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

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

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

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

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

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

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

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

which means that we get

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

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

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

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

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

$$\frac{d}{dq}\,u_i = \frac{dx_{i,b}}{dq} - \frac{dx_{i,a}}{dq}$$

$$\frac{d}{dq}\,u_i = \frac{d}{dq}\,(x_{i,c} - x_{i,a}) = \frac{dx_{i,c}}{dq} - \frac{dx_{i,a}}{dq}$$

$$\frac{d}{dq}\,n_i = \frac{d}{dq}\,v_i\times u_i = \frac{d}{dq}\,v_i\times u_i + v_i\times\frac{d}{dq}\,u_i$$

$$\frac{dz_{i,2}}{dq}=\delta_{qz_{i,2}}$$
$$\frac{dz_{i,3}}{dq}=\delta_{qz_{i,3}}$$

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

Correct Answer

```
In[12]:= correctAns = CartByZ ;

correctAnsSpec = Plus[...] ;

zCoords = Z-Coords ;
```

Code

```
In[1148]:= zmatDerivsC =
  With[{ee = Normal@LeviCivitaTensor[3]},
   Compile[{
     {zmCoords, _Real, 2},
     {spec, _Integer, 2},
     {origin, _Real, 1},
     {xAxis, _Real, 1},
```

```
  {yAxis, _Real, 1}
 },
 Block[
  {
   zeroMat = Table[0., {i, 3}, {j, 3}],
   zeroVec = Table[0., {i, 3}]
  },
  Block[
   {
    coords = Table[0., {i, Length[spec] + 1}, {j, 3}],
    derivs =
     Table[0., {i, Length[spec] + 1}, {j, 3}, {k, Length[spec] + 1}, {l, 3}],
    r, q, f,
    ia, ib, ic,
    vi = zeroVec, ui = zeroVec, ni = zeroVec,
    nvi = 0., nui = 0., nni = 0.,
    R1, R2, Q,
    dR1, dR2,
    drn, dqn, dfn,
    xi1 = zeroVec, xi2 = zeroVec, xi3 = zeroVec,
    dxi1 = zeroVec, dxi2 = zeroVec, dxi3 = zeroVec,
    dv = zeroVec, du = zeroVec, dni = zeroVec,
    dQ = zeroMat,
    i3 = IdentityMatrix[3],
    e3 = ee
   },
   coords[[1]] = origin;
   Do[
    {r, q, f} = zmCoords[[i, {1, 2, 3}]];
    {ia, ib, ic} = spec[[i, {2, 3, 4}]];
    (* choose prior coordinates based on passed spec *)
    xi1 = coords[[ia]];
    If[i > 1, xi2 = coords[[ib]], xi2 = zeroVec];
    If[i > 2, xi3 = coords[[ic]], xi3 = zeroVec];
    (* generate current coord *)
    If[i > 1, vi = xi2 - xi1, vi = xAxis];
    nvi = Sqrt[Total[vi^2]]; vi /= nvi;
    If[i > 2, ui = xi3 - xi2, ui = yAxis];
    nui = Sqrt[Total[ui^2]]; ui /= nui;
    ni = Cross[vi, ui]; nni = Sqrt[Total[ni^2]]; ni /= nni;
    R1 = Outer[Times, vi, vi]
      + (i3 - Outer[Times, vi, vi]) * Cos[f]
      - Dot[e3, vi] * Sin[f];
```

```
R2 = Outer[Times, ni, ni]
   + (i3 - Outer[Times, ni, ni]) * Cos[q]
   - Dot[e3, ni] * Sin[q];
Q = R1.R2;
coords[[i + 1]] = xi1 + Q.(r * vi);
Do[
 If[ia > 1, dxi1 = derivs[[n, m, ia - 1]], dxi1 = zeroVec];
 If[ib > 1, dxi2 = derivs[[n, m, ib - 1]], dxi2 = zeroVec];
 If[ic > 1, dxi3 = derivs[[n, m, ic - 1]], dxi3 = zeroVec];
 If[i > 1,
  dv = dxi2 - dxi1;
  dv = 1 / nvi * (dv - vi * Dot[dv, vi]),
  dv = zeroVec
 ];
 If[i > 2,
  du = dxi3 - dxi2;
  du = 1 / nui * (du - ui * Dot[du, ui]);,
  du = zeroVec
 ];
 drn = n == i && m == 1;
 dqn = n == i && m == 2;
 dfn = n == i && m == 3;
 If[i > 1,
  dni = Cross[dv, ui] + Cross[vi, du];
  dni = 1 / nni * (dni - ni * Dot[dni, ni]);
  (* generate terms in deriv expression *)
  dR2 =
   Outer[Times, dni, ni] + Outer[Times, ni, dni]
    - (Outer[Times, dni, ni] + Outer[Times, ni, dni]) * Cos[q]
    - If[dqn, (i3 - Outer[Times, ni, ni]) * Sin[q], zeroMat]
    - Dot[e3, dni] * Sin[q]
    - If[dqn, Dot[e3, ni] * Cos[q], zeroMat];
  If[i > 2,
   dR1 =
    Outer[Times, dv, vi] + Outer[Times, vi, dv]
     - (Outer[Times, dv, vi] + Outer[Times, vi, dv]) * Cos[f]
     - If[dfn, (i3 - Outer[Times, vi, vi]) * Sin[f],  zeroMat]
     - Dot[e3, dv] * Sin[f]
     - If[dfn, Dot[e3, vi] * Cos[f], zeroMat];
   dQ = dR1.R2 + R1.dR2,
   If[i == 2,
    dQ = R1.dR2,
    dQ = ConstantArray[0., {3, 3}]
```

```
          ]
         ]
        ];
        derivs[[n, m, i]] =
         dxi1 + If[i > 1,
           Dot[dQ, r * vi] + Dot[Q, If[drn, vi, zeroVec] + r * dv],
           If[drn, vi, zeroVec]
          ];
        (*If[n==2&&m==2,
          Echo[
           MatrixForm/@{
             {r,q,f},
             {vi, ui, ni},
             {dv, du, dni}, Q, dQ,
             dxi1,
             Total@{Dot[dQ, r*vi],Dot[Q,r*dv]},
             {Dot[dQ, r*vi],Dot[Q,r*dv]}
            }, {{n, m}, i}]
         ];*),
        {n, i},
        {m, 3}
       ],
       {i, 1, Length[spec]}
      ];
      derivs[[-1, 1]] = coords;
      derivs
     ]
    ]
   ]
  ];
```

