Quaternion Applications

Andrew J. Hanson
School of Informatics and Computing
Indiana University

Siggraph Asia 2012 Tutorial

OUTLINE

I: (55 min) **Introduction to Quaternions:**

What are they good for?

Understanding Rotation Sequences!

II a: (15 min) **Quaternion Tubing:**

Visualizing Framed Space Curves

...OUTLINE...

II b: (15 min) **Quaternion Protein Maps:**

Amino Acid Frame Sequences with Quaternions

II c: (20 min) **Intro to Dual Quaternions:**

Applications to Six-Degrees-of-Freedom

Part I

Introduction to Quaternions:

...Twisting Belts and Rolling Balls...

Explaining Rotation Sequences with Quaternions

Where Did Quaternions Come From?

...from the discovery of *Complex Numbers:*

- $\boxed{z = x + iy}$ Complex numbers = realization that $z^2 + 1 = 0$ cannot be solved unless you have an “imaginary” number with $i^2 = -1$.
- **Euler’s formula**: $e^{i\theta} = \cos\theta + i \sin\theta$ allows you to do most of 2D geometry.

Hamilton

The first to ask *“If you can do 2D geometry with complex numbers, how might you do 3D geometry?”* was William Rowan Hamilton, circa 1840.

Sir William Rowan Hamilton
4 August 1805 — 2 September 1865

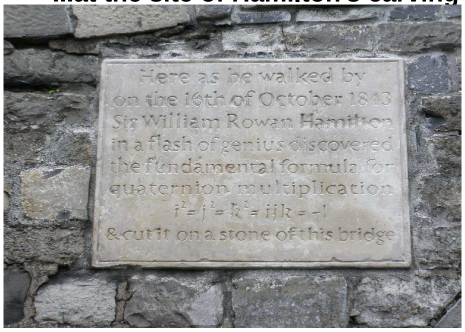
Hamilton's epiphany: 16 October 1843

"An electric circuit seemed to close; and a spark flashed forth . . . Nor could I resist the impulse – unphilosophical as it may have been – to cut with a knife on a stone of Brougham Bridge, as we passed it, the fundamental formula with the symbols, i, j, k; namely,

$$i^2 = j^2 = k^2 = ijk = -1$$

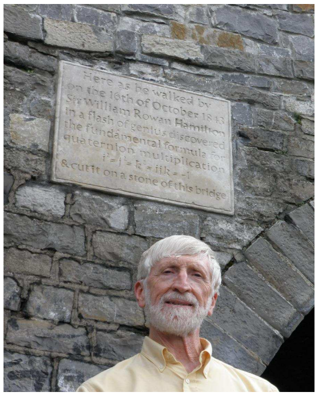
which contains the Solution of the Problem..."

...at the site of Hamilton's carving



The plaque on Broome Bridge in Dublin, Ireland, commemorating the legendary location where Hamilton conceived of the idea of quaternions. *(Photo taken July 2012)*.

...the author on Broome Bridge...



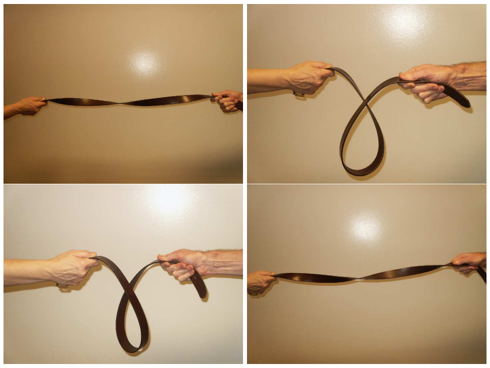
Yes, I have actually been there!

The Belt Trick

Quaternion Geometry in our daily lives

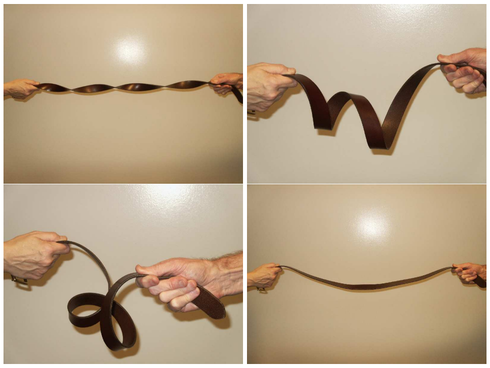
- Two people hold ends of a belt.
- Twist the belt either 360 degrees or 720 degrees.
- **Rule:** *Move belt ends any way you like but do not change orientation of either end.*
- Try to straighten out the belt.

360 Degree Belt



360 twist: stays twisted, can change DIRECTION!

720 Degree Belt



720 twist: CAN FLATTEN OUT WHOLE BELT!

The Beltless Trick

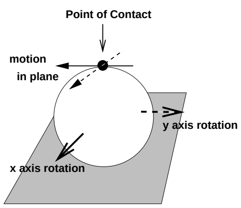
Quaternion Geometry is right in your hand!

- Hold a coffee cup (empty is a good idea) in the palm of your hand.
- Keeping the cup vertical, user your hand to twist the handle, first by 360 degrees (painful).
- *Now CONTINUE another 360 degrees,* for a total of 720 degrees.
- *Your arm is once again STRAIGHT!*

Rolling Ball Puzzle

1. Put a ball on a flat table.
2. Place hand flat on top of the ball
3. Make circular rubbing motion, as though polishing the tabletop.
4. Watch a point on the equator of the ball.
5. *small clockwise circles* $\rightarrow$ **equator goes counterclockwise**
6. *small counterclockwise circles* $\rightarrow$ **equator goes clockwise**

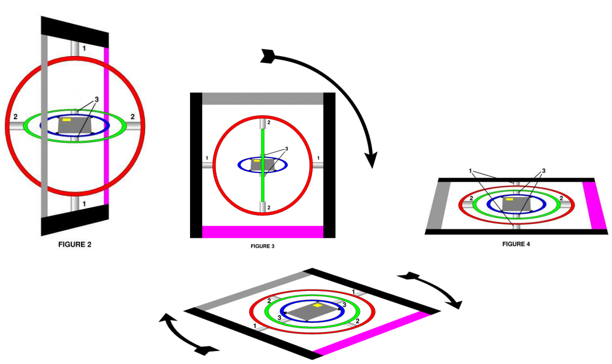
Rolling Ball Scenario



Gimbal Lock

Gimbal Lock occurs when a mechanical or computer system experiences an anomaly due to an (x, y, z)-based orientation control sequence.

