Quaternion Applications

Part II

Tubing, Bioinformatics, and Dual Quaternions

OUTLINE

- **Quaternion Curves and Tubing:** generalize the Frenet Frame, make quaternion map of *all tubings*, optimize for any tubing task.
- **Quaternions in Bioinformatics:** use quaternion frames to create GLOBAL orientation descriptions and statistics for any protein's amino acid structure.
- **Dual Quaternions:** Introduction to a generalization of quaternions that supports translation as well as rotation.

Part II a: Tubing

What Do Quaternions Have to do with Tubing??

∗ **Basic Idea:** Every point on a curve can be assigned a frame – sort of like a roller-coaster car running on a roller-coaster track.

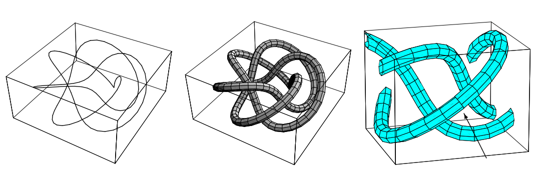
∗ We **FIX one direction**, generally the **tangent to the curve.**

∗ The remaining two directions **define a swept-out tube** (which can have any cross-section you like, typically a circle).

What are Frames used For?

- Our application: Attach tubes and textures to thickened lines.
- ...also... Move objects and object parts in an animated scene.
- Move the camera generating the rendered viewpoint of the scene.
- Compare shapes of similar curves.
- Collect orientation data of moving object (e.g., a joint), etc. etc.

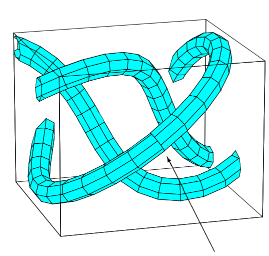
Examine Framing of Curves



The (3,5) torus knot.

- Line drawing ≈ useless.
- Tubing using **parallel transport**: **nice, but not periodic.**
- Closeup of the non-periodic mismatch.

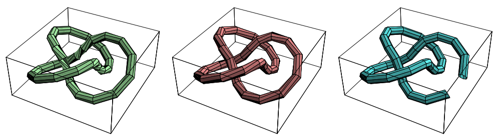
Example of Tubing Problems on Curves



Closeup of the non-periodic mismatch.

Can't apply texture.

More Tubing Issues on Curves...



Tubings of the 2,3 torus knot based on Frenet-Serret, Geodesic Reference, and Parallel Transport frames. Issues: FS: singular, excess twist. GR: singular point, PT: non-periodic.

General Solution: Invariant Quaternion Frames

REMARKS:

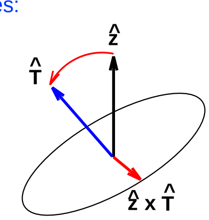
- **Ambiguity of Frame.** We have freedom to choose a "gauge," i.e., any additional rotation around tangent vector, at *any* curve point.
- **Circles in q space.** "Gauge freedom" generates *great circles* in S^3 quaternion space. Need 4π radians to get full quaternion circle.
- **Gauge-invariant swept tube.** Sweeping entire set of circles ($\approx$ dual to tangent vector) in q-space gives *invariant picture of ALL frame possibilities*.
- **Best paths in tube.** Minimal length in S^3 is PT frame! Other choices include minimal acceleration, constant rotation, etc. ...

Geometric Construction of Space of Frames:

- $R(\theta, \hat{\mathbf{T}})$ leaves $\hat{\mathbf{T}}$ invariant, but doesn't have $\hat{\mathbf{T}}$ as Last Column.
- Use Geodesic Reference to construct one instance of such a frame: $R(\hat{\mathbf{z}} \cdot \hat{\mathbf{T}}, \hat{\mathbf{z}} \times \hat{\mathbf{T}})$.

Geometric Construction of Space of Frames:

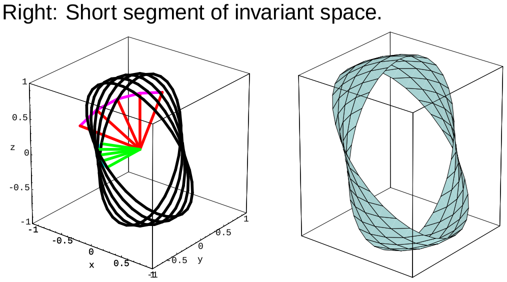
$q(\theta, \hat{\mathbf{T}}) * q(\hat{\mathbf{z}} \cdot \hat{\mathbf{T}}, \hat{\mathbf{z}} \times \hat{\mathbf{T}})$ generates the correct family of quaternion curves:



Invariant Quaternion Frames ...