In[981]:=
```
zmatDerivs // Clear
zmatDerivs[
  zmCoords_, spec_,
  Optional[{origin_, {xAxis_, yAxis_}},
   {{0., 0., 0.}, {{1., 0., 0.}, {0., 1., 0}}}]
 ] :=
 With[{cRes = zmatDerivsC[zmCoords, spec, origin, xAxis, yAxis]},
  {
   cRes[[-1, 1]],
   ArrayReshape[
    cRes[[ ;; -2,  ;; ,  ;; -2,  ;;]], {Length[spec] * 3, Length[spec] * 3}]
  }
 ]
```

Cartesian to Z-Matrix

Derivation

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

$$
\begin{aligned}
r_{ij} &= |a| && \text{where } a = x_j - x_i \\
\theta_{ijk} &= \operatorname{arctan2}\left(\frac{|a\times b|}{|a|\,|b|}, \frac{a\cdot b}{|a|\,|b|}\right) && \text{where } a = x_j - x_i \text{ and } b = x_k - x_i \\
\tau_{ijkl} &= \operatorname{arctan2}\left(\hat{n}_1\cdot\hat{n}_2, \left(\hat{n}_1\times\hat{b}\right)\cdot\hat{n}_2\right) && \text{where } a = x_j - x_i,\ b = x_k - x_j,\ c = x_l - x_k \\
& && \hat{n}_1 = \frac{a\times b}{|a\times b|},\ \hat{n}_2 = \frac{b\times c}{|b\times c|},\ \hat{b} = \frac{b}{|b|}
\end{aligned}
$$

Case 1: r

so for our r terms we have

$$
\begin{aligned}
r_{ij} &= |x_j - x_i| = \sqrt{(x_j - x_i)\cdot(x_j - x_i)} \\
\frac{\partial r_{ij}}{\partial x_{nm}} &= \frac{\partial (x_j - x_i)\cdot(x_j - x_i)}{\partial x_{nm}} \frac{1}{2\sqrt{(x_j - x_i)\cdot(x_j - x_i)}} \\
&= \left(\frac{\partial (x_j - x_i)}{\partial x_{nm}}\cdot(x_j - x_i) + (x_j - x_i)\cdot\frac{\partial (x_j - x_i)}{\partial x_{nm}}\right) \frac{1}{2\sqrt{(x_j - x_i)\cdot(x_j - x_i)}} \\
&= \left(\frac{\partial (x_j - x_i)}{\partial x_{nm}}\cdot(x_j - x_i)\right) \frac{1}{|x_j - x_i|}
\end{aligned}
$$

this is a bit weird to think about, given that these are vector quantities, but we can expand them out in terms of their components to get

$$
\begin{aligned}
\frac{\partial (x_j - x_i)}{\partial x_{nm}} &= \frac{\partial x_j}{\partial x_{nm}} - \frac{\partial x_i}{\partial x_{nm}} \\
&= \left(\frac{\partial x_{j1}}{\partial x_{nm}} - \frac{\partial x_{i1}}{\partial x_{nm}}, \frac{\partial x_{j2}}{\partial x_{nm}} - \frac{\partial x_{i2}}{\partial x_{nm}}, \frac{\partial x_{j3}}{\partial x_{nm}} - \frac{\partial x_{i3}}{\partial x_{nm}}\right)
\end{aligned}
$$

and from this we can determine each of these elements, giving us

$$\frac{\partial x_{jl}}{\partial x_{nm}}-\frac{\partial x_{il}}{\partial x_{nm}}=\begin{cases}1 & j=n \text{ and } l=m\\ -1 & i=n \text{ and } l=m\\ 0 & \text{else}\end{cases}$$
$$=(\delta_{jn}-\delta_{in})\,\delta_{lm}$$

and so in total we get

$$\frac{\partial r_{ij}}{\partial x_{nm}}=((\delta_{jn}-\delta_{in})\,\delta_{lm}\cdot(x_j-x_i))\,\frac{1}{|x_j-x_i|}$$

which we can read as being the m^{th} element of the vector

$$\frac{\partial r_{ij}}{\partial x_n}=\frac{(x_j-x_i)}{|x_j-x_i|}\,(\delta_{jn}-\delta_{in})$$