- *Mechanical systems cannot avoid all possible gimbal lock situations .*
- *Computer orientation interpolation systems can avoid gimbal-lock-related glitches* **by using quaternion interpolation.**



Mechanical Gimbal Lock: **Using x, y, z axes to encode orientation gives singular situations.**

Gimbal Lock — Apollo Systems

Red-painted area = Danger of real Gimbal Lock

2D Rotations

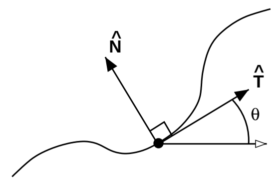
- 2D rotations $\leftrightarrow$ *complex numbers.*
- Why? $e^{i\theta}(x + iy) = (x' + iy')$

$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta$$

- **Complex numbers** are a subspace of quaternions — so exploit 2D rotations to **introduce us to quaternions** and their geometric meaning.

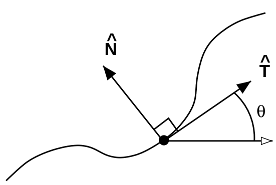
Frames in 2D

The tangent and normal to 2D curve move continuously along the curve:



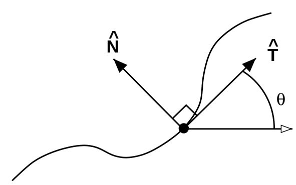
Frames in 2D

The tangent and normal to 2D curve move continuously along the curve:



Frames in 2D

The tangent and normal to 2D curve move continuously along the curve:



Frame Matrix in 2D

This motion is described at each point (or time) by the matrix:

$$R_2(\theta) = \begin{bmatrix} \hat{\mathbf{T}} & \hat{\mathbf{N}} \end{bmatrix}$$
$$= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

The Belt Trick Says:

There is a Problem...at least in 3D

How do you get $\cos\theta$ to know about 720 degrees?

The Belt Trick Says:

There is a Problem...at least in 3D

How do you get $\cos\theta$ to know about 720 degrees?

Hmmmmm. $\cos(\theta/2)$ knows about 720 degrees, right?

Half-Angle Transform:

A Fix for the Problem?

Let $a = \cos(\theta/2)$, $b = \sin(\theta/2)$,
(i.e., $\cos\theta = a^2 - b^2$, $\sin\theta = 2ab$),
and parameterize 2D rotations as:

$$R_2(a,b) = \begin{bmatrix} a^2 - b^2 & -2ab \\ 2ab & a^2 - b^2 \end{bmatrix}.$$

where orthonormality implies

$$(a^2 + b^2)^2 = 1$$

which reduces back to $a^2 + b^2 = 1$.

Frame Evolution in 2D

Examine the time-evolution of a 2D frame (on our way to 3D). First use $\theta(t)$ coordinates:

$$\begin{bmatrix} \hat{\mathbf{T}} & \hat{\mathbf{N}} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

Differentiate to find frame equations:

$$\begin{aligned} \dot{\hat{\mathbf{T}}}(t) &= +\kappa\hat{\mathbf{N}} \\ \dot{\hat{\mathbf{N}}}(t) &= -\kappa\hat{\mathbf{T}}, \end{aligned}$$

where $\kappa(t) = d\theta/dt$ is the **curvature**.

Frame Evolution in (a,b):

The basis $(\hat{\mathbf{T}}, \hat{\mathbf{N}})$ is nasty — **Four equations** with **Three constraints** from orthonormality, but just **One** true degree of freedom.

Major Simplification occurs in (a,b) coordinates!!

$$\dot{\hat{\mathbf{T}}} = 2\begin{bmatrix} a\dot{a} - b\dot{b} \\ a\dot{b} + b\dot{a} \end{bmatrix} = 2\begin{bmatrix} a & -b \\ b & a \end{bmatrix}\begin{bmatrix} \dot{a} \\ \dot{b} \end{bmatrix}$$

Frame Evolution in (a,b):

But this formula for $\dot{\hat{\mathbf{T}}}$ is just $\kappa\hat{\mathbf{N}}$, where

$$\kappa\hat{\mathbf{N}} = \kappa\begin{bmatrix} -2ab \\ a^2 - b^2 \end{bmatrix} = \kappa\begin{bmatrix} a & -b \\ b & a \end{bmatrix}\begin{bmatrix} -b \\ a \end{bmatrix}$$

or

$$\kappa\hat{\mathbf{N}} = \kappa\begin{bmatrix} a & -b \\ b & a \end{bmatrix}\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix}$$

2D Quaternion Frames!

Rearranging terms, *both* $\dot{\hat{\mathbf{T}}}$ and $\dot{\hat{\mathbf{N}}}$ eqns reduce to

$$\begin{bmatrix} \dot{a} \\ \dot{b} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & -\kappa \\ +\kappa & 0 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix}$$

This is the square root *of* frame equations.

2D Quaternions . . .

So *one equation* in the two *"quaternion"* variables (a, b) with the constraint $a^2+b^2 = 1$ contains *both* the frame equations

$$\dot{\hat{\mathbf{T}}} = +\kappa \hat{\mathbf{N}}$$

$$\dot{\hat{\mathbf{N}}} = -\kappa \hat{\mathbf{T}}$$

$\Rightarrow$ this is much better for computer implementation, etc.

Rotation as Complex Multiplication

If we let $(a + ib) = \exp{(i\,\theta/2)}$ we see that rotation is complex multiplication!

"Quaternion Frames" in 2D are just complex numbers, with

Evolution Eqns = derivative of $\exp{(i\,\theta/2)}$!

Rotation with no matrices!

