{(2k+1-i)}\left|n^{(i)}\right\rangle + \sum_{a=1}^{k}\sum_{i=0}^{k}\left\langle n^{(a)}\right|\Delta \mathrm{H}_n^{(2\,k+1-a-i)}\left|n^{(i)}\right\rangle$$

$$E^{(2k)} = \left\langle n^{(0)}\right| H^{(2k)}\left|n^{(0)}\right\rangle + \sum_{i=1}^{k}\left\langle n^{(0)}\right|\Delta \mathrm{H}_n^{(2k-i)}\left|n^{(i)}\right\rangle + \sum_{a=1}^{k-1}\sum_{i=0}^{k}\left\langle n^{(a)}\right|\Delta \mathrm{H}_n^{(2k-a-i)}\left|n^{(i)}\right\rangle$$

Images