Z-Matrix Derivatives

Z-Matrix to Cartesian

We do so much work with Z-matrices and Cartesians that we should really have an analytic representation of the Jacobian. So let's get on that. We'll start by going from Z-matrix to Cartesian. To do that we note that we need 3 pieces of info

- The origin x_0, can be assumed to be (0, 0, 0) by default
- The axis system $(e_1, \ e_2, \ e_3)$, can be assumed to be I_3 (well really e_3 actually isn't necessary)
- The Z-matrix

For every atom, our Z-matrix tells us 6 things

- What are the reference positions, x_{i-1}, x_{i-2}, x_{i-3}
- What are the reference values, $z_{i,1}$, $z_{i,2}$, $z_{i,3}$

Derivation

We define our coordinates recursively as

$$v_i = x_{i,b} - x_{i,a} \quad u_i = x_{i,c} - x_{i,a} \quad n_i = v_i \times u_i$$
$$x_i = x_{i,a} + R\left(z_{i,3}, \ v_i\right) \cdot R\left(z_{i,2}, \ n_i\right) \cdot z_{i,1} \ \hat{v}_i$$

where $R(\theta, v)$ is a rotation of θ radians about axis $\hat{v}$, given by

$$\begin{aligned} R(\theta, \ \hat{v}) &= \begin{pmatrix} \cos\theta & -v_z \sin\theta & v_y \sin\theta \\ v_z \sin\theta & \cos\theta & -v_x \sin\theta \\ -v_y \sin\theta & v_x \sin\theta & \cos\theta \end{pmatrix} + \begin{pmatrix} {v_x}^2 & v_x \, v_y & v_x \, v_z \\ v_x \, v_y & {v_y}^2 & v_y \, v_z \\ v_x \, v_z & v_y \, v_z & {v_z}^2 \end{pmatrix} (1-\cos\theta) \\ &= \begin{pmatrix} {v_x}^2 & v_x \, v_y & v_x \, v_z \\ v_x \, v_y & {v_y}^2 & v_y \, v_z \\ v_x \, v_z & v_y \, v_z & {v_z}^2 \end{pmatrix} + \begin{pmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{pmatrix} \sin\theta + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cos\theta + \\ &= \hat{v} \otimes \hat{v} + (\mathbb{I}_3 - \hat{v} \otimes \hat{v}) \cos(\theta) - \epsilon_3 \, \hat{v} \sin(\theta) \end{aligned}$$

which I'll agree is not the most intuitive thing in the world, but maybe is a bit nice to think about when you know that it comes from the Rodrigues rotation formula

$$R(\theta, \ k).v \ = \ \cos\theta \ v \ + \ \sin\theta \ (k \times v) \ + \ (k \cdot v) \ (1-\cos\theta) \ k$$

I think that's definitely got a clearer physical meaning you can work with.

In any case, this means we get

$$\frac{dx_i}{dq}=\frac{dx_{i-1}}{dq}+\frac{d}{dq}\left(z_{i,1}\,R\left(z_{i,3},\ v_i\right)\cdot R\left(z_{i,2},\ n_i\right)\cdot\hat{v}_i\right)$$

where q is one of our z_{nm} components.

It's worth noting that we have to compute this recursively, i.e. we need the $i-1$ result before we can get the i result. When implementing, keep in mind that the parts of rotation derivatives can be computed without knowing the $i-1$ results, though.

First off, for chain-rule reasons we'll condense the rotation matrices into a single rotation

$$\Theta=R\left(z_{i,3},\ v_i\right)\cdot R\left(z_{i,2},\ n_i\right)$$

which means that we get

$$\begin{aligned}\frac{dx_i}{dq}&=\frac{dx_{i-1}}{dq}+\left(\frac{d}{dq}\,\Theta\right)\cdot\left(z_{i,1}\,\hat{v}_i\right)+\Theta\cdot\frac{d}{dq}\left(z_{i,1}\,\hat{v}_i\right)\\&=\frac{dx_{i-1}}{dq}+\left(\frac{d}{dq}\,\Theta\right)\cdot\left(z_{i,1}\,\hat{v}_i\right)+\Theta\cdot\left(\frac{dz_{i,1}\,\hat{v}_i}{dq}\right)\end{aligned}$$

then since q is scalar, the product rule applies as normal so we get

$$\frac{d}{dq}\,\Theta=\left(\frac{d}{dq}\,R\left(z_{i,3},\ v_i\right)\right)\cdot R\left(z_{i,2},\ n_i\right)+R\left(z_{i,3},\ v_i\right)\cdot\frac{d}{dq}\,R\left(z_{i,2},\ n_i\right)$$

next we'll look at one of these rotations on its own, like

$$\begin{aligned}\frac{d}{dq}\,R(\theta,\ v)&=\frac{d}{dq}\,(v\otimes v)+\frac{d}{dq}\,(\mathbb{I}_3-v\otimes v)\cos(\theta)-\frac{d}{dq}\,\epsilon_3\,v\sin(\theta)\\&=v'\otimes v+v\otimes v'+\frac{d(\mathbb{I}_3-v\otimes v)}{dq}\cos(\theta)-(\mathbb{I}_3-v\otimes v)\sin(\theta)\,\frac{d\theta}{dq}-\epsilon_3\left(\frac{dv}{dq}\sin(\theta)+v\cos(\theta)\,\frac{d\theta}{dq}\right)\\&=v'\otimes v+v\otimes v'-(v'\otimes v+v\otimes v')\cos(\theta)-(\mathbb{I}_3-v\otimes v)\sin(\theta)\,\theta'-\epsilon_3(v'\sin(\theta)+v\cos(\theta)\,\theta')\end{aligned}$$

I'm sure this looks nasty, but most of this is actually pretty straightforward. To justify this claim, let's note that either $v=v_i$ or $v=n_i$, $\theta=z_{i,2}$ or $\theta=z_{i,3}$, and

$$\begin{aligned}\frac{d}{dq}\,\hat{a}&=\frac{1}{|a|}\,\frac{da}{dq}-\frac{a}{|a|^2}\,\frac{d}{dq}\,\sqrt{a\cdot a}\\&=\frac{1}{|a|}\,\frac{da}{dq}-\frac{a}{|a|^3}\,\frac{da}{dq}\cdot a\\&=\frac{1}{|a|}\left(\frac{da}{dq}-\hat{a}\,\frac{da}{dq}\cdot\hat{a}\right)\\&=\frac{1}{|a|}\,(\mathbb{I}_3-\hat{a}\otimes\hat{a})\,\frac{da}{dq}\end{aligned}$$

$$\frac{d}{dq}\ u_i = \frac{dx_{i,b}}{dq} - \frac{dx_{i,a}}{dq}$$
$$\frac{d}{dq}\ u_i = \frac{d}{dq}\ \left(x_{i,c} - x_{i,a}\right) = \frac{dx_{i,c}}{dq} - \frac{dx_{i,a}}{dq}$$
$$\frac{d}{dq}\ n_i = \frac{d}{dq}\ v_i \times u_i = \frac{d}{dq}\ v_i \times u_i + v_i \times \frac{d}{dq}\ u_i$$
$$\frac{dz_{i,2}}{dq} = \delta_{q z_{i,2}}$$
$$\frac{dz_{i,3}}{dq} = \delta_{q z_{i,3}}$$

and we've got $dx_{i,a}/dq$, $dx_{i,b}/dq$ and $dx_{i,c}/dq$ from prior steps.