Due to an extremely deep reason in Clifford Algebras,

$$a + ib = e^{i\theta/2}$$

represents rotations "more nicely" than the matrices $R(\theta)$.

$$(a' + ib')(a + ib) = e^{i(\theta'+\theta)/2} = A + iB$$

where if we *want* the matrix, we write:

$$R(\theta')R(\theta) = R(\theta' + \theta) = \begin{bmatrix} A^2 - B^2 & -2AB \\ 2AB & A^2 - B^2 \end{bmatrix}$$

The Algebra of 2D Rotations

The algebra corresponding to 2D rotations is easy: just complex multiplication!!

$$\begin{aligned} (a', b') * (a, b) &\cong (a' + ib')(a + ib) \\ &= a'a - b'b + i(a'b + ab') \\ &\cong (a'a - b'b,\ a'b + ab') \\ &= (A,\ B) \end{aligned}$$

2D Rotations are just **complex multiplication**, and take you around the unit circle!

Quaternion Frames

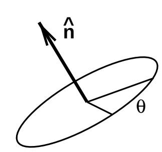
In 3D, repeat our trick: take square root of the frame, but now use *quaternions*:

- **Write down the 3D frame.**
- **Write as double-valued quadratic form.**
- **Rewrite frame evolution equations linearly in the new variables.**

The Geometry of 3D Rotations

We begin with a basic fact:

Euler theorem: *every* 3D frame can be written as a spinning by θ about a fixed axis $\hat{\mathbf{n}}$, the eigenvector of the rotation matrix:



Quaternion Frames . . .

The Matrix $R_3(\theta, \hat{\mathbf{n}})$ giving 3D rotation by θ about axis $\hat{\mathbf{n}}$ is :

$$\begin{bmatrix} c + (n_1)^2(1-c) & n_1 n_2(1-c) - s n_3 & n_3 n_1(1-c) + s n_2 \\ n_1 n_2(1-c) + s n_3 & c + (n_2)^2(1-c) & n_3 n_2(1-c) - s n_1 \\ n_1 n_3(1-c) - s n_2 & n_2 n_3(1-c) + s n_1 & c + (n_3)^2(1-c) \end{bmatrix}$$

where $c = \cos\theta$, $s = \sin\theta$, and $\hat{\mathbf{n}} \cdot \hat{\mathbf{n}} = 1$.

Can we find a 720-degree form?

Remember 2D: $a^2 + b^2 = 1$
then substitute $1 - c = (a^2 + b^2) - (a^2 - b^2) = 2b^2$
to find the remarkable expression for $\mathbf{R}(\theta, \hat{\mathbf{n}})$:

$$\begin{bmatrix} a^2 - b^2 + (n_1)^2(2b^2) & 2b^2 n_1 n_2 - 2ab n_3 & 2b^2 n_3 n_1 + 2ab n_2 \\ 2b^2 n_1 n_2 + 2ab n_3 & a^2 - b^2 + (n_2)^2(2b^2) & 2b^2 n_2 n_3 - 2ab n_1 \\ 2b^2 n_3 n_1 - 2ab n_2 & 2b^2 n_2 n_3 + 2ab n_1 & a^2 - b^2 + (n_3)^2(2b^2) \end{bmatrix}$$

Rotations and Quadratic Polynomials

Remember $(n_1)^2 + (n_2)^2 + (n_3)^2 = 1$ and $a^2 + b^2 = 1$;
letting

$$q_0 = a = \cos(\theta/2) \qquad \mathbf{q} = b\hat{\mathbf{n}} = \hat{\mathbf{n}}\sin(\theta/2)$$

We find a matrix $R_3(q)$

$$\begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1 q_2 - 2q_0 q_3 & 2q_1 q_3 + 2q_0 q_2 \\ 2q_1 q_2 + 2q_0 q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2 q_3 - 2q_0 q_1 \\ 2q_1 q_3 - 2q_0 q_2 & 2q_2 q_3 + 2q_0 q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

Quaternions and Rotations . . .

HOW does $q = (q_0, \mathbf{q})$ represent rotations?

LOOK at

$$R_3(\theta, \hat{\mathbf{n}}) \stackrel{?}{=} R_3(q_0, q_1, q_2, q_3)$$

THEN we can verify that choosing

$$q(\theta, \hat{\mathbf{n}}) = (\cos\frac{\theta}{2}, \ \hat{\mathbf{n}}\sin\frac{\theta}{2})$$

makes the R_3 equation an *IDENTITY*.

Quaternions and Rotations . . .

WHAT happens if you do **TWO** rotations?

EXAMINE the action of two rotations

$$R(q')R(q) = R(Q)$$

EXPRESS in quadratic forms in q and LOOK FOR an analog of complex multiplication:

Quaternions and Rotations . . .

RESULT: the following multiplication rule $q' * q = Q$ yields **exactly** the correct 3×3 rotation matrix $R(Q)$:

$$\begin{bmatrix} Q_0 = [q' * q]_0 \\ Q_1 = [q' * q]_1 \\ Q_2 = [q' * q]_2 \\ Q_3 = [q' * q]_3 \end{bmatrix} = \begin{bmatrix} q'_0 q_0 - q'_1 q_1 - q'_2 q_2 - q'_3 q_3 \\ q'_0 q_1 + q'_1 q_0 + q'_2 q_3 - q'_3 q_2 \\ q'_0 q_2 + q'_2 q_0 + q'_3 q_1 - q'_1 q_3 \\ q'_0 q_3 + q'_3 q_0 + q'_1 q_2 - q'_2 q_1 \end{bmatrix}$$

This is Quaternion Multiplication.

Algebra of Quaternions
= 3D Rotations!

2D rotation matrices are represented by **complex multiplication**

3D rotation matrices are represented by **quaternion multiplication**

Algebraic 2D/3D Rotations

Therefore in 3D, the 2D complex multiplication

$$(a', b') * (a, b) = (a'a - b'b, \; a'b + ab')$$

is replaced by 4D quaternion multiplication:

$$\begin{aligned} q' * q \;=\; & (q'_0 q_0 - q'_1 q_1 - q'_2 q_2 - q'_3 q_3, \\ & q'_0 q_1 + q'_1 q_0 + q'_2 q_3 - q'_3 q_2, \\ & q'_0 q_2 + q'_2 q_0 + q'_3 q_1 - q'_1 q_3, \\ & q'_0 q_3 + q'_3 q_0 + q'_1 q_2 - q'_2 q_1) \end{aligned}$$

Algebra of Quaternions . . .

