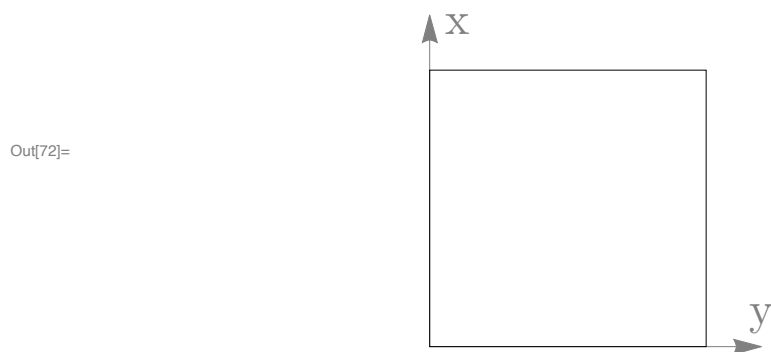
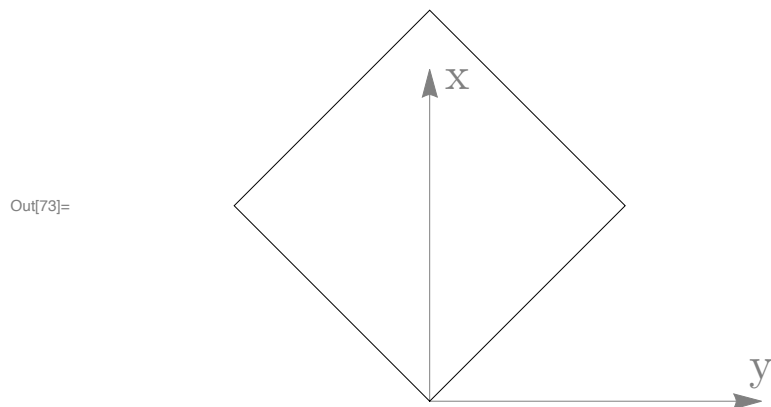

Vibrational Coordinates

We're going to ignore molecules for as long as we possibly can, since they're confusing and have too much associated baggage. Instead, we'll talk about 2D shapes to build intuition and show how we make a mapping between molecules and shapes later.

First, off, let's say we have a square

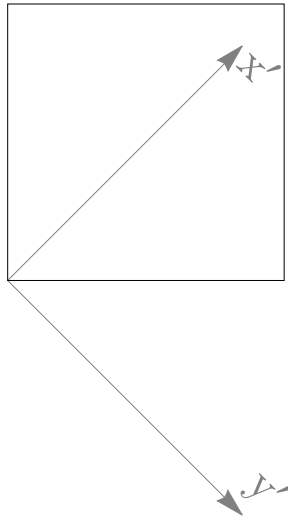


We can also imagine we have a diamond



But by rotating our axes, this diamond shape can be seen to be exactly the same as the original square

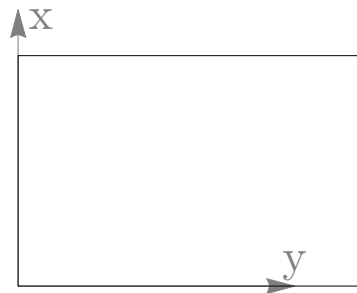
Out[78]=



What does this mean mathematically? Well just that by doing a *coordinate transformation* we can turn something that looks like a square into something that looks like a diamond. The object remains the exact same, all that changes are the coordinates we use to describe it.

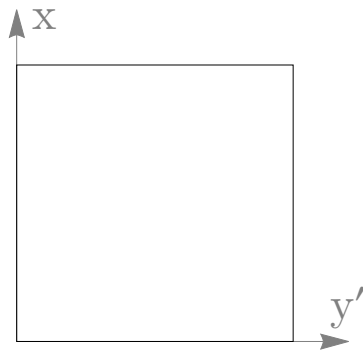
We can also imagine we have a rectangle instead of a square

Out[128]=



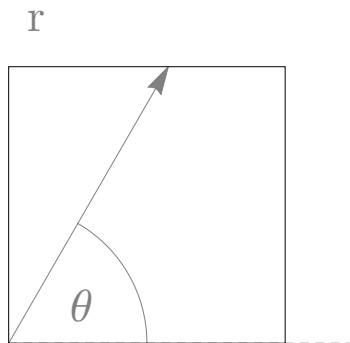
but if we just scale our axes appropriately, we can see that this *too* is just a square