Case 2: θ

Next, for θ we'll do this all in terms of a and b giving us

$$\begin{aligned}\frac{\partial\theta_{ijk}}{\partial x_{nm}}&=\frac{\partial}{\partial x_{nm}}\operatorname{arctan2}\left(\frac{|a\times b|}{|a|\,|b|},\ \frac{a\cdot b}{|a|\,|b|}\right)\\&=\frac{\partial a}{\partial x_{nm}}\,\frac{\partial}{\partial a}\operatorname{arctan2}\left(\frac{|a\times b|}{|a|\,|b|},\ \frac{a\cdot b}{|a|\,|b|}\right)+\frac{\partial b}{\partial x_{nm}}\,\frac{\partial}{\partial b}\operatorname{arctan2}\left(\frac{|a\times b|}{|a|\,|b|},\ \frac{a\cdot b}{|a|\,|b|}\right)\end{aligned}$$

this will get nasty quick, so we're going to do a replacement and say

$$s(a,\ b)=\frac{|a\times b|}{|a|\,|b|}$$
$$c(a,\ b)=\frac{a\cdot b}{|a|\,|b|}$$

next we'll do our derivatives term-wise to get

$$\frac{\partial}{\partial a}\operatorname{arctan2}(s(a,\ b)\ c(a,\ b))=\frac{\partial s(a,\ b)}{\partial a}\,\frac{c(a,b)}{c(a,\ b)^2+s(a,\ b)^2}-\frac{\partial c(a,\ b)}{\partial a}\,\frac{s(a,b)}{c(a,\ b)^2+s(a,\ b)^2}$$
$$\frac{\partial}{\partial b}\operatorname{arctan2}\,(s\,(a,\ b)\ c\,(a,\ b))=\frac{\partial s(a,\ b)}{\partial b}\,\frac{c(a,b)}{c(a,\ b)^2+s(a,\ b)^2}-\frac{\partial c(a,\ b)}{\partial b}\,\frac{s(a,b)}{c(a,\ b)^2+s(a,\ b)^2}$$

and we'll apply a bit of "special" knowledge that

$$s(a,\ b)=\sin\theta_{ijk}$$
$$c(a,\ b)=\cos\theta_{ijk}$$
$$c(a,\ b)^2+s(a,\ b)^2=1$$

giving us

$$\frac{\partial\theta_{ijk}}{\partial x_{nm}}=\left(\frac{\partial a}{\partial x_{nm}}\,\frac{\partial s(a,\ b)}{\partial a}+\frac{\partial b}{\partial x_{nm}}\,\frac{\partial s(a,\ b)}{\partial b}\right)c(a,b)-\left(\frac{\partial a}{\partial x_{nm}}\,\frac{\partial c(a,\ b)}{\partial a}+\frac{\partial b}{\partial x_{nm}}\,\frac{\partial c(a,\ b)}{\partial b}\right)s(a,\ b)$$

Now we get to deal with these chain rule terms. First off, we'll look at

$$\frac{\partial a}{\partial x_{nm}}=\frac{\partial}{\partial x_{nm}}(x_j-x_i)=\frac{\partial x_j}{\partial x_{nm}}-\frac{\partial x_i}{\partial x_{nm}}$$

$$\frac{\partial b}{\partial x_{nm}}=\frac{\partial}{\partial x_{nm}}(x_k-x_i)=\frac{\partial x_k}{\partial x_{nm}}-\frac{\partial x_i}{\partial x_{nm}}$$

and using the same argument as before, we get

$$\begin{aligned}\frac{\partial a}{\partial x_{nm}}&=(\delta_{jn}-\delta_{in})\,\delta_{lm}\\ \frac{\partial b}{\partial x_{nm}}&=(\delta_{kn}-\delta_{in})\,\delta_{lm}\end{aligned}$$

then plugging this back in, we get

$$\frac{\partial\theta_{ijk}}{\partial x_{nm}}=\left((\delta_{jn}-\delta_{in})\left(\frac{\partial c(a,\ b)}{\partial a}\,c(a,b)-\frac{\partial s(a,\ b)}{\partial a}\,s(a,b)\right)+(\delta_{kn}-\delta_{in})\left(\frac{\partial c(a,\ b)}{\partial b}\,c(a,b)-\frac{\partial s(a,\ b)}{\partial b}\,s(a,b)\right)\right)\delta_{lm}$$

which we note is the same thing as taking the m^{th} element of the vector quantity

$$\frac{\partial\theta_{ijk}}{\partial x_n}=(\delta_{jn}-\delta_{in})\left(\frac{\partial s(a,\ b)}{\partial a}\,c(a,b)-\frac{\partial c(a,\ b)}{\partial a}\,s(a,b)\right)+(\delta_{kn}-\delta_{in})\left(\frac{\partial s(a,\ b)}{\partial b}\,c(a,b)-\frac{\partial c(a,\ b)}{\partial b}\,s(a,b)\right)$$