The equation is easier to remember by dividing it into a *scalar* piece q_0 and a *vector* piece $\vec{q}$:

$$\begin{aligned} q' * q \;=\; & (q'_0 q_0 - \vec{q}' \cdot \vec{q}, \\ & q'_0 \, \vec{q} + q_0 \, \vec{q}' + \vec{q}' \times \vec{q}) \end{aligned}$$

Now we can SEE quaternions!

Since $(q_0)^2 + \mathbf{q} \cdot \mathbf{q} = 1$ then

$$q_0 = \sqrt{1 - \mathbf{q} \cdot \mathbf{q}}$$

Plot just the 3D vector: $\mathbf{q} = (q_x, \, q_y, \, q_z)$

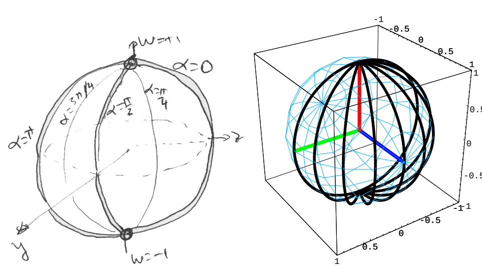
q_0 is KNOWN! We can also use any other triple: the fourth component is *dependent*.

DEMO

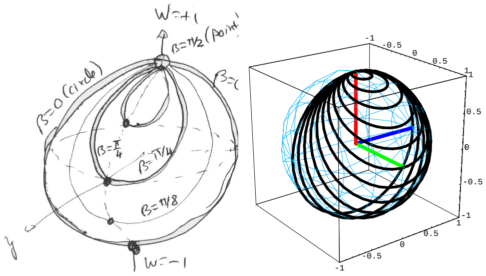
We can now make a Quaternion Picture of each of our favorite tricks

- 360° **Belt Trick in Quaternion Form.** DEMO:
- 720° **Belt Trick in Quaternion Form.**
- **Rolling Ball in Quaternion Form.** DEMO:
- **Gimbal Lock in Quaternion Form.**

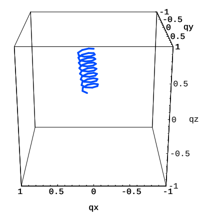
360° Belt Trick in Quaternion Form



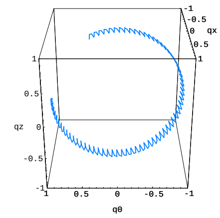
720° **Belt Trick in Quaternion Form**



Rolling Ball in Quaternion Form

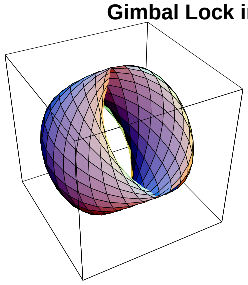


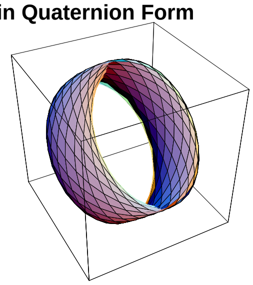
$\mathbf{q}$ vector-only plot.



(q_0, q_x, q_z) plot

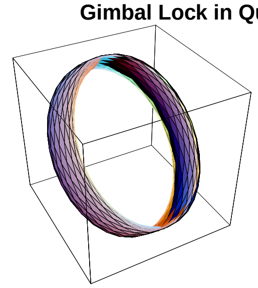
Gimbal Lock in Quaternion Form

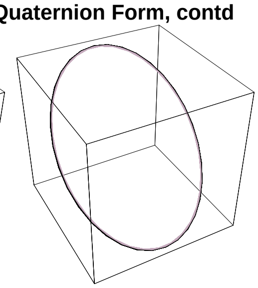




Quaternion Plot of the *remaining* orientation degrees of freedom of $\mathbf{R}(\theta, \hat{\mathbf{x}}) \cdot \mathbf{R}(\phi, \hat{\mathbf{y}}) \cdot \mathbf{R}(\psi, \hat{\mathbf{z}})$ at $\phi = 0$ and $\phi = \pi/6$

Gimbal Lock in Quaternion Form, contd





Choosing ϕ and plotting the *remaining* orientation degrees in the rotation sequence $\mathbf{R}(\theta, \hat{\mathbf{x}}) \cdot \mathbf{R}(\phi, \hat{\mathbf{y}}) \cdot \mathbf{R}(\psi, \hat{\mathbf{z}})$, we see degrees of freedom **decrease from TWO to ONE as** $\phi \rightarrow \pi/2$

Quaternion Interpolations

- Shoemake (Siggraph '85) proposed using quaternions instead of Euler angles to get smooth frame interpolations without **Gimbal Lock**:

  BEST CHOICE: Animate objects and cameras using rotations represented on S^3 by quaternions

Interpolating on Spheres

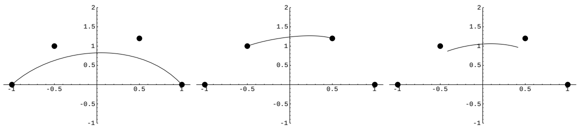
General quaternion spherical interpolation employs the "SLERP," a constant angular velocity transition between two directions, $\hat{\mathrm{q}}_1$ and $\hat{\mathrm{q}}_2$:

$$\begin{aligned} \hat{\mathrm{q}}_{12}(t) &= \mathrm{Slerp}(\hat{\mathrm{q}}_1, \hat{\mathrm{q}}_2, t) \\ &= \hat{\mathrm{q}}_1 \frac{\sin((1-t)\theta)}{\sin(\theta)} + \hat{\mathrm{q}}_2 \frac{\sin(t\theta)}{\sin(\theta)} \end{aligned}$$

where $\cos\theta = \hat{\mathrm{q}}_1 \cdot \hat{\mathrm{q}}_2$.