I'm not gonna write this out in full, though, since that's just asking for me to make a mistake.

Finally, let's think about the case that we have

$$\frac{dx_i}{dz_{nm}} = \frac{dx_{i,a}}{dz_{nm}} + \left(\frac{d}{dz_{nm}}\ \Theta\right) \cdot \hat{v}_i + \Theta \cdot \frac{d}{dz_{nm}}\ \hat{v}_i$$

we'll note that we have

$$\frac{dz_{ij}}{dz_{nm}} = 0$$

but recursively, we can still have

$$\frac{dx_{i,a}}{dz_{nm}} \neq 0$$

A Note on Formatting

A way I like to think about this is in terms of blocks matrices, where

$$\nabla_Z X = \begin{pmatrix} \nabla_{Z_1} x_0 & \nabla_{Z_1} x_1 & \nabla_{Z_1} x_2 & \nabla_{Z_1} x_3 \\ \nabla_{Z_2} x_0 & \nabla_{Z_2} x_1 & \nabla_{Z_2} x_2 & \nabla_{Z_2} x_3 \\ \nabla_{Z_3} x_0 & \nabla_{Z_3} x_1 & \nabla_{Z_3} x_2 & \nabla_{Z_3} x_3 \end{pmatrix}$$

Implementation

Higher Derivatives

These will be a huge pain to do correctly, but we'll give it a shot...someday. For the most part we shouldn't need beyond second derivatives, but doing the derivatives analytically can really help with numerical stability.

$$\frac{\partial^2 x_i}{\partial q_m\,\partial q_n} = \frac{\partial}{\partial q_m}\left(\frac{d x_{i-1}}{d q_n} + \left(\frac{d}{d q_n}\,\Theta\right)\cdot\left(z_{i,1}\,\hat{v}_i\right) + \Theta\cdot\left(\frac{d z_{i,1}\,\hat{v}_i}{d q_n}\right)\right)$$

$$= \frac{\partial x_{i-1}}{\partial q_m\,\partial q_n} + \frac{\partial}{\partial q_m}\left(\left(\frac{d}{d q_n}\,\Theta\right)\cdot\left(z_{i,1}\,\hat{v}_i\right)\right) + \frac{\partial}{\partial q_m}\left(\Theta\cdot\left(\frac{d z_{i,1}\,\hat{v}_i}{d q_n}\right)\right)$$

$$= \frac{\partial^2\,x_{i-1}}{\partial q_m\,\partial q_n} + \left(\frac{\partial^2\Theta}{\partial q_m\,\partial q_n}\right)\cdot\left(z_{i,1}\,\hat{v}_i\right) + \left(\frac{\partial\Theta}{\partial q_n}\right)\cdot\frac{\partial\left(z_{i,1}\,\hat{v}_i\right)}{\partial q_m} + \frac{\partial\Theta}{\partial q_m}\cdot\left(\frac{\partial z_{i,1}\,\hat{v}_i}{\partial q_n}\right) + \Theta\cdot\frac{\partial^2\left(z_{i,1}\,\hat{v}_i\right)}{\partial q_m\,\partial q_n}$$

which means now that we need two new classes of derivatives

$$\frac{\partial^2\Theta}{\partial q_m\,\partial q_n} = \frac{\partial^2}{\partial q_m\,\partial q_n}\left(R\left(z_{i,3},\ v_i\right)\cdot R\left(z_{i,2},\ n_i\right)\right)$$

$$\frac{\partial^2\left(z_{i,1}\,\hat{v}_i\right)}{\partial q_m\,\partial q_n} = \frac{\partial^2 z_{i,1}}{\partial q_m\,\partial q_n}\,\hat{v}_i + \frac{\partial z_{i,1}}{\partial q_n}\,\frac{\partial \hat{v}_i}{\partial q_m} + \frac{\partial z_{i,1}}{\partial q_m}\,\frac{\partial \hat{v}_i}{\partial q_n} + z_{i,1}\,\frac{\partial^2 \hat{v}_i}{\partial q_m\,\partial q_n}$$

$$= \frac{\partial z_{i,1}}{\partial q_n}\,\frac{\partial \hat{v}_i}{\partial q_m} + \frac{\partial z_{i,1}}{\partial q_m}\,\frac{\partial \hat{v}_i}{\partial q_n} + z_{i,1}\,\frac{\partial^2 \hat{v}_i}{\partial q_m\,\partial q_n}$$

and from these we have

$$\frac{\partial^2}{\partial q_n\,\partial q_m}\,R(\theta,\ v) = \frac{\partial^2}{\partial q_n\,\partial q_m}\,(v\otimes v) + \frac{\partial^2}{\partial q_n\,\partial q_m}\,(\mathbb{I}_3 - v\otimes v)\,\cos(\theta) - \frac{\partial^2}{\partial q_n\,\partial q_m}\,\epsilon_3\,v\sin(\theta)$$

and this is just going to be a bunch of product rule shit, so let’s look at what we’ll need for this

$$v_i = x_{i,b} - x_{i,a}\quad u_i = x_{i,c} - x_{i,b}\quad n_i = v_i\times u_i$$

$$x_i = x_{i,a} + R\left(z_{i,3},\ v_i\right)\cdot R\left(z_{i,2},\ n_i\right)\cdot z_{i,1}\,\hat{v}_i$$

$$\frac{\partial^2 v_i}{\partial q_n\,\partial q_m} = \frac{\partial^2 x_{i,b}}{\partial q_n\,\partial q_m} - \frac{\partial^2 x_{i,a}}{\partial q_n\,\partial q_m}$$

$$\frac{\partial^2 u_i}{\partial q_n\,\partial q_m} = \frac{\partial^2 x_{i,c}}{\partial q_n\,\partial q_m} - \frac{\partial^2 x_{i,b}}{\partial q_n\,\partial q_m}$$

$$\frac{\partial^2 n_i}{\partial q_n\,\partial q_m} = \frac{\partial^2 n_i}{\partial q_n\,\partial q_m}\,(v_i\times u_i)$$

$$\frac{\partial^2 \sin(\theta)}{\partial q_n\,\partial q_m} = \frac{\partial}{\partial q_n}\,\frac{\partial\theta}{\partial q_m}\,\cos(\theta)$$

$$= \frac{\partial^2\theta}{\partial q_n\,\partial q_m}\,\cos(\theta) - \frac{\partial\theta}{\partial q_m}\,\frac{\partial\theta}{\partial q_n}\,\sin(\theta)$$

$$= -\sin(\theta)\begin{cases} 1 & \theta \text{ is } q_m \text{ is } q_n \\ 0 & \text{else}\end{cases}$$

$$\frac{\partial^2 \cos(\theta)}{\partial q_n\,\partial q_m} = -\cos(\theta)\begin{cases} 1 & \theta \text{ is } q_m \text{ is } q_n \\ 0 & \text{else}\end{cases}$$

$$
\begin{aligned}
\frac{\partial^2 \hat{a}}{\partial q_m \partial q_n} &= \frac{\partial}{\partial q_m}\left(\frac{1}{|a|}\,(\mathbb{I}_3 - \hat{a} \otimes \hat{a})\,\frac{\partial a}{\partial q_n}\right) \\
\frac{\partial}{\partial q_m}\frac{1}{|a|} &= -\frac{1}{|a|^2}\,\frac{\partial}{\partial q_m}|a| \\
&= -\frac{1}{|a|^2}\,\frac{\partial}{\partial q_m}\sqrt{a \cdot a} \\
&= -\frac{1}{2}\,\frac{1}{|a|^3}\,\frac{\partial}{\partial q_m}\,a \cdot a \\
&= -\frac{1}{|a|^3}\,\frac{\partial a}{\partial q_m} \cdot a
\end{aligned}
$$