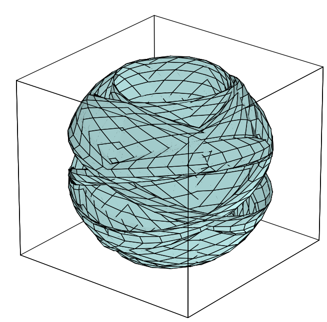
Invariant frame for trefoil knot:

∗ Left: Red fan = tangents; Magenta arc = tangent map; Green vectors = geodesic reference starting points.
∗ Right: Short segment of invariant space.



Invariant Quaternion Frames ...

The Whole Tubing Frame Space of the (2,3) Torus Knot!



3D Curves: Frenet and PT Frames

Now give more details of 3D frames: Classic Moving Frame:

$$\begin{bmatrix}\mathbf{T}'(t)\\ \mathbf{N}'(t)\\ \mathbf{B}'(t)\end{bmatrix} = \begin{bmatrix}0 & k_1(t) & k_2(t)\\ -k_1(t) & 0 & \sigma(t)\\ -k_2(t) & -\sigma(t) & 0\end{bmatrix}\begin{bmatrix}\mathbf{T}(t)\\ \mathbf{N}(t)\\ \mathbf{B}(t)\end{bmatrix}.$$

Serret-Frenet frame: $k_2 = 0$, $k_1 = \kappa(t)$ is the curvature, and $\sigma(t) = \tau(t)$ is the classical torsion. LOCAL.

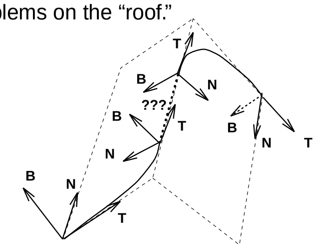
Parallel Transport frame (Bishop): $\sigma = 0$ to get minimal turning. NON-LOCAL = an INTEGRAL.

3D curve frames, contd

Frenet frame is *locally* defined, e.g., by

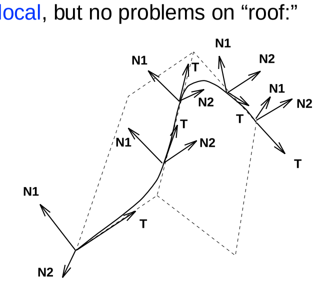
$$\mathrm{B}(t) = \frac{\mathbf{x}'(t)\times\mathbf{x}''(t)}{\|\mathbf{x}'(t)\times\mathbf{x}''(t)\|}$$

but has problems on the "roof."

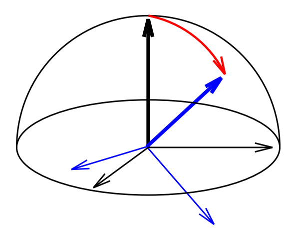


3D curve frames, contd

Bishop's Parallel Transport frame is *integrated over whole curve*, non-local, but no problems on "roof:"



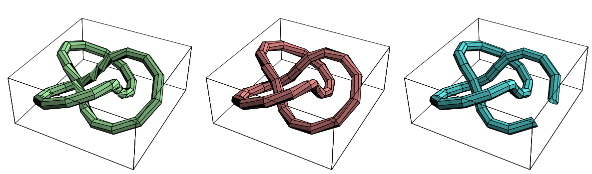
3D curve frames, contd



Geodesic Reference Frame is the frame found by tilting North Pole of "canonical frame" along a great circle until it points in desired direction (tangent for curves, normal for surfaces).

MAIN VALUE: A foolproof reference frame for *sliding rings*.

Sample Curve Tubings and their Frames



Frenet Geodesic Reference Parallel Transport

Easily see PT has least "Twist,"but lacks periodicity.

Conclusion: Quaternion Tubing

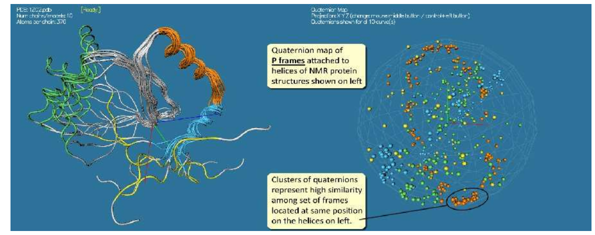
Observations:

- **Tubing and Quaternion Frame Space.** Any path of frames on this space can be used to solve the *tubing problem*.
- **Minimality.** The PT frame appears to be unique frame with *minimum total rotation*.
- **Distributed Twist.** A conventional compromise distributes a user-desired boundary twist uniformly across vertex frames: This is best done using uniform Quaternion distances between *uniformly spatially sampled* frames.
- **Hybrids.** On *closed curves*, Frenet frame is periodic, PT is not. Add fixed angular increment throughout to make PT periodic.
- **Initial angular velocity.** Can give the frame an arbitrary number of twists using $\sigma \neq 0$. Minimal tangential acceleration version corresponds to quaternion treatment by Barr, Currin, Gabriel, and Hughes (Siggraph 92).