Plane Interpolations

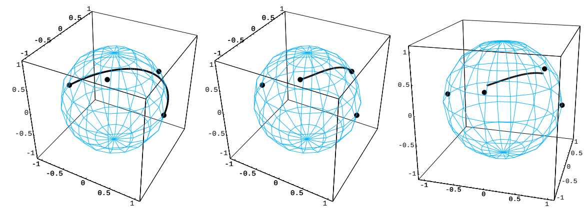
In Euclidean space, these three basic cubic splines look like this:



Bezier Catmull-Rom Uniform B

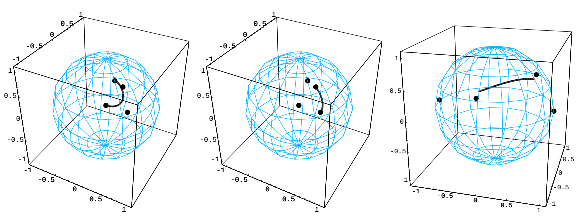
The differences are in the derivatives: Bezier has to start matching all over at every fourth point; Catmull-Rom matches the first derivative; and B-spline is the cadillac, matching all derivatives but *no control points*.

Spherical Interpolations



Bezier Catmull-Rom Uniform B

Quaternion Interpolations



Bezier Catmull-Rom Uniform B

Exp Form of Quaternion Rotations

In Hamilton's notation, we can generalize the 2D equation

$$a + ib = e^{i\theta/2}$$

Just set

$$\begin{aligned} q &= (q_0, q_1, q_2, q_3) \\ &= q_0 + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3 \\ &= e^{(\mathbf{I}\cdot\hat{\mathbf{n}}\theta/2)} \end{aligned}$$

with $q_0 = \cos(\theta/2)$ and $\vec{\mathbf{q}} = \hat{\mathbf{n}}\sin(\theta/2)$ and $\mathbf{I} = (\mathbf{i}, \mathbf{j}, \mathbf{k})$, with $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1$, and $\mathbf{i} * \mathbf{j} = \mathbf{k}$ (cyclic),

Cute Quaternion Tricks!

Square Roots are cool..

A quaternion p is the **square root** of a quaternion q if

$$p * p = q\ .$$

A hint: remember that if $c = \cos\theta$, and $\gamma = \cos(\frac{\theta}{2})$, then

$$\gamma = \sqrt{\frac{1+c}{2}} = \frac{1+c}{\sqrt{2(1+c)}}$$

Cute Quaternion Tricks...

Suppose we now look at $1 + q = (1 + q_0, \mathbf{q})$. Then

$$\begin{aligned} (1+q) * (1+q) &= \left((1+q_0)^2 - \mathbf{q}\cdot\mathbf{q},\ 2\mathbf{q}(1+q_0)\right) \\ &= 2(1+q_0)\,q \end{aligned}$$

Dividing through by $2(1 + q_0)$, we find the **square root:**

$$p = \sqrt{q} = \frac{1+q}{\sqrt{2(1+q_0)}}$$

Tricks, contd: Lining up $\hat{a}$ and $\hat{b}$

A common rotation task is to line up two directions, $\hat{a}$ and $\hat{b}$. There is a simple **quaternion form** for this operation. Let

$$\hat{a} \cdot \hat{b} = \cos\theta = c\,, \quad \hat{a} \times \hat{b} = \hat{n}\,\sin\theta$$

where we assume $\sin\theta > 0$. Then we can compute the rotation from $\hat{a}$ to $\hat{b}$ using, again, the half-angle formula:

$$\begin{aligned} R(\hat{a},\hat{b}) &= (\cos(\theta/2),\, \hat{n}\sin(\theta/2)) \\ &= \left(\sqrt{\frac{1+c}{2}},\, \hat{a}\times\hat{b}\sqrt{\frac{1}{2(1+c)}}\right) \end{aligned}$$

where we also used $\sin\theta = 2\cos(\theta/2)\sin(\theta/2)$.

Clifford Algebras

- **All Rotations in any dimension are represented by *two reflections* using Clifford Algebra:**
  $\mathbf{A}$ and $\mathbf{B}$ define the perpendicular directions to two reflection planes, $\mathbf{A}\cdot\mathbf{A} = \mathbf{B}\cdot\mathbf{B} = 1$.
- **Create Rotation Matrix R and solve for the Quaternion, and you amazingly get THIS:**

$$q(\mathbf{A},\mathbf{B}) = (\mathbf{A}\cdot\mathbf{B},\, \mathbf{A}\times\mathbf{B})$$

Clifford Algebra Quaternion Form . . .

Why is this a quaternion form?

$$\begin{aligned} q\cdot q &= (\mathbf{A}\cdot\mathbf{B})^2 + (\mathbf{A}\times\mathbf{B})\cdot(\mathbf{A}\times\mathbf{B}) \\ &= (\mathbf{A}\cdot\mathbf{A})\,(\mathbf{B}\cdot\mathbf{B}) \\ &\equiv 1 \end{aligned}$$

If Quaternions are like the Square Roots of Rotations, then Clifford Algebras are like the Square Roots of Quaternions!

Key to Quaternion Intuition

Fundamental Intuition: We know

$$q_0 = \cos(\theta/2),\quad \vec{q} = \hat{n}\sin(\theta/2)$$

We also know that *any coordinate frame* M can be written as $M = R(\theta,\hat{n})$.

Therefore

$\vec{q}$ points exactly along the axis we have to rotate around to go from identity I to M, and the length of $\vec{q}$ tells us how much to rotate.

Summarize Quaternion Properties

- **Unit four-vector.** Take $q = (q_0, q_1, q_2, q_3) = (q_0, \vec{q})$ to obey constraint $q\cdot q = 1$.
- **Multiplication rule.** The quaternion product q and p is
  $q * p = (q_0p_0 - \vec{q}\cdot\vec{p},\, q_0\vec{p} + p_0\vec{q} + \vec{q}\times\vec{p})$,
  or, alternatively,

$$\begin{bmatrix} [q*p]_0 \\ [q*p]_1 \\ [q*p]_2 \\ [q*p]_3 \end{bmatrix} = \begin{bmatrix} q_0p_0 - q_1p_1 - q_2p_2 - q_3p_3 \\ q_0p_1 + q_1p_0 + q_2p_3 - q_3p_2 \\ q_0p_2 + q_2p_0 + q_3p_1 - q_1p_3 \\ q_0p_3 + q_3p_0 + q_1p_2 - q_2p_1 \end{bmatrix}$$

Quaternion Summary . . .