Cartesian to Z-Matrix

For this we only have three types of things to calculate

- Bond distances
- Bond angles
- Dihedral angles

so if we can get expressions for each of these we're golden. Unfortunately the expressions are still super annoying to work with.

In this case we have

$$
\begin{aligned}
r_{ij} &= |a| && \text{where } a = x_j - x_i \\
\theta_{ijk} &= \tan^{-1}\left(\frac{|a \times b|}{|a|\,|b|},\, \frac{a \cdot b}{|a|\,|b|}\right) && \text{where } a = x_j - x_i \text{ and } b = x_k - x_i \\
\tau_{ijkl} &= \tan^{-1}\left(\left(\hat{n}_1 \times \hat{b}\right) \cdot \hat{n}_2,\, \hat{n}_1 \cdot \hat{n}_2\right) && \text{where } a = x_j - x_i,\ b = x_k - x_j,\ c = x_l - x_k \\
& && \hat{n}_1 = \frac{a \times b}{|a \times b|},\ \hat{n}_2 = \frac{b \times c}{|b \times c|},\ \hat{b} = \frac{b}{|b|}
\end{aligned}
$$

worth noting, also, that sometimes we have

$$
\theta_{ijk} = \tan^{-1}\left(\operatorname{sgn}((a \times b) \cdot z)\,\frac{|a \times b|}{|a|\,|b|},\, \frac{a \cdot b}{|a|\,|b|}\right)
$$

where z is a vector that defines "up" and therefore allows us to provide a consistent signed angle, rather than being restricted to the $[0, \pi]$ range. Dihedrals get this for free since 4 points are able to be non-coplanar and therefore handed-ness is implicit in the definition.

First Derivatives

Case 1: r

$$r_{ij}=|a| \text{ where } a=x_j-x_i$$

so

$$\frac{\partial r_{ij}}{\partial x_m}=\frac{\partial}{\partial x_m} a \cdot \nabla_a |a|$$

which we've mostly worked out lower down, but we'll note

$$\frac{\partial a}{\partial x_m}=\left(\mathbb{I}_3\, \delta_{jm} - \mathbb{I}_3\, \delta_{im}\right)$$

so

$$\frac{\partial r_{ij}}{\partial x_m}=\left(\nabla_a |a|\, \delta_{jm} - \nabla_a |a|\, \delta_{im}\right)$$

Case 2: θ

This term is mostly treated below, but we'll do the chain-rule part here

$$\theta_{ijk}=\tan^{-1}\left(\frac{|a\times b|}{|a|\,|b|}, \frac{a\cdot b}{|a|\,|b|}\right) \text{ where } a=x_j-x_i \text{ and } b=x_k-x_i$$

this is just

$$\theta_{ijk}=\tan^{-1}(\sin(a,\ b),\ \cos(a,\ b))$$

so we can just apply the chain rule to get

$$\frac{\partial \theta_{ijk}}{\partial x_m}=\frac{\partial}{\partial x_m}(a\ \ b) \cdot \nabla_{(a\ \ b)} \tan^{-1}(\sin(a,\ b),\ \cos(a,\ b))$$

and that first term is just some Kronecker deltas (we work it out later). The second term is expanded below.

Case 3: τ

We start with

$$\tau_{ijkl}=\tan^{-1}\left(\left(\hat{b}\times\hat{n}_1\right)\cdot\hat{n}_2,\ \hat{n}_1\cdot\hat{n}_2\right) \text{ where } a=x_j-x_i,\ b=x_k-x_j,\ c=x_l-x_k$$

$$\hat{n}_1=\frac{a\times b}{|a\times b|},\ \hat{n}_2=\frac{b\times c}{|b\times c|},\ \hat{b}=\frac{b}{|b|}$$

then we simplify that first part, making use of some cross/dot product rules

$$\left(\hat{b}\times\hat{n}_1\right)\cdot\hat{n}_2=\hat{n}_2\cdot\left(\hat{b}\times\hat{n}_1\right)$$

$$=\hat{b}\cdot(\hat{n}_1\times\hat{n}_2)$$

Now we make use of triple-product rules to get

$$\begin{aligned}\hat{n}_1\times\hat{n}_2&=\frac{\hat{n}_1\times(b\times c)}{|b\times c|}\\&=\frac{1}{|b\times c|}\left((\hat{n}_1\cdot c)\ b-c(\hat{n}_1\cdot b)\right)\\&=\frac{(\hat{n}_1\cdot c)}{|b\times c|}\ b\end{aligned}$$

and therefore $\hat{n}_1\times\hat{n}_2$ is in the direction of b (or 180° perpendicular to it)

This is dope, since it means

$$\begin{aligned}\left(\hat{b}\times\hat{n}_1\right)\cdot\hat{n}_2&=\hat{b}\cdot(\hat{n}_1\times\hat{n}_2)\\&=\operatorname{sgn}\left(\hat{b}\cdot(\hat{n}_1\times\hat{n}_2)\right)*|\hat{n}_1\times\hat{n}_2|\\&=\operatorname{sgn}\left(\hat{b}\cdot(\hat{n}_1\times\hat{n}_2)\right)*\frac{|n_1\times n_2|}{|n_1|\,|n_2|}\\&=\operatorname{sgn}\left(\hat{b}\cdot(\hat{n}_1\times\hat{n}_2)\right)*\sin(n_1,\ n_2)\end{aligned}$$

the *sgn* term is annoying, but is basically asking if a, b, and c form right-handed coordinate system or not. We will note that this function has derivatives given by

$$\operatorname{sgn}(x)=\begin{cases}0 & x\neq0\\ \text{undefined} & x=0\end{cases}$$

but when $x=0$, $\sin(n_1,\ n_2)=0$, so we still get well-behaved derivatives. This also means that in the chain rule it will always pull out/do nothing

