Visualizing Quaternions

Course Notes for SIGGRAPH 2007

Andrew J. Hanson
Computer Science Department
Indiana University
Bloomington, IN 47405 USA
Email: hansona@indiana.edu

Abstract

This intermediate-level tutorial provides a comprehensive approach to the visualization of quaternions and their relationships to computer graphics and scientific visualization. The introduction focuses on a selection of everyday phenomena involving rotating objects whose explanation is essentially impossible without a quaternion visualization. The presentation will then pursue selected examples of quaternion-based visualization methods to help explain the behavior of *quaternion manifolds*: quaternion representations of orientation frames attached to curves, surfaces, and volumes.

Presenter's Biography

Andrew J. Hanson is a professor of computer science at Indiana University, and has regularly taught courses in computer graphics, computer vision, and scientific visualization. He received a BA in chemistry and physics from Harvard College in 1966 and a PhD in theoretical physics from MIT in 1971. Before coming to Indiana University, he did research in theoretical physics at the Institute for Advanced Study, Stanford, and Berkeley, and then in computer vision at the SRI Artificial Intelligence Center in Menlo Park, CA. He has published in IEEE Computer, CG&A, TVCG, ACM Computing Surveys, and has over a dozen papers in the IEEE Visualization Proceedings. He has also contributed three articles to the Graphics Gems series dealing with user interfaces for rotations and with techniques of N-dimensional geometry. Previous experience with conference tutorials includes a Siggraph '98 tutorial on N-dimensional graphics, a Visualization '98 course on Clifford Algebras and Quaternions, and tutorials on Visualizing Quaternions presented at Siggraph 1999, Siggraph 2000, at Siggraph 2001 in tandem with a course on Visualizing Relativity for a graphics audience, and again at Siggraph 2005. Major research interests include scientific visualization, machine vision, computer graphics, perception, and the design of interactive user interfaces for virtual reality and visualization applications. Particular visualization applications currently being studied include astrophysical visualizations, interactive interfaces for very large scales such as astrophysics and cosmology, and the exploitation of constrained navigation for visualization environments. Mathematical visualization interests include the development of interactive high-dimensional geometry visualization, multimedia haptic interfaces for exploration and manipulation of mathematical objects in dimensions three and four, the depiction of Calabi-Yau spaces, and the general problems of graphics and visualization in dimensions greater than three and their applications to mathematics and theoretical physics.

Contents

General Information on the Tutorial

Course Syllabus

Summary: This tutorial will deal with visualizable representations of quaternions, their features, technology, folklore, and applications. The introduction will focus on visually understanding quaternions themselves by exploiting parallels to complex variables and 2D rotations. Starting from this basis, the tutorial will proceed to give visualizations of advanced quaternion applications.

Prerequisites: Participants should be comfortable with and have an appreciation for conventional mathematical methods of 3D computer graphics and geometry used in geometric transformations and polygon rendering. The material will be of most interest to those wishing to deepen their intuitive understanding of moving coordinate frames and quaternion-based animation techniques.

Objectives: Participants will learn the basic facts relating quaternions to ordinary 3D rotations, as well as methods for examining the properties of quaternion constructions using interactive visualization methods. A variety of applications, including the use of quaternions to samples coordinate frames for curves, surfaces, and volumes, will be explored.

Outline: This tutorial will last approximately two hours including a break and time for questions and discussion. The material will be arranged as follows:

I. *(50 min + questions)* **Twisting Belts, Rolling Balls, and Locking Gimbals:** *Explaining Rotation Sequences with Quaternions*

Sequences of orientations are manifestly evident in our everyday lives. While we can immediately observe strange things that happen when we twist a leather belt, roll a baseball, or push on a gyroscope, if you ask "why" and expect a real explanation, most of us hit a dead end. Quaternion visualization provides satisfying answers to such questions. Interactive demonstrations are provided.

II. *(50 min + questions)* **Quaternion Fields:** *Curves, Surfaces, and Volumes*

Once we have mastered the visualization of quaternion paths, we have the tools to take a fresh look at many problems in graphics and visualization. The quaternion *field* is a continuous map from a set of orientation frames such as framed curves, surfaces, and volumes into the corresponding quaternions. We examine a family of examples showing how quaternion curves, surfaces, and volumes can solve old problems and reveal new properties. Examples include general approaches to textured tubings.

1 Overview

Practitioners of computer graphics and animation frequently represent 3D rotations using the quaternion formalism, a mathematical tool that originated with William Rowan Hamilton in the 19th century, and is now an essential part of modern analysis, group theory, differential geometry, and even quantum physics. Quaternions are in many ways very simple, and yet there are enormous subtleties to address in the process of fully understanding and exploiting their properties. The purpose of this Tutorial is to construct an intuitive bridge between our intuitions about 2D and 3D rotations and the quaternion representation.