- **Rotation Correspondence.** The unit quaternions q and $-q$ correspond to a single 3D rotation R_3:

$$\begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2q_1q_2-2q_0q_3 & 2q_1q_3+2q_0q_2 \\ 2q_1q_2+2q_0q_3 & q_0^2-q_1^2+q_2^2-q_3^2 & 2q_2q_3-2q_0q_1 \\ 2q_1q_3-2q_0q_2 & 2q_2q_3+2q_0q_1 & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix}$$

If

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

with $\hat{n}$ a unit 3-vector, $\hat{n}\cdot\hat{n} = 1$. Then $R(\theta,\hat{n})$ is usual 3D rotation by θ in the plane $\perp$ to $\hat{n}$.

SUMMARY

- Quaternions represent 3D frames
- Quaternion multiplication represents 3D rotation
- Quaternions are points on a hypersphere
- Quaternions paths can be visualized with 3D display
- Belt Trick, Rolling Ball, and Gimbal Lock can be understood as Quaternion Paths.

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.

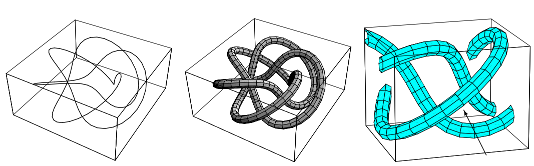
$*$ We **FIX one direction**, generally the **tangent to the curve.**

$*$ 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.

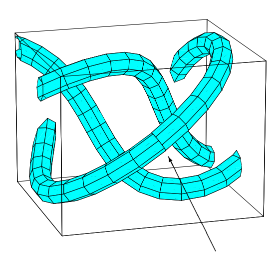
Examine Framing of Curves



The (3,5) torus knot.

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

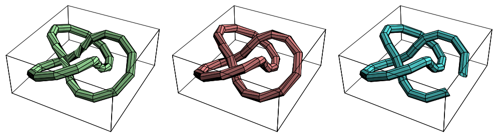
Example of Tubing Problems on Curves



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:

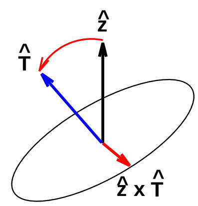
- **Ambiguity of Frame.** We have freedom to choose a "gauge," i.e., any additional rotation around tangent vector, at *any* curve point.
- **Circles in q space.** "Gauge freedom" generates *great circles* in S^3 quaternion space. Need 4π radians to get full quaternion circle.
- **Gauge-invariant swept 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:

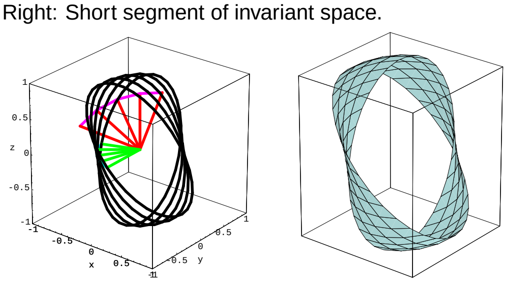
$q(\theta, \hat{\mathbf{T}}) * q(\hat{\mathbf{z}} \cdot \hat{\mathbf{T}}, \hat{\mathbf{z}} \times \hat{\mathbf{T}})$ generates the correct family of quaternion curves:



Invariant Quaternion Frames . . .

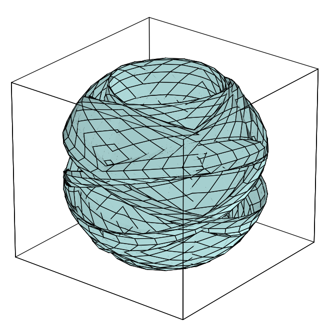
Invariant frame for trefoil knot:

$*$ 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.

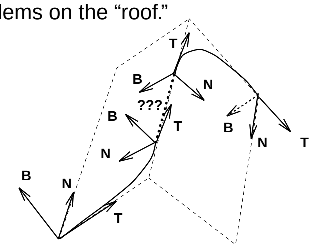
Parallel Transport frame (Bishop): $\sigma = 0$ to get minimal turning. NON-LOCAL = an INTEGRAL.

3D curve frames, contd

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

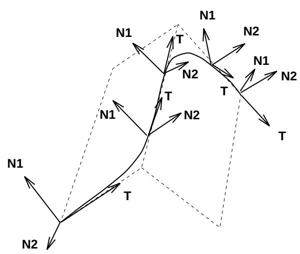
$$\mathbf{B}(t) = \frac{\mathbf{x}'(t) \times \mathbf{x}''(t)}{\|\mathbf{x}'(t) \times \mathbf{x}''(t)\|}$$

but has problems on the "roof."

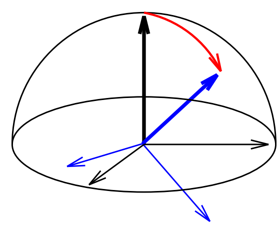


3D curve frames, contd

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



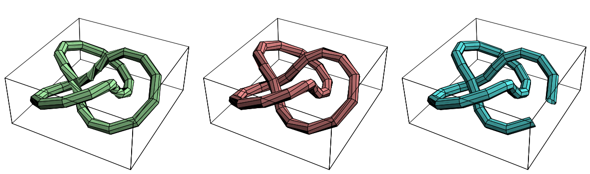
3D curve frames, contd



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

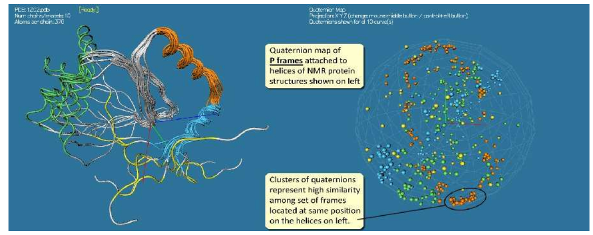
Observations:

- **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.