Finally, therefore,

$$\begin{aligned}\tau_{ijkl}&=\tan^{-1}\left(\left(\hat{b}\times\hat{n}_1\right)\cdot\hat{n}_2,\ \hat{n}_1\cdot\hat{n}_2\right)\\&=\operatorname{sgn}\left(\hat{b}\cdot(\hat{n}_1\times\hat{n}_2)\right)\tan^{-1}(\sin(n_1,\ n_2),\ \cos(n_1,\ n_2))\end{aligned}$$

and so

$$\frac{\partial}{\partial x_n}\tau_{ijkl}=\operatorname{sgn}\left(\hat{b}\cdot(\hat{n}_1\times\hat{n}_2)\right)\frac{\partial}{\partial x_n}(a\ \ b\ \ c)\cdot\nabla_{(a\ \ b\ \ c)}(n_1\ \ n_2)\cdot\nabla_{(n_1\ \ n_2)}\tan^{-1}(\sin(n_1,\ n_2),\ \cos(n_1,\ n_2))$$

And so that last term is just built off of prior stuff. The first term isn't bad either. The second term requires a bit of thought. The total term is

$$\nabla_{(a\ \ b\ \ c)}(n_1\ \ n_2)=\begin{pmatrix} \frac{\partial n_1}{\partial a} & \frac{\partial n_2}{\partial a} \\ \frac{\partial n_1}{\partial b} & \frac{\partial n_2}{\partial b} \\ \frac{\partial n_1}{\partial c} & \frac{\partial n_2}{\partial c} \end{pmatrix}$$

which is basically a matrix of matrices (i.e. a 4D tensor). Each of those elements can then be tabulated like

$$\begin{array}{ll} \frac{\partial n_1}{\partial a}=\frac{\partial\, a\times b}{\partial a}=\mathbb{I}_3\times b & \frac{\partial n_2}{\partial a}=\frac{\partial\, a\times b}{\partial a}=\emptyset \\ \frac{\partial n_1}{\partial b}=a\times\mathbb{I}_3 & \frac{\partial n_2}{\partial b}=\mathbb{I}_3\times c \\ \frac{\partial n_1}{\partial c}=\emptyset & \frac{\partial n_2}{\partial c}=b\times\mathbb{I}_3 \end{array}$$

Next we consider

$$\begin{aligned} \frac{\partial}{\partial x_n}(a\ \ b\ \ c) &= \begin{pmatrix} \frac{\partial a}{\partial x_n} & \frac{\partial b}{\partial x_n} & \frac{\partial c}{\partial x_n} \end{pmatrix} \\ &= \begin{pmatrix} \frac{\partial\left(x_j-x_i\right)}{\partial x_n} & \frac{\partial\left(x_k-x_j\right)}{\partial x_n} & \frac{\partial(x_l-x_k)}{\partial x_n} \end{pmatrix} \\ &= \left(\mathbb{I}_3\left(\delta_{nj}-\delta_{ni}\right)\ \ \mathbb{I}_3\left(\delta_{nk}-\delta_{nj}\right)\ \ \mathbb{I}_3(\delta_{nl}-\delta_{nk})\right) \end{aligned}$$

and so

$$\begin{aligned} \frac{\partial}{\partial x_n}\tau_{ijkl} &= \frac{\partial}{\partial x_n}(a\ \ b\ \ c)\cdot\nabla_{(a\ \ b\ \ c)}(n_1\ \ n_2)\cdot\nabla_{(n_1\ \ n_2)}\tan^{-1}(\sin(n_1,\ n_2),\ \cos(n_1,\ n_2)) \\ &= \frac{\partial}{\partial x_n}(n_1\ \ n_2)\,\nabla_{(n_1\ \ n_2)}\tan^{-1}(\sin(n_1,\ n_2),\ \cos(n_1,\ n_2)) \end{aligned}$$

where

$$\begin{aligned} \frac{\partial}{\partial x_m}(n_1\ \ n_2) &= \frac{\partial}{\partial x_m}(a\ \ b\ \ c)\cdot\nabla_{(a\ \ b\ \ c)}(n_1\ \ n_2) \\ &= \left((\mathbb{I}_3\times b)\left(\delta_{mj}-\delta_{mi}\right)+(a\times\mathbb{I}_3)\left(\delta_{mk}-\delta_{mj}\right)\ \ (\mathbb{I}_3\times c)\left(\delta_{mk}-\delta_{mj}\right)+(b\times\mathbb{I}_3)(\delta_{ml}-\delta_{mk})\right) \end{aligned}$$

so

$$\frac{\partial}{\partial x_m}(n_1\ \ n_2)\,\nabla_{(n_1\ \ n_2)}\theta(n_1,\ n_2)$$

$$\mathbb{I}_3\times b)\cdot\frac{\partial\theta}{\partial n_1}\ \delta_{mi}+\left(((\mathbb{I}_3\times b)-(a\times\mathbb{I}_3))\cdot\frac{\partial\theta}{\partial n_1}-(\mathbb{I}_3\times c)\cdot\frac{\partial\theta}{\partial n_2}\right)\delta_{mj}+\left((a\times\mathbb{I}_3)\cdot\frac{\partial\theta}{\partial n_1}+((\mathbb{I}_3\times c)-(b\times\mathbb{I}_3))\cdot\frac{\partial\theta}{\partial n_2}\right)\delta_{mk}+(b\times$$

$$b)\cdot\frac{\partial\theta}{\partial n_1}\ \delta_{mi}+\left((\epsilon_3 c)\cdot\frac{\partial\theta}{\partial n_2}-(\epsilon_3 a+\epsilon_3 b)\cdot\frac{\partial\theta}{\partial n_1}\right)\delta_{mj}+\left((\epsilon_3\ a)\cdot\frac{\partial\theta}{\partial n_1}-(\epsilon_3 b+\epsilon_3 c)\cdot\frac{\partial\theta}{\partial n_2}\right)\delta_{mk}+(\epsilon_3 b)\cdot\frac{\partial\theta}{\partial n_2}\ \delta_{ml}$$

$$\frac{\partial \theta}{\partial n_1}\ \delta_{mi} - \left(c\times\frac{\partial \theta}{\partial n_2} - (a+b)\times\frac{\partial \theta}{\partial n_1}\right)\delta_{mj} + \left(a\times\frac{\partial \theta}{\partial n_1} - (b+c)\times\frac{\partial \theta}{\partial n_2}\right)\delta_{mk} + b\times\frac{\partial \theta}{\partial n_2}\ \delta_{ml}$$

we can now treat the angle derivatives as we have before

$$\frac{\partial \theta}{\partial n_1} = -\frac{n_1 \perp n_2}{|n_1|}$$
$$\frac{\partial \theta}{\partial n_2} = -\frac{n_2 \perp n_1}{|n_2|}$$

where $n_1 \perp n_2$ is the unit vector perpendicular to n_1 pointing to n_2, i.e.