we've got to do the same thing for the terms like $\frac{\partial c(a,\ b)}{\partial a}$ and $\frac{\partial s(a,\ b)}{\partial a}$. Here we have

$$\begin{aligned}\frac{\partial}{\partial a}s(a,\ b)&=\frac{\partial}{\partial a}\frac{|a\times b|}{|a|\,|b|}\\ \frac{\partial}{\partial a}c(a,\ b)&=\frac{\partial}{\partial a}\frac{a\cdot b}{|a|\,|b|}\end{aligned}$$

and first we'll apply the product rule to get

$$\begin{aligned}\frac{\partial}{\partial a}\frac{|a\times b|}{|a|\,|b|}&=\frac{1}{|a|\,|b|}\,\frac{\partial}{\partial a}|a\times b|-\frac{|a\times b|}{(|a|\,|b|)^2}\,|b|\,\frac{\partial}{\partial a}|a|\\ \frac{\partial}{\partial a}\frac{a\cdot b}{|a|\,|b|}&=\frac{1}{|a|\,|b|}\,\frac{\partial}{\partial a}a\cdot b-\frac{a\cdot b}{(|a|\,|b|)^2}\,|b|\,\frac{\partial}{\partial a}|a|\\ &=\frac{\mathbb{1}\cdot b}{|a|\,|b|}-\frac{a\cdot b}{(|a|\,|b|)^2}\,|b|\,\frac{\partial}{\partial a}|a|\end{aligned}$$

then we rewrite these norms like

$$\begin{aligned}\frac{\partial}{\partial a}|a|&=\frac{\partial}{\partial a}\sqrt{a\cdot a}\\ &=\frac{1}{2\,|a|}\,\frac{\partial}{\partial a}a\cdot a\\ &=\frac{1}{|a|}\,\mathbb{1}_3\,a\\ &=\frac{a}{|a|}\end{aligned}\qquad\begin{aligned}\frac{\partial}{\partial a}|a\times b|&=\frac{\partial}{\partial a}\sqrt{(a\times b)\cdot(a\times b)}\\ &=\frac{1}{|a\times b|}\left(\frac{\partial}{\partial a}(a\times b)\right)\cdot(a\times b)\\ &=\frac{1}{|a\times b|}\left(\frac{\partial a}{\partial a}\times b\right)\cdot(a\times b)\\ &=\frac{1}{|a\times b|}\,(\mathbb{1}_3\times b)\,(a\times b)\\ &=-\frac{(\epsilon_3\,b)\,(a\times b)}{|a\times b|}\end{aligned}$$

where ϵ_3 is the 3D Levi-Cevita tensor, in total giving us

$$
\begin{aligned}
\frac{\partial}{\partial a}s(a,\ b) &= \frac{\partial}{\partial a}\frac{|a\times b|}{|a|\,|b|}\\
&= -\frac{(\epsilon_3\, b)\,(a\times b)}{|a|\,|b|\,|a\times b|} - \frac{a}{|a|}\,\frac{|a\times b|\,|b|}{(|a|\,|b|)^2}\\
&= \frac{1}{|a|}\left(-\epsilon_3\,\frac{b}{|b|}\,\frac{a\times b}{|a\times b|} - s(a,\ b)\,\frac{a}{|a|}\right)\\
\frac{\partial}{\partial a}c(a,\ b) &= \frac{\partial}{\partial a}\frac{a\cdot b}{|a|\,|b|}\\
&= \frac{b}{|a|\,|b|} - \frac{a}{|a|}\,\frac{(a\cdot b)\,|b|}{(|a|\,|b|)^2\,|a|}\\
&= \frac{1}{|a|}\left(\frac{b}{|b|} - c(a,b)\,\frac{a}{|a|}\right)\\
\frac{\partial}{\partial b}s(a,\ b) &= \frac{\partial}{\partial b}\frac{|a\times b|}{|a|\,|b|}\\
&= \frac{(\epsilon_3\ a)\,(a\times b)}{|a|\,|b|\,|a\times b|} - b\,\frac{|a\times b|\,|a|}{(|a|\,|b|)^2\,|b|}\\
&= \frac{1}{|b|}\left(\epsilon_3\,\frac{a}{|a|}\,\frac{a\times b}{|a\times b|} - s(a,\ b)\,\frac{b}{|b|}\right)\\
\frac{\partial}{\partial b}c(a,\ b) &= \frac{\partial}{\partial b}\frac{a\cdot b}{|a|\,|b|}\\
&= \frac{a}{|a|\,|b|} - b\,\frac{(a\cdot b)\,|a|}{(|a|\,|b|)^2\,|b|}\\
&= \frac{1}{|b|}\left(\frac{a}{|a|} - c(a,b)\,\frac{b}{|b|}\right)
\end{aligned}
$$

In the name of saving space, we won't write out the full expressions for these terms here, but now we'll note that we can plug back into our original expressions to get these terms.