PART II b: Quaternion Protein Frames

- **AMINO ACIDS in proteins are oriented structures.**
- **Exactly HOW they are oriented of great biological interest. Usual Ramachandran-frame method is local. Thus one cannot measure global orientation similarities or statistics.**
- **Quaternions fix this — global similarities can be displayed.**

Example: Quaternion Protein Frame Statistics



Quaternion maps for NMR data describing 10 different observed geometries for the protein YvyC from Bacillus subtilis, 2HC5. (left) The collection of alternative geometries. (right) Quaternion maps showing the *orientation space* geometry spreads for each individual amino acid.

Basic Background: Orientation Frames in Bioinformatics

- **Proteins are important.** The entire machinery of life depends on the geometry of proteins, which control the chemical reactions of metabolism.
- **Proteins are long chains of frames.** Proteins consist of hundreds, or even thousands, of amino acids with well-defined orientation frames arranged in a sequence, but with very complicated 3D geometry.
- **Traditional orientation tools describing proteins are primitive.** The Ramachandran plot relates amino acid n to amino acid $n \pm 1$ — that's it!
- **Ramachandran statistics are impossible.** With only local information, you can't compare distant active sites, or gather statistics on non-rigid protein orientation distribution.

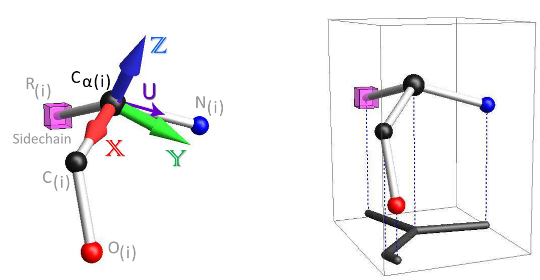
New Progress: Quaternion Frames in Proteomics

- **The PDB has massive protein geometry data.** We can mine that data to construct precise, *amino-acid-residue by amino-acid-residue*, orientation frame labels.
- **Amino acid quaternion frames.** It is straightforward to convert the PDB geometry to quaternion frame sequences.
- **Using our quaternion display tricks, global information about residue alignment is directly visualizable.**
- **Our just-published JMGM paper applies quaternions to many proteomics problems.** For additional information, see A. Hanson and S. Thakur, *Journal of Molecular Graphics and Modelling*, "Quaternion Maps of Global Protein Structure." (Fall 2012).

Basic Procedure

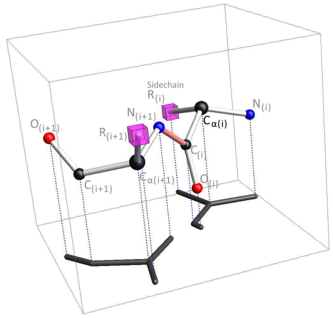
- **Library of 20 amino acids.** Proteins link these together with *peptide bonds*: a C'–OH unit on one end sees an NH_2–C_α on the other side, and joins together as C'–NH–C_α, kicking off a water, H_2O.
- **Pick Three Atoms.** Any three noncollinear atoms are sufficient to define a quaternion frame, but some are more useful for specific purposes than others.
- **Compute Quaternion Frames for the whole protein.**
- **View frame sequence on the quaternion sphere.** Global comparisons as well as local comparisons can be made with a sequence of quaternion frames.
- **Study the map.** The map itself can be used to perform orientation statistics and similarities unobtainable by other methods.

Basis of an Amino Acid Orientation Frame



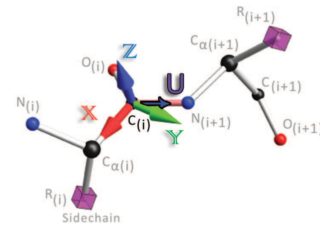
Amino acid geometry showing the computation of our default discrete frame based on the direction from the C_α to the neighboring C and N atoms. The frame vectors $\mathbb{X}$ (red), $\mathbb{Y}$ (green), and $\mathbb{Z}$ (blue) are superimposed on the basic amino acid unit structure.