$$n_1 \perp n_2 = \frac{n_2 - (n_2 \cdot n_1)\ n_1}{|n_2 - (n_2 \cdot n_1)\ n_1|}$$

but it’s worth noting that

In[374]:=

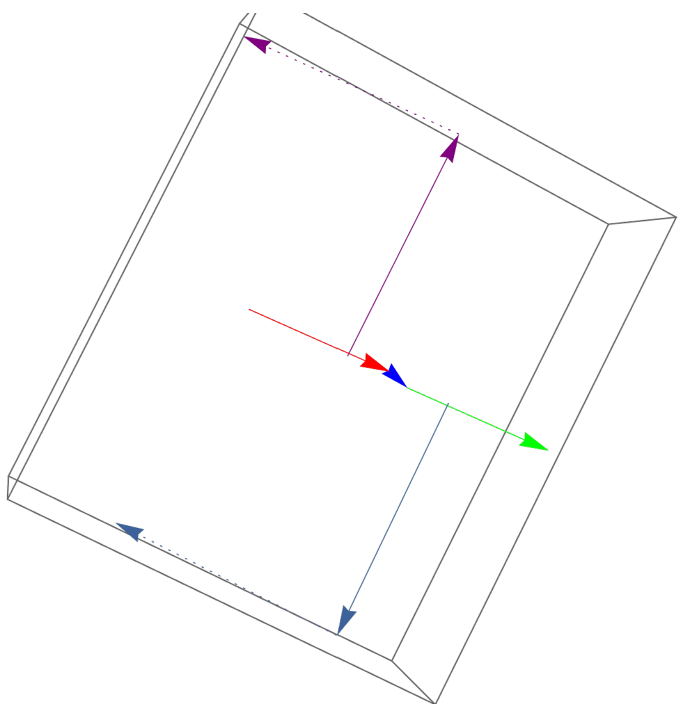
```
plotDihedralAngleDerivs[coords_, i_, j_, k_, l_] :=
 Block[{a, b, c, n, m, dn, dm},
  a = coords⟦j⟧ - coords⟦i⟧;
  b = coords⟦k⟧ - coords⟦j⟧;
  c = coords⟦l⟧ - coords⟦k⟧;
  n = Normalize@Cross[a, b];
  m = Normalize@ Cross[b, c];
  {dn, dm} = testAngleDerivComps[n, m];
  Graphics3D[{
    {
     Red,
     Arrow[{coords⟦i⟧, coords⟦j⟧}],
     Blue,
     Arrow[{coords⟦j⟧, coords⟦k⟧}],
     Green,
     Arrow[{coords⟦k⟧, coords⟦l⟧}]
    },
    {
     Purple,
     Arrow[{
       Mean@coords⟦{i, j, k}⟧,
       Mean@coords⟦{i, j, k}⟧ + n
      }],
     Dotted,
     Arrow[{
       Mean@coords⟦{i, j, k}⟧ + n,
       Mean@coords⟦{i, j, k}⟧ + n + dn
      }]
    },
```

```
      {
       Hue[.6, .6, .6],
       Arrow[{
         Mean@coords⟦{j, k, l}⟧,
         Mean@coords⟦{j, k, l}⟧ + m
        }],
       Dotted,
       Arrow[{
         Mean@coords⟦{j, k, l}⟧ + m,
         Mean@coords⟦{j, k, l}⟧ + m + dm
        }]
      }
     }
    ]
   ]
```

In[383]:=

```
plotDihedralAngleDerivs[{
   {0, 0, .00},
   {.5, .7, .00},
   {.7, .2, .00},
   {1.2, .7, -.01}
  },
  1, 2, 3, 4
 ]
```

Out[383]=

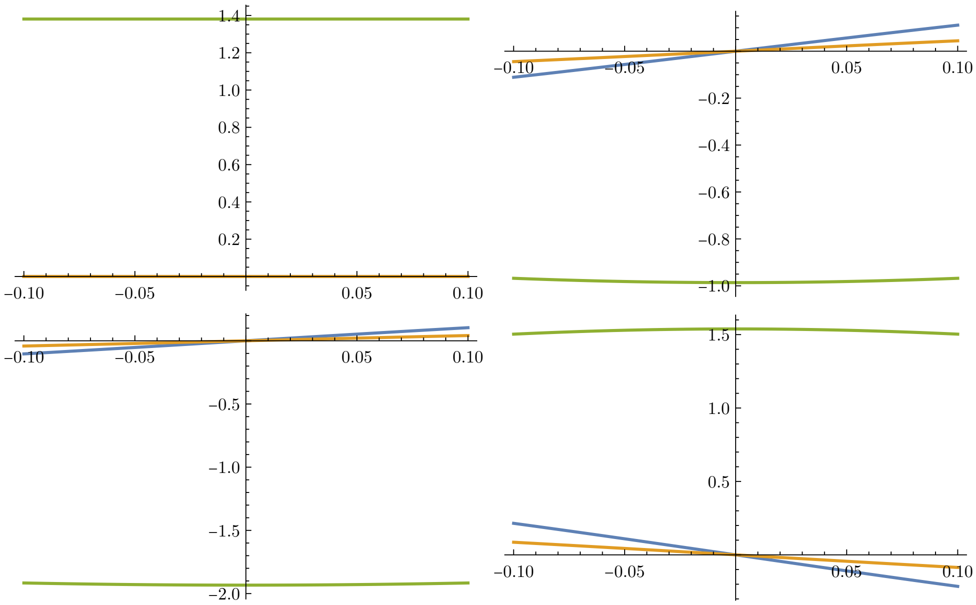


```
plotVectorAngleDerivs[Cross[a]
```

Limit of Parallel Vectors

There's one important limiting case to consider where $n_1 = kn_2$. We can see from finite difference tests that this will actually work