Out[132]=



Another flavor of coordinate transformations that you probably saw in calc. is the polar coordinate transformation. It doesn't matter what this actually looks like in equations, all that matters is that we can pictorially draw it like

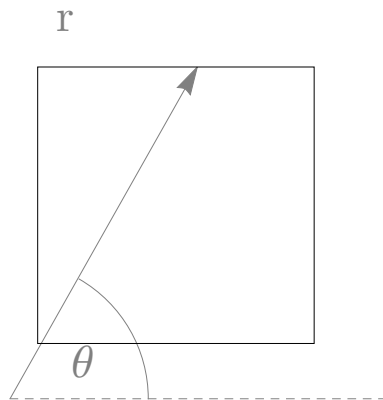
Out[111]=



And we see that in this set of coordinates, a square is still a square. What *is* different though is that the shape of our axes has changed. We no longer define the points on the square in terms of its position along two lines.

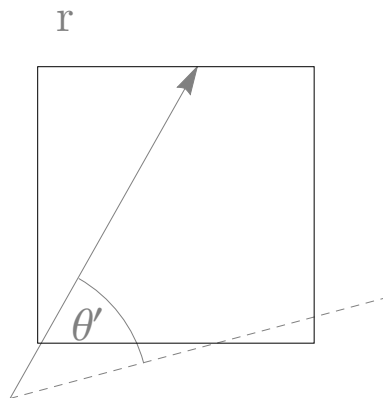
We can also imagine taking our square and shifting it. We can still describe it in this set of coordinates.

Out[110]=



Finally, we need to point out that θ here requires a reference axis, i.e. this would have been just as well described as

Out[120]=

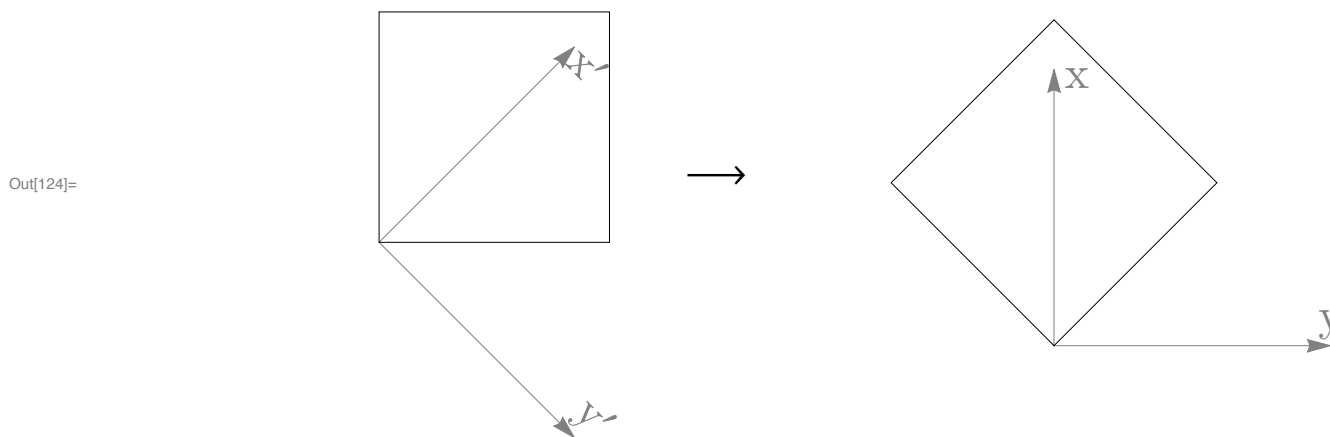


Categorizing Flavors Transformations

We've seen four types of coordinate transformations here.

The very first, where our axes simply rotated, was a *linear* transformation (in this case a rotation), that is, it's a transformation that doesn't change our linear axes into curvilinear ones.

Another way to put this, we can consider transforming our *square* instead of our axes, i.e. writing



and we see that, since this was a linear transformation, any thing that is a line in the original depiction remains a line in the transformed depiction.

The second was a *scaling* transformation, where we just scaled an axis. This is *also* a linear transformation.