Amino Acid Orientation Frames for Neighbors



Drop shadow geometry for two adjacent residues. C-N peptide bond is in orange tint. C_α-frames are defined for distinct residues, alternative P-frame includes the linking peptide bond.

Basis of the Amino Acid P-frame



The coordinates of the P-frame definition; the frame centered on the C carbon, and extending to the nitrogen on the neighboring residue.

Basic Geometric Structures:

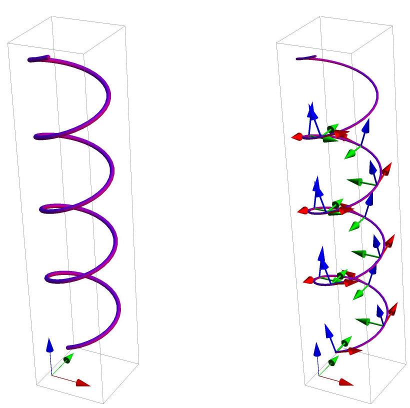
- **Alpha Helix.** One of the most common structures is the Alpha Helix, formed when sequences of residues relax into a low-energy state that coils them into a spiral.
- **Beta Sheet.** Another common structure is essentially sequence of residues related to each other by 180-degree flips, giving the geometric appearance of a “sheet” — really a very flat ellipse.

Model of an Alpha Helix

(a) A helix defined by the parametric equation

$$(r\cos(t),\, r\sin(t),\, pt)\ .$$

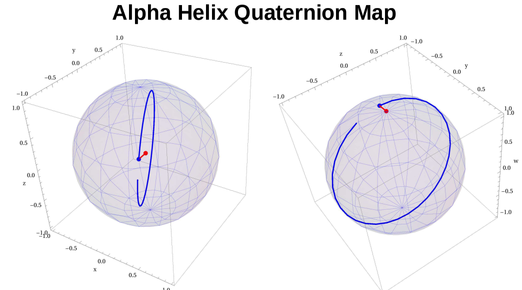
(b) A set of frames on the helical curve defined by the Frenet-Serret equation. Note the relation of the identity frame at bottom left to the first actual helix frame.



(a) (b)

Model of an Alpha Helix

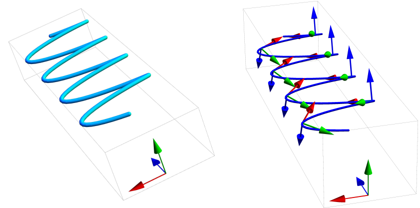
Alpha Helix Quaternion Map



(a) (b)

The quaternion maps for a helix defined by the parametric equation $(r\cos(t),\, r\sin(t),\, pt)$. (a) xyz map. (b) wyz map. Red dot is the identity frame.

Beta Sheet Model



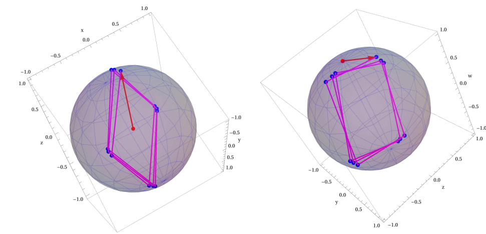
(a) (b)

(a) A beta sheet modeled by the parametric equation

$$(\cos(t),\, 0.1\sin(t),\, 0.5t)$$

(b) A set of Frenet-Serret frames at roughly the expected places on the equation of the curve. Note the relation of the identity frame at foreground to the first actual sampled frame.

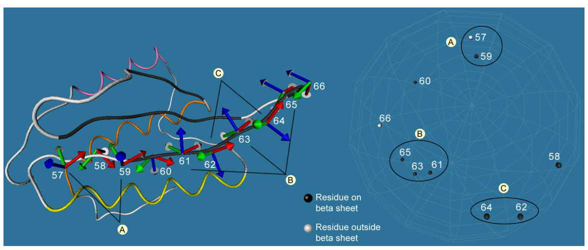
Beta Sheet Quaternion Map



(a) (b)

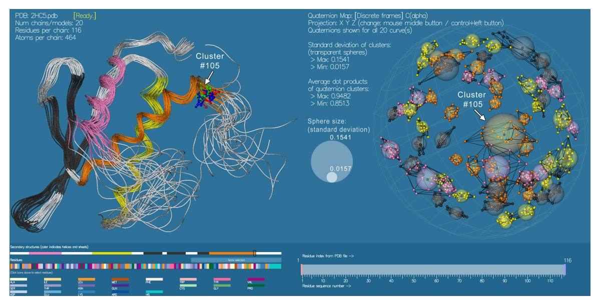
A beta sheet modeled by the parametric equation $(\cos(t),\, 0.1\sin(t),\, 0.5t)$. (a) xyz map. (b) wyz map. Red dot is identity frame.