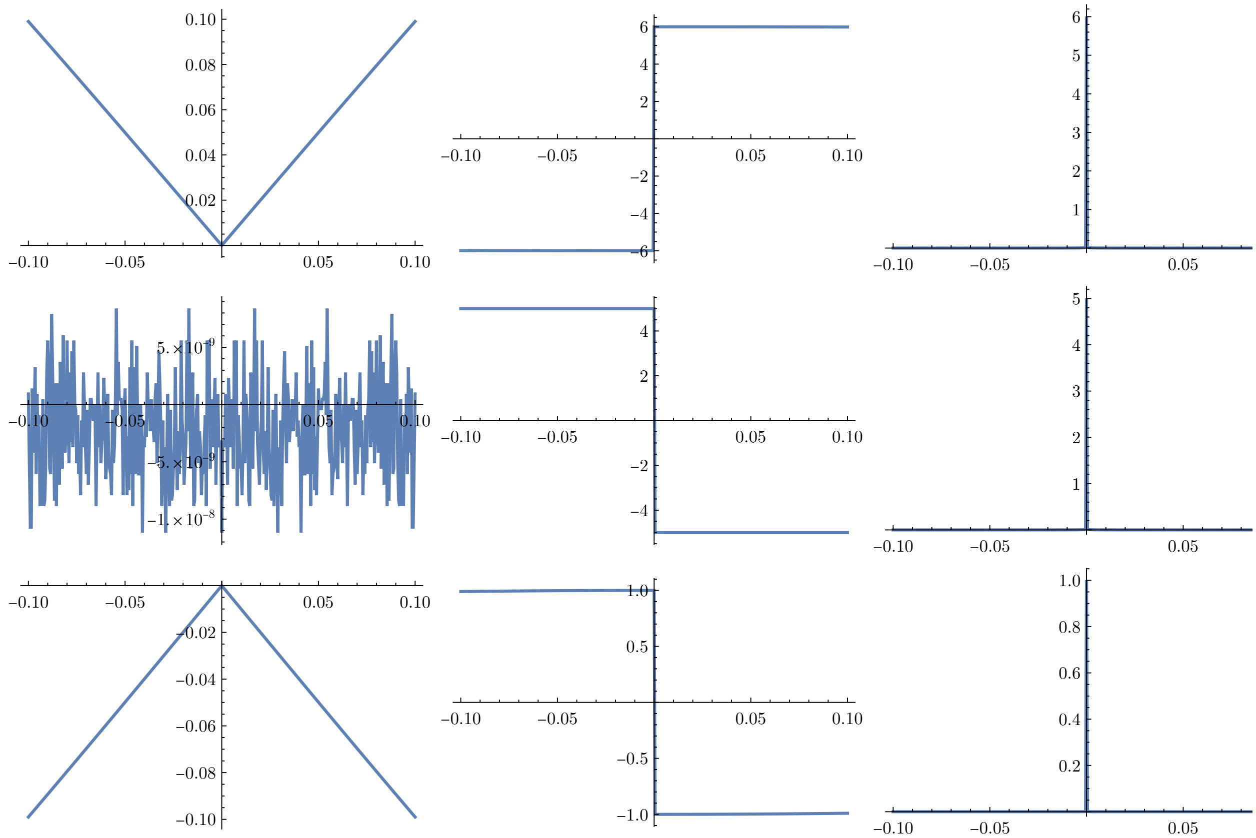
Out[736]=



where these components go smoothly through 0 iff 0 is explicitly excluded as a point.

Note that this doesn't apply to the angle between any two vectors

Out[738]=



there are some clear discontinuities and turning points here. That means the appropriate limiting behavior is really coming from something about how the dihedral angle is defined...or from the fact that it takes care of any sign issues by definition? But the issue is that I'm finding that these derivatives should all be 0 at 0...

In[403]:=

```
dihedScan = DeleteCases[#, 0.] &@Range[-.005, .005, .001];
```

In[406]:=

```
getDihedFD[
 {
  {0, 0, 0},
  {.5, .7, 0},
  {.7, .2, 0},
  {1.2, .7, 0.}
 },
 1,
 2,
 3,
 4
]
```

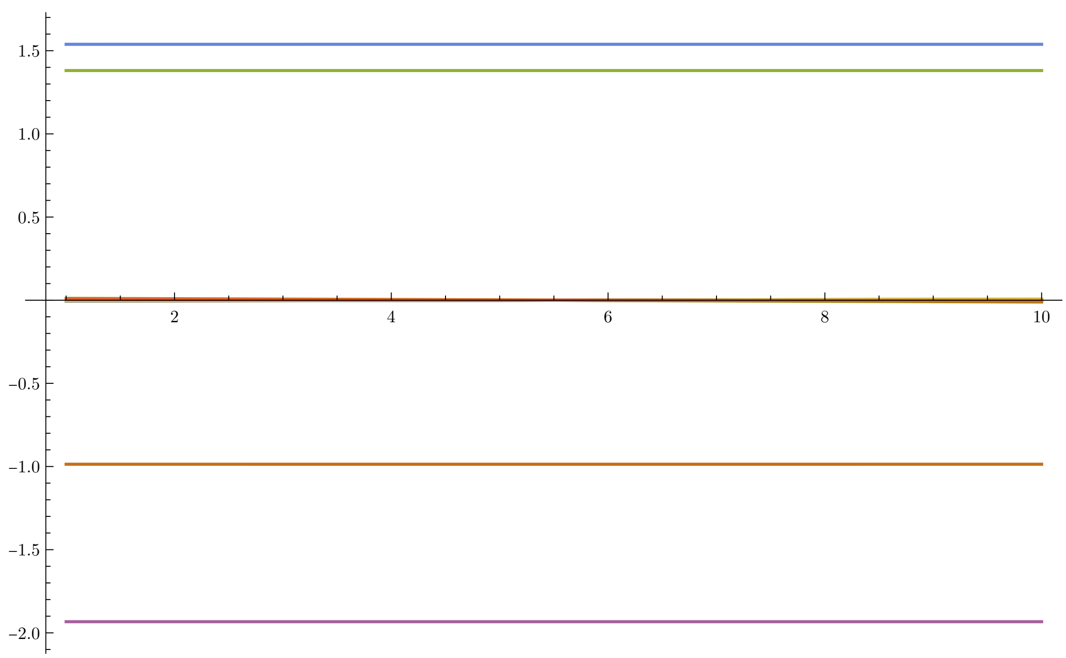
Out[406]=

$\{-1.11759\times10^{-8}, -1.11759\times10^{-8}, 1.38081, -1.11759\times10^{-8}, -1.11759\times10^{-8}, -0.986294, -1.11759\times10^{-8}, -1.11759\times10^{-8}, -1.93314, -1.11759\times10^{-8}, -1.11759\times10^{-8}, 1.53862\}$

In[407]:=

```
ListLinePlot[
 Transpose@Table[
   getDihedFD[
    {
     {0, 0, 0},
     {.5, .7, 0},
     {.7, .2, 0},
     {1.2, .7, x}
    },
    1,
    2,
    3,
    4
   ],
   {x, dihedScan}
  ],
 PlotRange → All
]
```