Just because it's worth noting, though, for the future or something, here are the derivatives for the sine and cosine of the angle between two vectors

$$
\begin{aligned}
\frac{\partial}{\partial a}\sin(a,\ b) &= \frac{1}{|a|}\left(-\epsilon_3\,\frac{b}{|b|}\,\frac{a\times b}{|a\times b|} - \sin(a,\ b)\,\frac{a}{|a|}\right)\\
\frac{\partial}{\partial a}\cos(a,\ b) &= \frac{1}{|a|}\left(\frac{b}{|b|} - \cos(a,b)\,\frac{a}{|a|}\right)\\
\frac{\partial}{\partial b}\sin(a,\ b) &= \frac{1}{|b|}\left(\epsilon_3\,\frac{a}{|a|}\,\frac{a\times b}{|a\times b|} - \sin(a,\ b)\,\frac{b}{|b|}\right)\\
\frac{\partial}{\partial b}\cos\,(a,\ b) &= \frac{1}{|b|}\left(\frac{a}{|a|} - \cos(a,b)\,\frac{b}{|b|}\right)
\end{aligned}
$$

Vector Derivatives

I've done a slightly odd thing here that I probably need to justify.

Each of these derivatives is actually a *vector* of derivatives, i.e.

$$\frac{\partial}{\partial a}s(a,\ b)=\begin{pmatrix}\frac{\partial}{\partial a_1} & \frac{\partial}{\partial a_2} & \frac{\partial}{\partial a_3}\end{pmatrix} s(a_1,\ a_2,\ a_3,\ b_1,\ b_2,\ b_3)$$

when we take these derivatives, however, most terms are untouched by the derivative (i.e. think of the product rule), so we get stuff like

$$\frac{\partial}{\partial a}|a|=\frac{1}{2\,|a|}\frac{\partial}{\partial a}a\cdot a$$

but sometimes we *do* need to think about the elements, like in

$$\begin{aligned}\frac{\partial}{\partial a}a\cdot a&=\left(\frac{\partial a}{\partial a}\cdot a+a\cdot\frac{\partial a}{\partial a}\right)\\ &=2\,\frac{\partial a}{\partial a}\cdot a\\ &=2\left(\begin{pmatrix}\frac{\partial}{\partial a_1} & \frac{\partial}{\partial a_2} & \frac{\partial}{\partial a_3}\end{pmatrix} a\right)\cdot a\end{aligned}$$

and yet because everything is linear and vectors it usually ends up cleaning up nicely

$$\begin{aligned}\frac{\partial}{\partial a}a\cdot a&=2\left(\begin{pmatrix}\frac{\partial}{\partial a_1} & \frac{\partial}{\partial a_2} & \frac{\partial}{\partial a_3}\end{pmatrix} a\right)\cdot a\\ &=2\left(\begin{pmatrix}\frac{\partial}{\partial a_1} & \frac{\partial}{\partial a_2} & \frac{\partial}{\partial a_3}\end{pmatrix}(a_1\quad a_2\quad a_3)\right)\cdot a\\ &=2\begin{pmatrix}\frac{\partial a_1}{\partial a_1} & \frac{\partial a_2}{\partial a_1} & \frac{\partial a_3}{\partial a_1}\\ \frac{\partial a_1}{\partial a_2} & \frac{\partial a_2}{\partial a_2} & \frac{\partial a_3}{\partial a_2}\\ \frac{\partial a_1}{\partial a_3} & \frac{\partial a_2}{\partial a_3} & \frac{\partial a_3}{\partial a_3}\end{pmatrix} a\\ &=2\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix} a\\ &=2\ \mathbb{I}_3\ a\\ &=2\ a\end{aligned}$$

one other particularly interesting case is

$$\begin{aligned}\frac{\partial}{\partial a}(a\times b)&=\frac{\partial a}{\partial a}\times b+a\times\frac{\partial b}{\partial a}\\ &=\frac{\partial a}{\partial a}\times b\\ &=\mathbb{I}_3\times b\end{aligned}$$

and then the question is “what the heck does $\mathbb{I}_3\times b$ mean?”

So we break the overall derivative down by elements as

$$\frac{\partial}{\partial a}(a\times b)=\begin{pmatrix}\frac{\partial}{\partial a_1} & \frac{\partial}{\partial a_2} & \frac{\partial}{\partial a_3}\end{pmatrix}(a\times b)$$

$$
\begin{aligned}
&=\left(\frac{\partial a}{\partial a_1}\times b \quad \frac{\partial a}{\partial a_2}\times b \quad \frac{\partial a}{\partial a_3}\times b\right)\\
&=((1\quad 0\quad 0)\times b\quad (0\quad 1\quad 0)\times b\quad (0\quad 0\quad 1)\times b)\,(a\times b)
\end{aligned}
$$

and so by assuming that $\times$ threads over the outermost axis of a tensor, just like $\cdot$, this makes sense as an operation. But we can do one better. To avoid having to track both cross and dot products through calculations like these, mathematicians invented a tensor that encodes the operation of a cross product in a dot product, called either the *antisymmetric tensor* or the *Levi-Cevita tensor*, which this time we're calling ϵ. That means we can write

$$
\begin{aligned}
b\times\mathbb{I}_3&=\epsilon_3\, b\\
\mathbb{I}_3\times b&=-\epsilon_3\, b
\end{aligned}
$$

and now we have no more cross products.