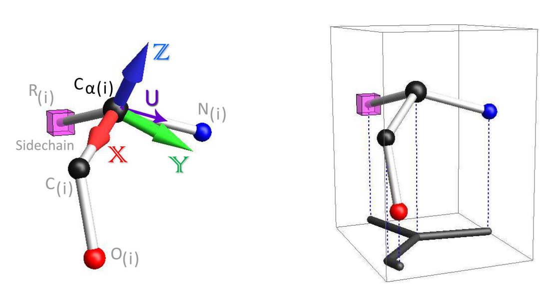
New Progress: Quaternion Frames in Proteomics

- **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

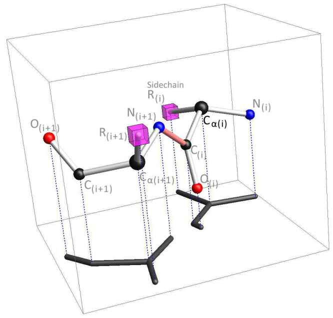
- **Library of 20 amino acids.** Proteins link these together with *peptide bonds*: a C'–OH unit on one end sees an NH_2–C_α on the other side, and joins together as C'–NH–C_α, kicking off a water, H_2O.
- **Pick Three Atoms.** Any three noncollinear atoms are sufficient to define a quaternion frame, 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



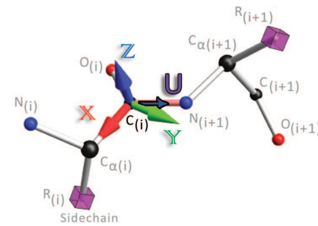
Amino acid geometry showing the computation of our default discrete frame based on the direction from the C_α to the neighboring C and N atoms. The frame vectors $\mathbb{X}$ (red), $\mathbb{Y}$ (green), and $\mathbb{Z}$ (blue) are superimposed on the basic amino acid unit structure.

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:

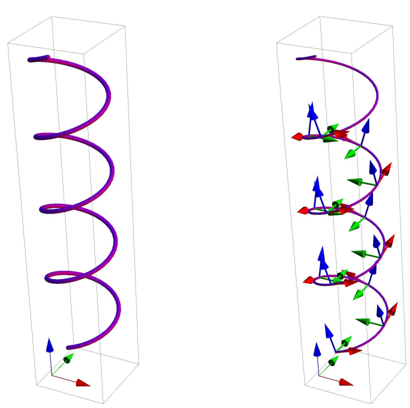
- **Alpha Helix.** One of the most common structures is the Alpha Helix, formed when sequences of residues relax into a low-energy state that coils them into a spiral.
- **Beta Sheet.** Another common structure is essentially sequence of residues related to each other by 180-degree flips, giving the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

Model of an Alpha Helix

(a) A helix defined by the parametric equation

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

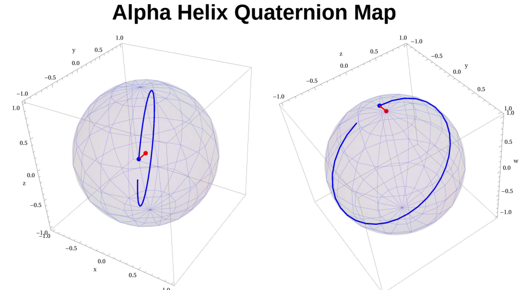
(b) A set of frames on the helical curve defined by the Frenet-Serret equation. Note the relation of the identity frame at bottom left to the first actual helix frame.



(a) (b)

Model of an Alpha Helix

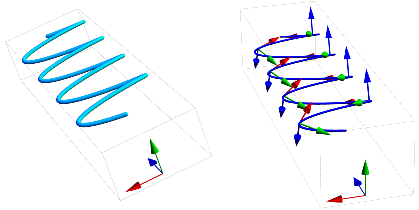
Alpha Helix Quaternion Map



(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



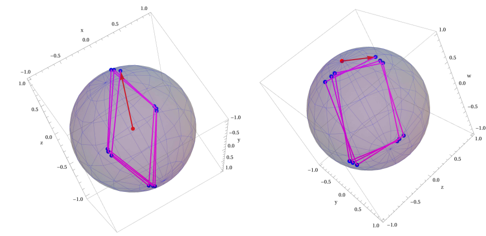
(a) (b)

(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.

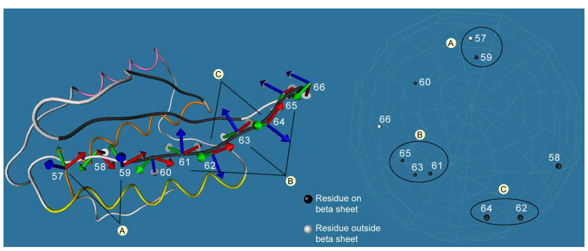
Beta Sheet Quaternion Map



(a) (b)

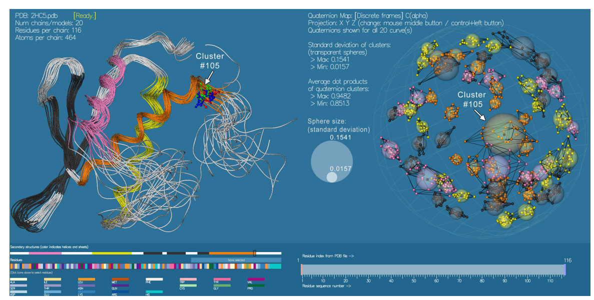
A beta sheet modeled by the parametric equation $(\cos(t),\, 0.1\sin(t),\, 0.5t)$. (a) xyz map. (b) wyz map. Red dot is identity frame.

Example: Beta Sheet Quaternion Map



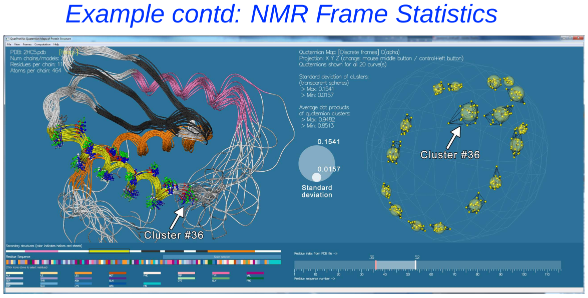
Protein structure of 2HC5 and a quaternion map of its beta sheet structure. Neighboring frames are given matching quaternion signs in this map.