Example: Beta Sheet Quaternion Map



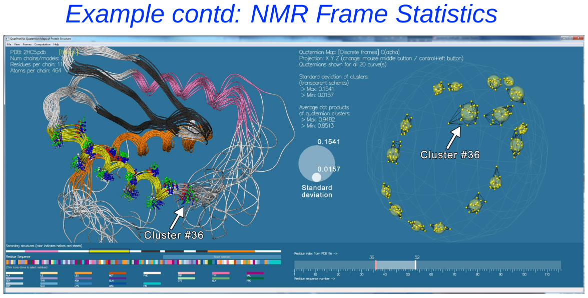
Protein structure of 2HC5 and a quaternion map of its beta sheet structure. Neighboring frames are given matching quaternion signs in this map.

Example: Quaternion NMR Frame Statistics



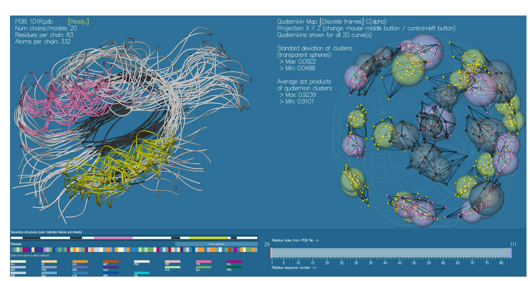
Quaternion maps for NMR data describing 10 deformations of YvyC. (left) Spatial geometries. (right) Quaternion *orientation space* geometry spreads for each amino acid residue.

Example contd: NMR Frame Statistics



Isolating a selected section of the protein YvyC from Bacillus subtilis, 2HC5. (left) The selected region. (right) Quaternion maps showing the *orientation space* geometry spreads for each individual amino acid in this region.

Example contd: NMR Frame Statistics



(a) (b)

Quaternion maps for NMR data describing 20 different geometries for the protein obtained from 1D1R (YciH gene of E. Coli). (a) Alternative geometries. (b) Quaternion map clusters.

Quaternion Protein Maps: Summary and Conclusions

- **Step I: Select a framing.**
- **Step II: Convert to quaternions.**
- **Step III: Enforce Continuity.**
- **Step IV: View the 4D map projected to 3D.** The map itself can be rotated in 4D to different viewpoints that expose selected properties of the similarity space.

Part II c: Dual Quaternions

- **Quaternions Describe Only 3D Rotations.** A computer graphics scene must place elements using both Rotations and Translations.
- **Worse: Quaternion "Vector" Is Wrong.** The "vector" $\boxed{(0, \mathbf{x})}$ in $R \cdot \mathbf{x} = q \star (0, \mathbf{x}) \star q*$ is a 180-degree *rotation*, not a vector (1989 paper by Altmann). $(0, \mathbf{x})$ in pure quaternions *does not result from any sensible translation-like transformation.*

Dual Quaternions...

- **Solution:Dual Quaternions Can Do 3D Translations.** Dual quaternions are a mathematical trick that effectively creates an **infinite-radius rotation**, and that is exactly a translation.
- **Mathematical device: dual numbers.** We already know that quaternions use a "generalized complex number" with $i^2 = j^2 = k^2 = ijk = -1$: Dual numbers add another copy of a quaternion multiplied by $\boxed{\epsilon}$, where $\epsilon^2 = 0$.

Dual Quaternions...

- **Brief History.** Dual quaternions (biquaternions) were first investigated by Clifford (1873), and elaborated by Study (1891). Modern treatments can be found, e.g., from the German school of Blaschke (1960), and are used in theoretical mechanics (Bottema and Roth, 1979; McCarthy, 1990), and in robotics. See also See Dorst et al., *Geometric Algebra for Computer Science* for the connection between dual quaternions and Clifford algebras.
- **Resources for Graphics Applications.** Kavan et al. (TOG, 2008) have spurred the transfer of dual quaternion methods from robotics to graphics for skinning problems, etc. (Appendix of Kavan et al. has an excellent summary, but misses a couple of fine points that we will look at below.)