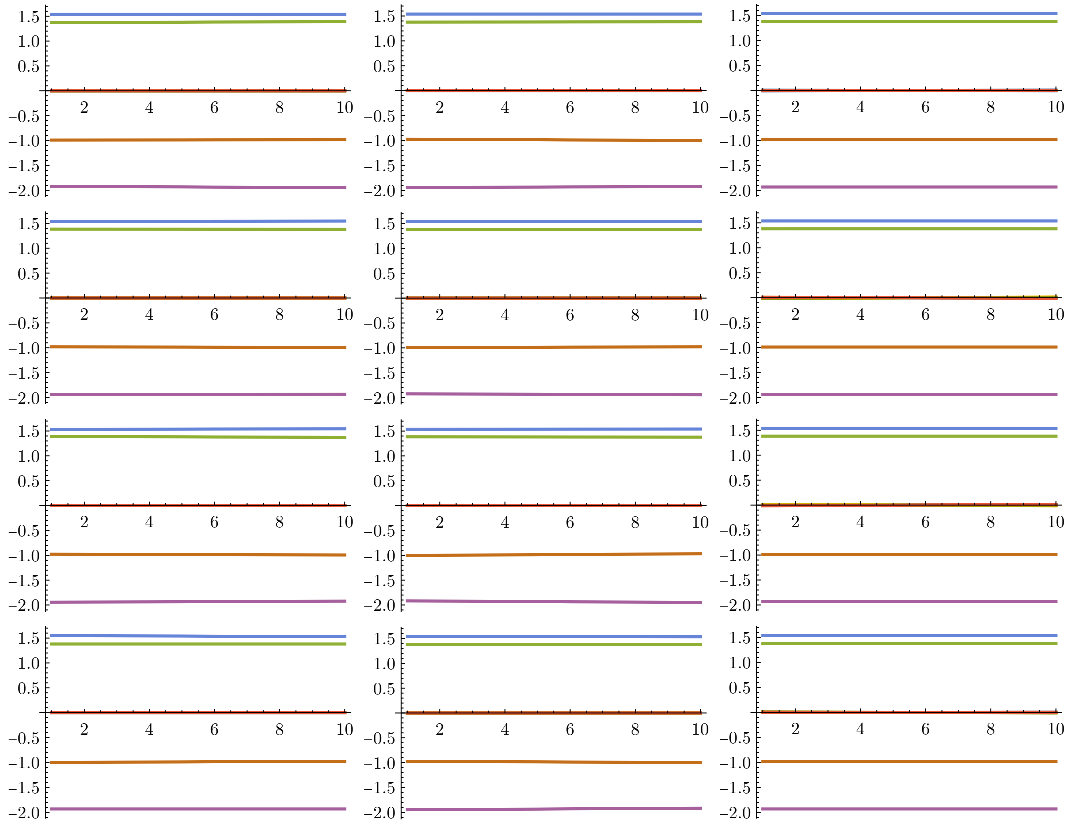
Out[407]=



In[408]:=

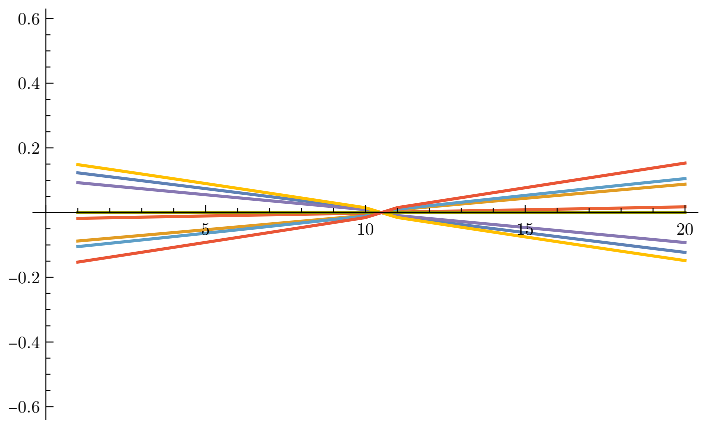
```
Table[
  ListLinePlot[
   Transpose@Table[
     getDihedFD[
      ReplacePart[#, {i, j} → #〚i, j〛 + x] &@{
        {0, 0, 0},
        {.5, .7, 0},
        {.7, .2, 0},
        {1.2, .7, 0.}
       },
      1,
      2,
      3,
      4
     ],
     {x, dihedScan}
    ],
   PlotRange → All
  ],
  {i, 4},
  {j, 3}
 ] // Grid
```

Out[408]=



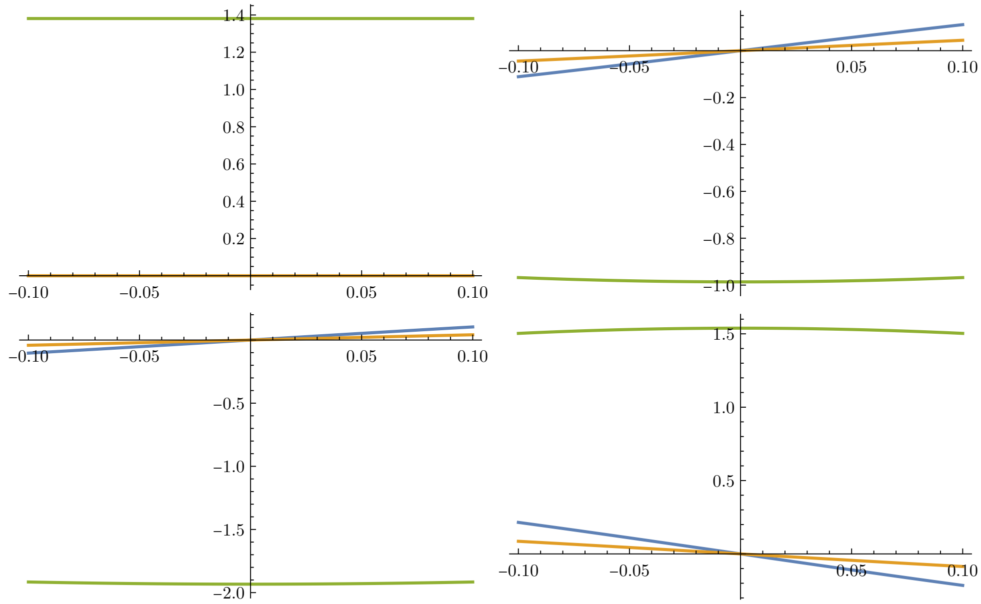
```
ListLinePlot[
 Transpose@Table[
   getDihedFD[
    {
     {0, 0, 0},
     {.5, .7, 0},
     {.7, .2, 0. + x},
     {1.2, .7, 0.}
    },
    1,
    2,
    3,
    4
   ],
   {x, dihedScan}
  ]
]
```

Out[401]=



```
Block[{d = .00005, struct,
    g = DeleteCases[#, 0.] &@Range[-.1, .1, .0005]},
   Table[
    ListLinePlot[
     Map[Transpose[{g, #}] &]@
      Table[
       struct = {
         {0, 0, 0},
         {.5, .7, 0},
         {.7, .2, 0},
         {1.2, .7, x}
        };
       ({-1 / 2, 0, 1 / 2} / d).
        Table[
         getDihed[
          ReplacePart[struct,
           j → struct〚j〛 + d * s * IdentityMatrix[3]〚i〛
          ],
          1,
          2,
          3,
          4
         ],
         {s, -1, 1}
        ],
       {i, 3},
       {x, g}
      ],
     ImageSize → 250
    ],
    {j, 4}
   ]
  ] // Partition[#, 2] & // Grid
```

Out[389]=



Expanded Terms

Second Derivatives

Implementation

Test implementations

```
In[1]:= testCoords = Input Coords ;
testCoordsSpec = Plus[…] ;
```

Analytic

Angles

```
sinCosDeriv[coords_, i_, j_, k_] :=
 Block[{a, b},
  a = coords〚j〛 - coords〚i〛;
  b = coords〚k〛 - coords〚i〛;

 ]
```

FD

Distances

Angles

FD Diheds

In[386]:=

```
getDihed[coords_, i_, j_, k_, l_] :=
  Block[
   {
    a = coords⟦j⟧ - coords⟦i⟧,
    b = coords⟦k⟧ - coords⟦j⟧,
    c = coords⟦l⟧ - coords⟦k⟧,
    axb,
    bxc,
    n,
    d1,
    d2
   },
   axb = Cross[a, b] // Normalize;
   bxc = Cross[b, c] // Normalize;
   n = Cross[Normalize[b], axb];
   ArcTan[Dot[axb, bxc], Dot[n, bxc]]
  ];
getDihed[i_, j_, k_, l_][coords_] :=
  getDihed[coords, i, j, k, l];
```