Case 3: τ

We start with

$$
\tau_{ijkl}=\operatorname{arctan2}\left(\left(\hat{b}\times\hat{n}_1\right)\cdot\hat{n}_2,\ \hat{n}_1\cdot\hat{n}_2\right)\quad \text{where } a=x_j-x_i,\ b=x_k-x_j,\ c=x_l-x_k
$$
$$
\hat{n}_1=\frac{a\times b}{|a\times b|},\ \hat{n}_2=\frac{b\times c}{|b\times c|},\ \hat{b}=\frac{b}{|b|}
$$

and we'll do all of this with respect to a, b, and c, giving us

$$
\begin{aligned}
\frac{\partial\tau_{ijkl}}{\partial x_{nm}}&=\frac{\partial}{\partial x_{nm}}\operatorname{arctan2}\left(\hat{n}_1\cdot\hat{n}_2,\ \left(\hat{b}\times\hat{n}_1\right)\cdot\hat{n}_2\right)\\
&=\left(\frac{\partial a}{\partial x_{nm}}\quad \frac{\partial b}{\partial x_{nm}}\ \frac{\partial c}{\partial x_{nm}}\right)\cdot\left(\frac{\partial}{\partial a}\quad\frac{\partial}{\partial b}\quad\frac{\partial}{\partial c}\right)\operatorname{arctan2}\left(\hat{n}_1\cdot\hat{n}_2,\ \left(\hat{b}\times\hat{n}_1\right)\cdot\hat{n}_2\right)
\end{aligned}
$$

as before, we know how to deal with the first three terms, so next we'll look at the arctan2 terms, where we have things like

$$
\frac{\partial}{\partial\alpha}\operatorname{arctan2}\left(\hat{n}_1\cdot\hat{n}_2,\ \left(\hat{b}\times\hat{n}_1\right)\cdot\hat{n}_2\right)
$$

which, noting once again that our arguments are a $sin\tau$ and $cos\tau$, will give us terms like

$$
\frac{\partial}{\partial\alpha}\operatorname{arctan2}\left(\hat{n}_1\cdot\hat{n}_2,\ \left(\hat{n}_1\times\hat{b}\right)\cdot\hat{n}_2\right)=\frac{\partial\left(\left(\hat{b}\times\hat{n}_1\right)\cdot\hat{n}_2\right)}{\partial\alpha}\,\hat{n}_1\cdot\hat{n}_2-\frac{\partial\hat{n}_1\cdot\hat{n}_2}{\partial\alpha}\left(\hat{b}\times\hat{n}_1\right)\cdot\hat{n}_2
$$

in the interest of space I won't expand that out in full, but we'll jump in and evaluate the derivatives

$$
\begin{aligned}
\frac{\partial\hat{n}_1\cdot\hat{n}_2}{\partial\alpha}&=\left(\frac{\partial\hat{n}_1}{\partial\alpha}\cdot\hat{n}_2+\hat{n}_1\cdot\frac{\partial\hat{n}_2}{\partial\alpha}\right)\\
\frac{\partial\left(\left(\hat{n}_1\times\hat{b}\right)\cdot\hat{n}_2\right)}{\partial\alpha}&=\left(\hat{b}\times\frac{\partial\hat{n}_1}{\partial\alpha}+\frac{\partial}{\partial\alpha}\hat{b}\times\hat{n}_1\right)\cdot\hat{n}_2+\left(\hat{b}\times\hat{n}_1\right)\cdot\frac{\partial\hat{n}_2}{\partial\alpha}
\end{aligned}
$$

here we see three derivative terms (even though they're all basically a normalization)

$$
\begin{aligned}
\frac{\partial \hat{n}_1}{\partial \alpha} &= \frac{\partial}{\partial \alpha}\frac{a\times b}{|a\times b|} \\
&= \frac{1}{|a\times b|}\left(\frac{\partial}{\partial \alpha}a\times b - \left(\frac{\partial}{\partial \alpha}a\times b\right)\frac{a\times b}{|a\times b|}\,\hat{n}_1\right) \\
&= \frac{1}{|a\times b|}\,\frac{\partial a\times b}{\partial \alpha}\left(\mathbb{I}_3 - \hat{n}_1\hat{n}_1\right) \\
\frac{\partial \hat{n}_2}{\partial \alpha} &= \frac{\partial}{\partial \alpha}\frac{b\times c}{|b\times c|} \\
&= \frac{1}{|b\times c|}\left(\frac{\partial}{\partial \alpha}b\times c - \left(\frac{\partial}{\partial \alpha}b\times c\right)\frac{b\times c}{|b\times c|}\,\hat{n}_2\right) \\
&= \frac{1}{|b\times c|}\,\frac{\partial b\times c}{\partial \alpha}\left(\mathbb{I}_3 - \hat{n}_2\hat{n}_2\right) \\
\frac{\partial \hat{b}}{\partial \alpha} &= \frac{\partial}{\partial \alpha}\frac{b}{|b|} \\
&= \frac{1}{|b|}\,\frac{\partial b}{\partial \alpha} - \frac{\partial |b|}{\partial \alpha}\,\frac{b}{|b|^2} \\
&= \frac{1}{|b|}\left(\frac{\partial b}{\partial \alpha} - \frac{\partial |b|}{\partial \alpha}\,\hat{b}\right) \\
&= \frac{1}{|b|}\,\frac{\partial b}{\partial \alpha}\left(\mathbb{I}_3 - \hat{b}\,\hat{b}\right)
\end{aligned}
$$