The third, taking our axes and converting them into polar ones, is fundamentally a *non-linear* transformation. It takes our axes and rewrites them as non-linear ones.

Finally, we had a *translation* transformation, where we simply shifted our origin, and an embedding transformation, where we rotated that reference axis for θ . A translation transformation is in general *not* considered a linear transformation, since it cannot be written as a matrix operation. I'm happy to talk details on this if you want.

Relation To Molecular Cartesian Coordinates

We're gonna define our vibrational coordinates like

$$q_n = \sum_{i=1}^{3N} l_i^{(n)} \Delta x_i$$

where the Δx_i are defined so that we have a function that takes a set of Δx_i coordinates and can return a set of plain old Cartesians by

$$f(\Delta x_1, \Delta x_2, \dots, \Delta x_{3N}) = (x_{1,e} + \Delta x_1, x_{2,e} + \Delta x_2, \dots, x_{3N,e} + \Delta x_{3N})$$

we can write f (which converts displacement coordinates into regular Cartesian coordinates) like

$$f(\vec{\Delta x}) = \vec{\Delta x} + \vec{x}_e$$

alternately we can describe f^{-1} which converts from regular Cartesian coordinates to displacement coordinates like

$$f^{-1}(\vec{x}) = \vec{x} - \vec{x}_e$$

this is a different type of transformation than, say, the one that centers a molecule around its center of mass, which would look like

$$g(x_1, y_1, z_1, \dots, x_N, y_N, z_N) = (x_1 - x_{\text{COM}}, y_1 - y_{\text{COM}}, z_1 - z_{\text{COM}}, \dots)$$

Normal Modes

Why do we do this in the first place? Well for that we need a physical motivation, where we'll write the Hamiltonian for a harmonic potential in Cartesian coordinates as

$$H = \sum_{i=1}^{3N} -\frac{1}{2m_i} \frac{\partial^2}{\partial x_i^2} + \frac{1}{2} \sum_{i=1}^{3N} \sum_{j=1}^{3N} \frac{\partial^2 V(\vec{x}_e)}{\partial x_i \partial x_j} (x_i - x_{i,e}) (x_j - x_{j,e})$$

in displacement coordinates this looks like

$$H = \sum_{i=1}^{3N} -\frac{1}{2m_i} \frac{\partial^2}{\partial \Delta x_i^2} + \frac{1}{2} \sum_{i=1}^{3N} \sum_{j=1}^{3N} \frac{\partial^2 V(\vec{x}_e)}{\partial \Delta x_i \partial \Delta x_j} \Delta x_i \Delta x_j$$

we can do this because

$$\Delta x_i = x_i - x_{i,e}$$

so

$$\frac{\partial \Delta x_i}{\partial x_j} = \delta_{ij}$$

and the chain rule in 1D looks like

$$\frac{d}{dx} = \frac{dy}{dx} \frac{d}{dy}$$

Now that form of the Hamiltonian still sucks. We want

$$H = \sum_{i=1}^{3N-6} -\frac{1}{2m_i} \frac{\partial^2}{\partial q_i^2} + \frac{1}{2} \frac{\partial^2 V(\vec{x}_e)}{\partial q_i^2} q_i^2$$

Why? Because we know how to solve

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

and so if we can match up

$$\frac{\partial^2 V(\vec{x}_e)}{\partial q_i^2} = m_i \omega_i^2$$

then the world is good since we can solve any separable Hamiltonian with 0 work

$$H = h_1 + h_2 \Rightarrow \psi_n = \phi_1^{(n)} \phi_2^{(n)}$$

Basically, we want to find the linear transformation matrix L that will express each q_n like

$$q_n = \sum_{i=1}^{3N} \ell_i^{(n)} \Delta x_i$$

or as a matrix operation relating our entire set of normal modes Q to the entire vector of displacement coordinates ΔX

$$Q = L \Delta X$$

and once we have these q_n we know the solutions to our (harmonic) Hamiltonian without doing any extra work.

The idea will be to write

$$F = \frac{\partial^2 V(\vec{x}_e)}{\partial \Delta x_i \partial \Delta x_j}$$

and diagonalize this, taking into account the masses of the atoms, i.e. we want to make sure F in normal mode coordinates is separable & the kinetic energy in normal mode coordinates is separable.