Example: Quaternion NMR Frame Statistics



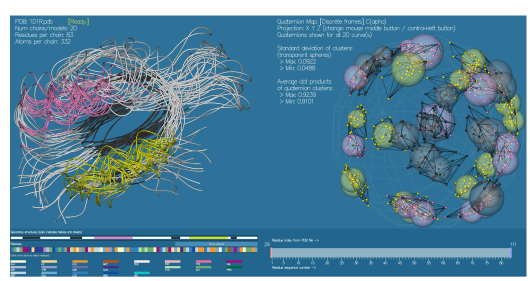
Quaternion maps for NMR data describing 10 deformations of YvyC. (left) Spatial geometries. (right) Quaternion *orientation space* geometry spreads for each amino acid residue.

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.
- **Dual quaternions can do 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 ϵ, where $\epsilon^2 = 0$.

Dual Quaternions...

- **Long 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. Kavan et al. (TOG, 2008) have spurred their use in graphics for skinning problems, etc.

Dual Quaternions...

- **Closely related to Clifford Algebra (Geometric Algebra).** See Dorst et al., *Geometric Algebra for Computer Science* for the connection between dual quaternions and Clifford algebras. Note that quaternion rotation properties can be generalized to N-dimensions using Clifford Algebras. The same is true for the properties of dual quaternions. We will not pursue the Clifford Algebra connection here, since we have time only for a brief introduction. We will use the more straightforward dual-number approach favored in the kinematics literature, but the reader should be aware of the Clifford Algebra association.

Approach to Adding in Translations

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

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

So that's basically all there is to it...

toward Dual Quaternions...

Well, almost all there is to it...

- Try 2D rotation with complex numbers:
  $e^{i\theta}(x+iy) = (x\cos\theta - y\sin\theta) + i(x\sin\theta + y\cos\theta)$.
  Seems ok.
- Translation needs *two* pieces, non-dual and dual:

$$
\begin{aligned}
e^{\epsilon(x+iy)}e^{\epsilon(a+ib)} &= (1+\epsilon(x+iy))(1+\epsilon(a+ib)) \\
&= 1 + \epsilon(x+a+i(y+b))
\end{aligned}
$$

- BUT then if you try the $e^{i\theta}$ rotation trick, you get into trouble, e.g., $1 \to e^{i\theta}$.

Introduce Dual Quaternions

So we fix this by introducing a **dual version of the sandwich operation,** and dualizing the **vector** as well:

$$\text{Usual quat rot:} \quad R \cdot \mathbf{x} \;=\; q * (0, \mathbf{x}) * q^*$$
$$\text{Add trans:} \quad R \cdot \mathbf{x} + \mathbf{t} \;=\; \hat{q} * (1 + \epsilon(0), 0 + \epsilon\,\mathbf{x}) * \bar{\hat{q}}^* \,,$$

Here * is quaternion conjugation and $^-$ is dual conjugation,

$$\hat{q} = \left(1 + \frac{1}{2}\epsilon(0, \mathbf{t})\right) * (\text{Usual } \; q)$$

NOTE: when *sandwiched*, this **one-half translation** produces precisely $R \cdot \mathbf{x} + \mathbf{t}$. So this is the key to putting translation into the usual quaternion. **IDEA:** $\epsilon^2 = 0$ **truncates the** cos, sin **series that would produce a** *rotation*, **and leaves only a single term, the** *translation*.

Basics of Dual Quaternion Operations

- **Basic 3-vector** in dual quaternion notation is
$$\mathbf{x} = (1, 0, 0, 0) + \epsilon(0, x, y, z) = 1 + \epsilon\,\mathbf{x}$$
- **Basic dual quaternion** becomes
$$\hat{q} = \left((1, 0, 0, 0) + \tfrac{1}{2}\epsilon\,(0,\, t_x, t_y, t_z)\right) * q,$$
where (t_x, t_y, t_z) are the translation parameters and q, with $q \cdot q = 1$, is a standard rotational unit quaternion. Note the $\frac{1}{2}t$ terms, similar to $\theta \rightarrow \frac{\theta}{2}$ in q.
- **The full space-motion transformation formula** is then
$$\mathbf{x}' \;=\; \hat{q} * \mathbf{x} * \bar{\hat{q}}^* \,.$$

Useful Properties of Dual Quaternions

- **Unit Dual Quaternions:** The general form of $\hat{q}$ is assumed to be unit length as usual. This is tricky because the **norm** of $\hat{q} = q + \epsilon\, e$ is $\quad \|\hat{q}\| = \|q\| + \epsilon \frac{q \cdot e}{\|q\|}$, which means that, since $\|q\| = 1$, we must enforce $q \cdot e = 0$. It turns out that this can be done compatibly with a general spatial motion.
- **Trigonometric/Exponential Form:** In general, any unit dual quaternion can be written as
$$\begin{aligned} \hat{q} \;&=\; \cos\frac{\hat{\theta}}{2} + \hat{s}\sin\frac{\hat{\theta}}{2} \\ &=\; \cos\frac{\theta + \epsilon\tau}{2} + (s + \epsilon t)\sin\frac{\theta + \epsilon\tau}{2} \end{aligned}$$
To compute this, we need formulas like $\cos(a + \epsilon\, b) \;=\; \cos a - \epsilon\, b \sin a$ and $\sin(a + \epsilon\, b) = \sin a + \epsilon\, b \cos a$, which follow from series expansion.

Useful Properties of Dual Quaternions. . .

- **Exponential and Log:** Using power series, one can extend the usual quaternion exponential and log formulas:
$\exp(\hat{s}\hat{\theta}) = \cos\frac{\hat{\theta}}{2} + \hat{s}\sin\frac{\hat{\theta}}{2}$
and so obviously $\log \hat{q} = \hat{s}\,\hat{\theta}$.
- **Inverse:** The inverse of a dual object is:
$(a + \epsilon\, b)^{-1} = \frac{1}{a + \epsilon\, b} = \frac{1}{a} - \epsilon\,\frac{b}{a^2}$
as can be easily confirmed from $(a + \epsilon\, b)(c + \epsilon\, d) = ac + \epsilon\,(ad + bc)$
- **Inverse:** A trick similar to the one we saw for quaternions works for the square root of a dual quaternion:
$\sqrt{a + \epsilon\, b} = \sqrt{a} + \epsilon\,\frac{b}{2\sqrt{a}}$.

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.**