Approach to Adding in Translations

IDEA: ***Terminate*** **the exponential series.** This changes a rotation into a translation.

$$\begin{aligned}
\text{Usual: } i^2 = -1\text{: } e^{i\theta} &= 1 + i\theta - \frac{1}{2}\theta^2 - \frac{1}{3!}i\theta^3 + \cdots \\
&= \cos\theta + i\sin\theta \\
e^{i\theta}e^{i\phi} &= \cos(\theta+\phi) + i\sin(\theta+\phi) \\
\text{Dual: } \epsilon^2 = 0\text{: } e^{\epsilon t} &= 1 + \epsilon t + 0 \\
e^{\epsilon x}e^{\epsilon t} &= 1 + \epsilon(x+t)\ .
\end{aligned}$$

Like the $\theta \to 0$ limit of rotation. So the dual algebra looks like *small* θ or *large radius* rotation, which is effectively *translation*.

...toward Dual Quaternions...

New look for 3D Vectors: The key to removing Altmann's objection is surprising:

- Replace the *zero* in $(0, \mathbf{x})$ by a *one*.
- Then multiply the $\mathbf{x}$ by ϵ.
- **This gives us a way to make a True Vector.**

$$\begin{aligned}
\mathbf{x} &= (1, 0) + \epsilon(0, \mathbf{x}) \\
&\equiv (1, \epsilon\mathbf{x})\ .
\end{aligned}$$

- **Now we can make a vector from nothing** using quaternion conjugation:

$$(1, \epsilon\mathbf{x}) = (1, \epsilon\mathbf{x}/2) \star (1, \epsilon 0) \star (1, \epsilon\mathbf{x}/2)\ .$$

Dual Quaternions...

- **Dual translation features:** We next pick a notation for the dual quaternion for translation: *Note* that we need a special *conjugation*: $(q_0, \mathbf{q})^* = (q_0, -\mathbf{q})$ combined with *dual conjugation* $\overline{a + \epsilon b} = a - \epsilon b$. Need *both* to give needed **plus sign** in sandwiched translation.
- **Dual translation $\tau(\mathbf{d})$:** Our new tool (with half-vectors) is the dual quaternion
$$\tau(\mathbf{d}) = (1, 0) + \epsilon\,(0, \frac{\mathbf{d}}{2}) \equiv (1, \epsilon\,\frac{\mathbf{d}}{2})$$
- **Translate $\mathbf{x}$ to $\mathbf{x} + \mathbf{d}$** by conjugate multiplication:
$$\tau(\mathbf{d}) \star (1, \epsilon\mathbf{x}) \star \overline{\tau(\mathbf{d})^*} = (1, \epsilon\,(\mathbf{x} + \mathbf{d}))$$

Dual Quaternions Fix Hamilton

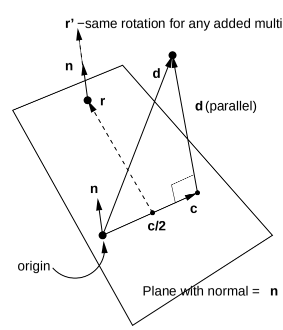
- **Rotation done right!** SAME answer, of course, but now $\mathbf{x}$ is *no longer* confounded with 180^o rotation:
$$(1, \epsilon R \cdot \mathbf{x}) = q \star (1, \epsilon\mathbf{x}) \star \overline{q^*}$$
- **Full $\mathrm{SE}(3)$ frame now possible!** We can perform a complete OpenGL-style transformation, $\mathbf{x}' = T \cdot R \cdot \mathbf{x} = R \cdot \mathbf{x} + \mathbf{d}$, as:
$$\tau(\mathbf{d}) \star q \star (1, \epsilon\mathbf{x}) \star \overline{(\tau(\mathbf{d}) \star q)^*} = (1, \epsilon(R \cdot \mathbf{x} + \mathbf{d}))$$
- *However, this is only half the story.*

Moving Centers with Dual Quaternions

Problem: what happened to (cos, $\mathbf{n}$ sin)**?** Don't we want to use that representation for the FULL frame?

- **First step:** $T(r) \cdot R \cdot T^{-1}(r)$**.** What happens to the *fixed-point rule*?
$$\tau(\mathbf{r}) \star \left(\cos\frac{\theta}{2}, \mathbf{n}\sin\frac{\theta}{2}\right) \star \tau(-\mathbf{r})$$
$$= \left(\cos\frac{\theta}{2}, (\mathbf{n} + \epsilon\,\mathbf{r} \times \mathbf{n})\sin\frac{\theta}{2}\right)$$
- **So the fixed point** $\boxed{\mathbf{r}}$ **appears as a dual rotation axis** $\boxed{\mathbf{r} \times \mathbf{n}}$, automatically *in the plane perpendicular to* $\mathbf{n}$.

Figure: Displacing Rotations



Fixed Point Rule of an arbitrary displacement: **result is unchanged by moving $\mathbf{r}$ in the $\mathbf{n}$ direction.**

Moving Centers . . .