This will also bring all of the mass contribution into the mode itself, so our derivatives will formally be expressed as

$$\frac{\partial^2 V(\vec{x}_e)}{\partial q_i^2} = \omega_i^2$$

and so the associated vibrational frequency of the normal modes will be the square roots of the eigenvalues of this diagonalization.

Going from NMs back to regular Cartesians

We have our old friend

$$f(\Delta x_1, \Delta x_2, \dots, \Delta x_{3N}) = (x_{1,e} + \Delta x_1, x_{2,e} + \Delta x_2, \dots, x_{3N,e} + \Delta x_{3N})$$

we can define a similar function for normal mode coordinates by first transforming from

$$f_{\text{NM}}(q_1, q_2, \dots, q_{3N}) = f\left(\sum_{n=1}^{3N} q_n \ell_1^{(n)} \Delta x_1, \sum_{n=1}^{3N} q_n \ell_2^{(n)} \Delta x_2, \dots, \sum_{n=1}^{3N} q_n \ell_{3N}^{(n)} \Delta x_{3N}\right)$$

this allows us to “scan” a normal mode coordinate, i.e. writing something like

$$f_{\text{NM}}(0, \dots, 0, \delta_i, 0, \dots, 0) = \dots$$

for multiple values of δ_i , giving us an animation of the normal mode vibration itself

Reaction Path Idea

When we express our Hamiltonian at a point other than the equilibrium we have

$$\begin{aligned} H &= \sum_{i=1}^{3N} -\frac{1}{2m_i} \frac{\partial^2}{\partial x_i^2} + \frac{1}{2} \sum_{i=1}^{3N} \frac{\partial V(\vec{x}_e)}{\partial x_i} (x_i - x_{i,e}) + \frac{1}{2} \sum_{i=1}^{3N} \sum_{j=1}^{3N} \frac{\partial^2 V(\vec{x}_e)}{\partial x_i \partial x_j} (x_i - x_{i,e}) (x_j - x_{j,e}) \\ &= -\sum_{i=1}^{3N} \frac{1}{2m_i} \frac{\partial^2}{\partial \Delta x_i^2} + \sum_{i=1}^{3N} \frac{\partial V(\vec{x}_e)}{\partial \Delta x_i} \Delta x_i + \frac{1}{2} \sum_{i=1}^{3N} \sum_{j=1}^{3N} \frac{\partial^2 V(\vec{x}_e)}{\partial \Delta x_i \partial \Delta x_j} \Delta x_i \Delta x_j \end{aligned}$$

and the goal is to find a *subspace* of normal modes in which we can write

$$H = -\sum_{i=1}^{3N-7} \frac{1}{2m_i} \frac{\partial^2}{\partial q_i^2} + \frac{1}{2} \frac{\partial^2 V(\vec{x}_e)}{\partial q_i^2} q_i^2$$

i.e. we want to find a set of normal mode coordinates that don't care that the gradient is non-zero.

The idea here will be to note that in $3N$ dimensional space, the identity operator can be written as

$$I = \sum_{i=1}^{3N} P_i$$

where P_i is a function that returns the projection onto the i^{th} coordinate, e.g. if we have a set of Cartesian coordinates $(x_1, x_2, \dots, x_{3N})$, then P_i just returns the set of coordinates $(0, \dots, 0, x_i, 0, \dots, 0)$.

So now if we want to remove a set of coordinates $\{s_k\}$ from our calculation, we can just define a projection operator that removes this as

$$P_S = I - \sum_k P_{s_k}$$

This will be the key to our approach. By defining a coordinate that includes the entirety of the contribution to the potential from the gradient, we can “project this out” and be working in our desired subspace.

From that, it should be relatively unsurprising that the trick will be to define our coordinate as being the gradient itself (once we normalize). We call this the reaction path mode. An important thing to recognize is that the projection operator onto this mode, P_{RP} , can be expressed in terms of displacement coordinates in the same way that we related normal modes to a linear combination displacement coordinates.

At the end of the day, then, we can write a version of

$$F = \frac{\partial^2 V(\vec{x}_e)}{\partial \Delta x_i \partial \Delta x_j}$$

where we've factored out the gradient coordinate and all will work well. The upshot of this is that we've removed a coordinate, so we end up with $3N-7$ normal modes.