here it's important to note that we read, e.g., $\hat{b}\,\hat{b}$ as being the *outer product* of $\hat{b}$ with itself, i.e a column vector times a row vector. Shape considerations would lead to that conclusion naturally, but it's worth pointing out.

Returning to the original calculation, we'll recall from before that our cross products work like

$$
\begin{aligned}
\frac{\partial}{\partial \alpha}a\times b &= \left(\frac{\partial a}{\partial \alpha}\times b + a\times\frac{\partial b}{\partial \alpha}\right) \\
&= \left(\epsilon_3\, a\,\delta_{b\alpha} - \epsilon_3\, b\,\delta_{a\alpha}\right)
\end{aligned}
$$

and so all told we get stuff like

$$
\begin{aligned}
\frac{\partial \hat{n}_1}{\partial a} &= \frac{1}{|a\times b|}\left(\epsilon_3\, a\,\delta_{b\alpha} - \epsilon_3\, b\,\delta_{a\alpha}\right)\left(\mathbb{I}_3 - \hat{n}_1\hat{n}_1\right) \\
\frac{\partial \hat{n}_2}{\partial \alpha} &= \frac{1}{|b\times c|}\left(\epsilon_3\, b\,\delta_{c\alpha} - \epsilon_3\, c\,\delta_{b\alpha}\right)\left(\mathbb{I}_3 - \hat{n}_2\hat{n}_2\right) \\
\frac{\partial \hat{b}}{\partial \alpha} &= \frac{1}{|b|}\,\delta_{b\alpha}\left(\mathbb{I}_3 - \hat{b}\,\hat{b}\right)
\end{aligned}
$$

or in maybe more readable format

$$
\frac{\partial \hat{n}_1}{\partial a} = -\frac{\epsilon_3\, b}{|a\times b|}\left(\mathbb{I}_3 - \hat{n}_1\hat{n}_1\right)
$$

$$\frac{\partial \hat{n}_1}{\partial b} = \frac{\epsilon_3 a}{|a \times b|} \left(\mathbb{I}_3 - \hat{n}_1 \hat{n}_1\right)$$
$$\frac{\partial \hat{n}_1}{\partial c} = \emptyset$$
$$\frac{\partial \hat{n}_2}{\partial a} = \emptyset$$
$$\frac{\partial \hat{n}_2}{\partial b} = -\frac{\epsilon_3 c}{|b \times c|} \left(\mathbb{I}_3 - \hat{n}_2 \hat{n}_2\right)$$
$$\frac{\partial \hat{n}_2}{\partial c} = \frac{\epsilon_3 b}{|b \times c|} \left(\mathbb{I}_3 - \hat{n}_2 \hat{n}_2\right)$$
$$\frac{\partial \hat{b}}{\partial a} = \emptyset$$
$$\frac{\partial \hat{b}}{\partial b} = \frac{1}{|b|} \left(\mathbb{I}_3 - \hat{b}\,\hat{b}\right)$$
$$\frac{\partial \hat{b}}{\partial c} = \emptyset$$

and so to recap, we get

$$\frac{\partial \tau_{ijkl}}{\partial x_{nm}} = \frac{\partial}{\partial x_{nm}} \mathrm{arctan2}\left(\hat{n}_1 \cdot \hat{n}_2,\ \left(\hat{b} \times \hat{n}_1\right) \cdot \hat{n}_2\right)$$
$$= \left(\frac{\partial a}{\partial x_{nm}} \quad \frac{\partial b}{\partial x_{nm}} \quad \frac{\partial c}{\partial x_{nm}}\right) \cdot \left(\frac{\partial}{\partial a} \tau_{ijkl} \quad \frac{\partial}{\partial b} \tau_{ijkl} \quad \frac{\partial}{\partial c} \tau_{ijkl}\right)$$
$$\frac{\partial \tau_{ijkl}}{\partial \alpha} = \left(\left(\hat{b} \times \frac{\partial \hat{n}_1}{\partial \alpha} + \frac{\partial}{\partial \alpha} \hat{b} \times \hat{n}_1\right) \cdot \hat{n}_2 + \frac{\partial \hat{n}_2}{\partial \alpha} \left(\hat{b} \times \hat{n}_1\right)\right) \hat{n}_1 \cdot \hat{n}_2 - \left(\frac{\partial \hat{n}_1}{\partial \alpha} \cdot \hat{n}_2 + \hat{n}_1 \cdot \frac{\partial \hat{n}_2}{\partial \alpha}\right) \left(\hat{b} \times \hat{n}_1\right) \cdot \hat{n}_2$$

Implementation

We expect a set of Cartesian coordinates and will take derivatives of the Z-matrix coordinates with respect to them