Problem: $\mathbf{r} \equiv \mathbf{r} + t\mathbf{n}$: How do you move out of the $\mathbf{n}_\perp$ plane?

- **Second step: dual angle.** Now suppose $\mathbf{r} = 0$, but make *angle* dual: $\theta \to \theta + \epsilon\lambda$.
- **Dual trig formulas:** We need $\cos(a + \epsilon\,b) = \cos a - \epsilon\,b \sin a$ and $\sin(a + \epsilon\,b) = \sin a + \epsilon\,b \cos a$, which follow from the Taylor series expansion. Then
$$q(\theta + \epsilon\lambda, \mathbf{n}) = \left(\cos\frac{\theta}{2} - \epsilon\frac{\lambda}{2}\sin\frac{\theta}{2},\ \mathbf{n}(\sin\frac{\theta}{2} + \epsilon\frac{\lambda}{2}\cos\frac{\theta}{2})\right)$$

Moving Centers . . .

- $\theta \to 0$ **limit:** Simplest case is vanishing θ, so
$$q(\epsilon\lambda, \mathbf{n}) = (1, \epsilon\,\mathbf{n}\frac{\lambda}{2})$$
- In other words, *the dual angle is the displacement along* $\mathbf{n}$*!*
- **Split displacement** $\mathbf{d}$ into $\mathbf{d}_\perp$ and $\mathbf{d}_\parallel$ relative to $\mathbf{n}$ using $\lambda = \mathbf{n} \cdot \mathbf{d}$, so that:
$$\mathbf{d} = \mathbf{d}_\perp + \mathbf{d}_\parallel = \mathbf{c} + \lambda\mathbf{n}$$
and by definition $\boxed{\mathbf{c} = \mathbf{d} - \lambda\mathbf{n}}$ and $\boxed{\mathbf{n}}$ are the *Plucker coordinates* of the line through $\mathbf{d}$ parallel to $\mathbf{n}$.

Screw Motion: Chasles' Theorem

Last step: find the $\boxed{\textbf{rotation center } \mathbf{r}}$ that makes the equations work:

- We can easily solve the equations

$$\boxed{T(\mathbf{r}) \cdot R \cdot T(-\mathbf{r}) = T(\mathbf{d}_\perp = \mathbf{c}) \cdot R}$$

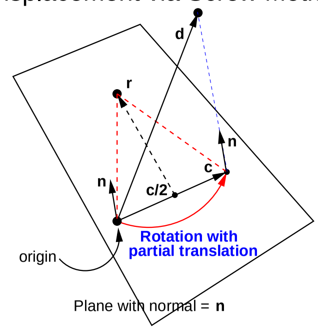
assuming $\mathbf{r}$ is perpendicular to $\mathbf{n}$, so $\lambda = \mathbf{n} \cdot \mathbf{d}$ is the *whole* component of $\mathbf{d}$ in the $\mathbf{n}$ direction.

- We find (remember $\mathbf{c} = \mathbf{d}_\perp = \mathbf{d} - \lambda\mathbf{n}$):

$$\boxed{\mathbf{r} = \tfrac{1}{2}\mathbf{c} + \tfrac{1}{2}\mathbf{n} \times \mathbf{c} \cot\tfrac{\theta}{2}}$$

unless $\theta = 0$, in which case the action becomes a pure translation. (This is part of **Chasles' Theorem**.)

Displacement via Screw Motion



Rotation can produce Displacement. But you can't use Fixed Point Rule for *arbitrary* displacement. Must **ADD SEPARATELY** any motion along the $\mathbf{n}$ direction.

Full Frame with Unit Dual Quaternion

Finally, we can see how to decompose $T(\mathbf{d}) \cdot R$ into a *unit dual trigonometric quaternion* $\hat{q}$. Note that $\|\hat{q} \star \hat{p}\| = \|\hat{q}\|\|\hat{p}\|$, so the unit property is preserved under *any* dual quaternion multiplication.

General trig form for dual quaternion $\boxed{\hat{q}}$:

$$\begin{aligned}
\hat{q}(\hat{\theta}, \hat{\mathbf{n}}) &= q_0 + \epsilon q_\epsilon \\
&= (q_0 + \epsilon q_\epsilon,\, \mathbf{q}_0 + \epsilon \mathbf{q}_\epsilon) \\
&= \left(\cos\frac{\theta + \epsilon\lambda}{2}, (\mathbf{n} + \epsilon\mathbf{r} \times \mathbf{n}) \sin\frac{\theta + \epsilon\lambda}{2}\right)
\end{aligned}$$

Full Frame with Unit Dual Quaternion . . .