In[388]:=

```
getDihedFD[coords_, i_, j_, k_, l_, dx_ : .0000001] :=
  Table[
   NDSolve`FiniteDifferenceDerivative[
     1,
     Sequence @@
      Transpose@Table[
        {
         dx * a,
         getDihed[
          coords +
           ReplacePart[ConstantArray[0., Dimensions[coords]],
            {n, m} → dx * a
           ],
          i, j, k, l
         ]
        },
        {a, -5, 5, 1}
       ]
     ][[2]],
    {n, Dimensions[coords][[1]]},
    {m, Dimensions[coords][[2]]}
   ] // Flatten;
```

Useful Identities

Here are a list of useful transformations, esp. useful when implementing. We'll assume a and b are vectors that don't depend on one another

Dot Product

$$\begin{aligned}\frac{\partial}{\partial a}\,a\cdot b &= \frac{\partial a}{\partial a}\cdot b\\ &= \mathbb{I}_3\cdot b\\ &= b\end{aligned}$$

Vector Norm

$$\begin{aligned}\frac{\partial}{\partial a}|a| &= \frac{\partial}{\partial a}\sqrt{a\cdot a}\\ &= \frac{1}{2\,|a|}\,\frac{\partial}{\partial a}\,a\cdot a\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{|a|}\,\frac{\partial a}{\partial a} \cdot a \\
&= \frac{1}{|a|}\,\mathbb{I}_3\, a \\
&= \frac{a}{|a|} \\
&= \hat{a}
\end{aligned}
$$

although there is one subtlety when all of the elements are zero, because we end up with a bunch of 0/0 terms and we need to decide what they are. This is formally indeterminate but we will choose the convention that it is zero.

Normalized Vector (Vector Norm 2nd Derivative)

$$
\begin{aligned}
\frac{\partial \hat{a}}{\partial a} &= \frac{\partial}{\partial a}\,\frac{a}{|a|} \\
&= -\frac{1}{|a|^2}\left(\frac{\partial |a|}{\partial a} \otimes a\right) + \frac{1}{|a|}\,\frac{\partial a}{\partial a} \\
&= \frac{1}{|a|}\left(\mathbb{I}_3 - \frac{a}{|a|} \otimes \frac{a}{|a|}\right) \\
&= \frac{1}{|a|}\,(\mathbb{I}_3 - \hat{a} \otimes \hat{a})
\end{aligned}
$$

Cross Product Norm

$$
\begin{aligned}
\frac{\partial}{\partial a}|a \times b| &= \frac{\partial}{\partial a}\,\sqrt{(a \times b)\cdot(a \times b)} \\
&= \frac{1}{|a \times b|}\left(\frac{\partial}{\partial a}(a \times b)\right)\cdot(a \times b) \\
&= \frac{1}{|a \times b|}\left(\frac{\partial a}{\partial a} \times b\right)\cdot(a \times b) \\
&= \frac{1}{|a \times b|}\,(\mathbb{I}_3 \times b)\,(a \times b) \\
&= -\frac{(\epsilon_3\, b)\,(a \times b)}{|a \times b|} \\
&= \frac{b \times (a \times b)}{|a \times b|} \\
&= \frac{1}{|a \times b|}\,(a(b \cdot b)\ -\ b(a \cdot b))
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{|a|\,|b|\sin\theta}\left(a\,|b|^2 - b\,|a|\,|b|\cos\theta\right)\\
&= \hat{a}\,|b|\csc\theta - b\tan\theta
\end{aligned}
$$

Angle Derivatives

Derivatives of sines and cosines between vectors keep coming up, so we'll address them on their own. To start, we have

$$
\begin{aligned}
\sin(a,\ b) &= \frac{|a\times b|}{|a|\,|b|}\\
\cos(a,\ b) &= \frac{a\cdot b}{|a|\,|b|}
\end{aligned}
$$

and we'll also want to get derivatives of things like

$$\theta(a,\ b) = \tan^{-1}(\sin(a,\ b),\ \cos(a,\ b))$$

First Derivatives

Sin/Cos

Angles

Second Derivatives

Embedded Internal Derivatives

It turns out that the Cartesian coordinates we generate are actually expected to be referenced to the Eckart frame...where this is made explicit I don't know.

This means we actually generate coordinates like

$$
\begin{aligned}
X_O &= \text{zmtocart(internals)}\\
X &= \text{Ek}(X_O,\ \text{ref})\cdot(X_O - \text{COM}(X_O))
\end{aligned}
$$

and so when we take derivatives of this we have

$$\frac{\partial}{\partial r}X = \frac{\partial\ \text{Ek}(X_O,\ \text{ref})}{\partial r}\cdot(X_O - \text{COM}(X_O)) + \text{Ek}(X_O,\ \text{ref})\cdot\left(\frac{\partial}{\partial r}X_O - \frac{\partial}{\partial r}\text{COM}(X_O)\right)$$

and I can already evaluate

$$\frac{\partial}{\partial r}X_O$$

and from this it's relatively straightforward to evaluate

$$\frac{\partial}{\partial r}\mathrm{COM}(X_O)$$

so then the question is what can be done about

$$\frac{\partial\ \mathrm{Ek}(X_O,\ \mathrm{ref})}{\partial r}$$

for that we'll use the definition that

$$U,\ P\ =\ \mathrm{pd}\ \sum_i \frac{m_i}{M_T}\ (X_O)_i \otimes (\mathrm{ref})_i$$

$$\mathrm{Ek}(X_O,\ \mathrm{ref}) = U$$

where *pd* is the *polar decomposition*, defined for a real symmetric matrix by

$$P = \left(A^T A\right)^{\frac{1}{2}}$$
$$U = A P^{-1}$$

which gives us the relation

$$A\ =\ \sum_i \frac{m_i}{M_T}\ (X_O)_i \otimes (\mathrm{ref})_i$$

$$\mathrm{Ek}\,(X_O,\ \mathrm{ref})\ =\ A\left(A^T A\right)^{-\frac{1}{2}}$$

Much of this is straightforwardly differentiable...but $\left(A^T A\right)^{-\frac{1}{2}}$ is a challenge. It's formally defined by first diagonalizing $A^T A$ to get

$$A^T A = Q D Q^{-1}$$

and then

$$\left(A^T A\right)^{-\frac{1}{2}} = Q D^{-\frac{1}{2}} Q^{-1}$$

where

$$\left(D^{-\frac{1}{2}}\right)_{ij} = \lambda_i^{-\frac{1}{2}}\ \delta_{ij}$$

the issue now is that we need explicit algebraic forms for λ_i and Q if we want to differentiate this...

Because of this, it seems like the easiest approach might be to write

$$\frac{\partial\ \mathrm{Ek}(X_O,\ \mathrm{ref})}{\partial r} = \frac{\partial X_O}{\partial r}\ \frac{\partial\ \mathrm{Ek}(X_O,\ \mathrm{ref})}{\partial X_O}$$

and then numerically evaluate

$$\frac{\partial \text{ Ek}(X_O, \text{ ref})}{\partial X_O}$$

since direct numerical evaluation of

$$\frac{\partial \text{ Ek}(X_O, \text{ ref})}{\partial r}$$

will be slowed by the need to transform to Cartesian coordinates for each displacement in r