Distances

In[771]:=
```
distanceDeriv[i_Integer, j_Integer, n_Integer][coords_] :=
  If[j ≠ n && i ≠ n,
   {0., 0., 0.},
   Module[{vec = coords[[j]] - coords[[i]]},
    If[i == n, -1, 1] * Normalize[vec]
   ]
  ];
distanceDeriv[i_Integer, j_Integer, n_Integer, m_Integer][coords_] :=
  distanceDeriv[i, j, n][coords][[m]];
```

Angles

In[773]:=
```
angleDeriv[i_Integer, j_Integer, k_Integer, n_Integer][coords_] :=
  If[j ≠ n && i ≠ n && k ≠ n,
   {0., 0., 0.},
   Block[
    {
     a = coords[[j]] - coords[[i]], b = coords[[k]] - coords[[i]],
     s, c,
     dsa, dca,
     dsb, dcb,
     axb, adb,
     na, nb,
     naxb,
     au, bu,
     axbu,
     e3 = LeviCivitaTensor[3]
    },
    axb = Cross[a, b];
    adb = Dot[a, b];
    naxb = Norm[axb]; na = Norm[a]; nb = Norm[b];
    au = a / na; bu = b / nb;
    axbu = axb / naxb;
    c = Dot[a, b] / (na * nb);
    s = naxb / (na * nb);
    If[i == n || j == n,
     dsa = 1 / na (-Dot[e3, bu, axbu] - au * s) * c;
     dca = 1 / na (bu - au * c) * s;
    ];
    If[i == n || k == n,
     dsb = 1 / nb (Dot[e3, au, axbu] - bu * s) * c;
     dcb = 1 / nb (au - bu * c) * s;
    ];
    Which[
     i == n,
     -(dsa + dsb - dca - dcb),
     j == n,
     (dsa - dca),
     k == n,
     (dsb - dcb)
    ]
   ]
  ];
```

```
angleDeriv[i_Integer, j_Integer, k_Integer, n_Integer, m_Integer][coords_] :=
  angleDeriv[i, j, k, n][[m]];
```

Torsions

```
In[2284]:= dihedDeriv[i_Integer, j_Integer, k_Integer, l_Integer, n_Integer][coords_] :=
  If[j ≠ n && i ≠ n && k ≠ n && l ≠ n,
   {0., 0., 0.},
   Block[
    {
     a = coords[[j]] - coords[[i]],
     b = coords[[k]] - coords[[j]],
     c = coords[[l]] - coords[[k]],
     n1, n2,
     e3 = LeviCivitaTensor[3],
     i3 = IdentityMatrix[3],
     axb,
     bxc,
     bu,
     na, nb, nc,
     naxb, nbxc,
     dn1a, dn1b,
     dn2b, dn2c,
     dbu,
     i3n1, i3n2,
     n1xb,
     dta, dtb, dtc
    },
    axb = Cross[a, b]; bxc = Cross[b, c];
    na = Norm[a]; nb = Norm[b]; nc = Norm[c];
    naxb = Norm[axb]; nbxc = Norm[bxc];
    n1 = axb / naxb; n2 = bxc / nbxc; bu = b / nb;
    i3n1 = i3 - Outer[Times, n1, n1];
    i3n2 = i3 - Outer[Times, n2, n2];
    (* now we just compute the elements we need to,
    based on what will be non-zero from the Kronecker delta structure*)
    If[i == n || j == n || k == n, dn1a = -Dot[e3, b, i3n1] / naxb];
    dn1b = Dot[e3, a, i3n1] / naxb;
    dn2b = -Dot[e3, c, i3n2] / nbxc;
    If[j == n || k == n || l == n, dn2c = Dot[e3, b, i3n2] / nbxc];
    dbu = 1 / nb (i3 - Outer[Times, bu, bu]);
    n1xb = Cross[bu, n1];
    If[i == n || j == n,
     dta =
```

```
      Dot[
         Map[Cross[bu, #] &, dn1a], n2] * Dot[n1, n2] -
       Dot[dn1a, n2] * Dot[n1xb , n2]
    ];
    If[j == n || k == n,
     dtb = (
          Dot[
            MapThread[
             Cross[bu, #] + Cross[#2, n1] &,
             {dn1b, dbu}
            ],
            n2
           ] + Dot[dn2b, n1xb]
        ) * Dot[n1, n2] - (Dot[dn1b, n2] + Dot[dn2b, n1]) * Dot[n1xb , n2];
    ];
    If[k == n || l == n,
     dtc =
      Dot[dn2c, n1xb] * Dot[n1, n2] - Dot[dn2c, n1] * Dot[n1xb , n2]
    ];
    Which[
     i == n,
     -dta,
     j == n,
     dta - dtb,
     k == n,
     dtb - dtc,
     l == n,
     dtc
    ]
   ]
  ];
dihedDeriv[i_Integer, j_Integer,
    k_Integer, l_Integer, n_Integer, m_Integer][coords_] :=
  dihedDeriv[i, j, k, l, n][[m]];
```

Tests

Input Data

Correct Answer

Distances

Angles

Diheds

Test Values