- **Dual Quat Norm:** Easy to show that our norm is

$$\|\hat{q}\|^2 = \|q_0\|^2 + 2\epsilon q_0 \cdot q_\epsilon \,.$$

- **So a unit dual quaternion must have** $q_0 \cdot q_\epsilon = 0$**.** That is satisfied by the trigonometric form because $\mathbf{n} \cdot (\mathbf{r} \times \mathbf{n}) = 0$.

- Thus we can always parameterize $T(\mathbf{d}) \cdot R(\theta, \mathbf{n})$ using the unit dual quaternion

$$\hat{q}\left(\hat{\theta} = \theta + \epsilon(\mathbf{d} \cdot \mathbf{n}),\, \hat{\mathbf{n}} = \mathbf{n} + \epsilon\mathbf{r} \times \mathbf{n}\right),$$

where $\mathbf{r}$ is the rotation center, computable from θ, $\mathbf{n}$, and $\mathbf{d}$ as $\mathbf{r} = \frac{1}{2}\mathbf{c} + \frac{1}{2}\mathbf{n} \times \mathbf{c} \cot\frac{\theta}{2}$.

Interpolating Dual Quaternions

- **SLERP extends directly:** Using power series if needed, easily extend the usual quaternion interpolation formulas:

$$\hat{q}(t) = \cos t\frac{\hat{\theta}}{2} + \hat{n} \sin t\frac{\hat{\theta}}{2}$$

where, e.g., $\hat{n} = \mathbf{n} + \epsilon\,\mathbf{r} \times \mathbf{n}$, $\hat{\theta} = \theta + \epsilon\,\mathbf{d} \cdot \mathbf{n}$.

- **WITH one small problem:** The "rotation-induced" translation along $\mathbf{c} \perp \mathbf{n}$ has a *different speed* from the translation along $\mathbf{n}$ (the $\lambda = (\mathbf{d} \cdot \mathbf{n})$ part). **PLUS** the translation is *not straight* (see Screw Motion Figure).

This is the cost of the dual trigonometric form.

Useful Properties of Dual Quaternions. . .

- **Exponential and Log:** Using power series, one can extend the usual quaternion exponential and log formulas:

$$\exp(\hat{s}\hat{\theta}) = \cos\tfrac{\hat{\theta}}{2} + \hat{s}\sin\tfrac{\hat{\theta}}{2}$$

and so obviously $\log\hat{q} = \hat{s}\,\hat{\theta}$.

- **Inverse:** The inverse of a dual object is defined only for $a \neq 0$:

$$(a + \epsilon\, b)^{-1} = \tfrac{1}{a+\epsilon\, b} = \tfrac{1}{a} - \epsilon\,\tfrac{b}{a^2}$$

as can be easily confirmed from $(a+\epsilon\, b)(c+\epsilon\, d) = ac+\epsilon\,(ad+bc)$

- **Square Root:** A trick similar to the one we saw for quaternions works for the square root of a dual quaternion (again, for $a \neq 0$):

$$\sqrt{a + \epsilon\, b} = \sqrt{a} + \epsilon\,\tfrac{b}{2\sqrt{a}}.$$

Applications: Blending and Interpolation

- **Blending for Skinning:** Dual quaternions permit an unusually smooth combination of weighted skin vertices associated to two or more skeletal elements in character animation. The most rigorous methods are essentially dual quaternion extensions of the spherical center-of-mass methods of Buss and Fillmore (TOG, 2001). Faster, but less accurate methods, use the concept of Phong shading, renormalizing a linear combination of data sets (Kavan et al., TOG 2008).
- **Interpolation:** Blending is a static process, and needs to be done to combine character body elements such as skin vertices at each moment. Interpolation for simulating moving object kinematics and controlling camera motion can also be accomplished by extending standard quaternion interpolation techniques to dual quaternions, though challenging issues such as how to control dual parameters and how to match rotational and translational speeds in a single interpolation introduce additional complexity and possible artifacts.

FINAL TUTORAL SUMMARY

- **Quaternions nicely represent frame sequences.**
- **TUBES: Curve frames ⇒ quaternion curves.** Exploit quaternion space of frames to design any type of frame.
- **PROTEINS: Amino acid residue coordinates ⇒ quaternion frame maps.** Apply to global comparisons and statistical distributions.
- **DUAL QUATERNIONS: (From Clifford, 1873.) Extend quaternion rotation algebra to include translations. Applications include blending for skinning in figure animation, robot arm motion planning, etc.**

ftp://ftp.cs.indiana.edu/pub/hanson/Siggraph12QuatCourse/