The Tutorial will begin with an introduction to various natural phenomena that can be understood using quaternions. Rotations in 2D, which will be found to have surprising richness, will lead the way to the construction of the relation between 3D rotations and quaternions. Quaternion visualization methods of various sorts will be introduced, followed by applications of the quaternion frame representation to problems of interest by graphicists and visualization scientists. An extensive bibliography of related literature is included, as well as several relevant reprints and technical reports, a Mathematica implementation of the Quaternion Frenet Equations, and a basic GLUT quaternion visualization application.

2 Twisting Belts, Rolling Balls, and Locking Gimbals

We will begin with a basic introduction to the ways in which sequences of rotations enter our lives in surprising ways. We will then proceed to look at a variety of methods for understanding quaternions and making meaningful pictures of constructs involving them. These methods will range from some of the concepts pointed out by Hart, Francis, and Kauffman [54] to theoretical methods given in [47, 48, 40, 51].

Traditional treatments of quaternions range from the original works of Hamilton and Tait [35, 85] to a variety of recent studies such as those of Altmann, Pletincks, Juttler, and Kuipers [2, 73, 63, 67].

In our pedagogical treatment, we will focus on the use of 2D rotations as a rich but algebraically simple proving ground in which we can see many of the key features of quaternion geometry in a very manageable context. The relationship between 3D rotations and quaternions is then introduced as a natural extension of the 2D systems. Quaternion visualization itself utilizes a basic trick: since a four-vector quaternion $q = (q_0, \mathbf{q})$ obeying $q \cdot q = 1$, then the four-vector lies on the three-sphere S^3 and has only three independent components: if we display just $\mathbf{q}$, we can in principle *infer* the value of $q_0 = \sqrt{1 - \mathbf{q} \cdot \mathbf{q}}$.

3 Quaternion Fields

After the conceptual introduction, we proceed to study the nature of quaternions as representations of frames in 3D. The now-traditional application of quaternion animation splines was introduced to the graphics community originally by Shoemake [77]. Our visualizations of these and other applications exploit the fact that quaternions are points on the three-sphere embedded in 4D; the three-sphere (S^3) is analogous to an ordinary ball or two-sphere (S^2) embedded in 3D, except that the three-sphere is a solid object instead of a surface. To manipulate, display, and visualize rotations in 3D, we may convert 3D rotations to 4D quaternion points and treat the entire problem in the framework of 4D geometry.

We pursue three main applications, which involve the identification of quaternion frames with sampled curves, surfaces, and volumes. The curve methods follow closely techniques introduced in Hanson and Ma [47, 48] for representing families of coordinate frames on curves in 3D as curves in the 4D quaternion space. Interesting insights result from studying the problem of applying a texture to a tube surrounding an arbitrary open or closed curve. The extension to surfaces and the corresponding induced surfaces in quaternion space follow the treatment by Hanson [40, 51], and volumetric quaternions are studied using the methods of Herda, et al. [56, 57, 55].

4 Demonstration Software

We provide an elementary OpenGL-based interactive quaternion visualization application, *QuatRot*, that should be essentially system-independent and run on any platform. In addition, we supply our own version, `quatutils.nb`, of some basic Mathematica routines for quaternions (which serve as the basis for a number of the illustrations in the notes), as well as a Mathematica notebook `qfrmint.nb` that explicitly implements a numerical integration of the Frenet frame equations in quaternion form, vastly improving the exactly equivalent calculation for the standard Frenet equations implemented by Gray [33].

Acknowledgments

Portions of the course notes are adapted from the book, *Visualizing Quaternions*, by Andrew J. Hanson, published by Morgan Kaufmann Publishers, Copyright 2006 by Elsevier Inc. We thank the publisher for permission to use this material; all rights not explicitly assigned to Siggraph for the purpose of providing these course notes are reserved.

Republished in the Course Notes with permission are two key papers from IEEE Transactions on Visualization and Computer Graphics [48], and from the Proceedings of IEEE Visualization [40]; we thank the IEEE Computer Society Press for permitting us to include this material.

Finally we would like to thank the National Science Foundation for their support: the research incorporated in portions of this work was supported in part by NSF grant CCR-0204112.

References

[1] B. Alpern, L. Carter, M. Grayson, and C. Pelkie. Orientation maps: Techniques for visualizing rotations (a consumer's guide). In *Proceedings of Visualization '93*, pages 183–188. IEEE Computer Society Press, 1993.

[2] S. L. Altmann. *Rotations, Quaternions, and Double Groups*. Oxford University Press, 1986.

[3] M.F. Atiyah, R. Bott, and A. Shapiro. Clifford modules. *Topology*, 3, Suppl. 1:3–38, 1986.

[4] T. Banchoff and J. Werner. *Linear Algebra through Geometry*. Springer-Verlag, 1983.

[5] T. F. Banchoff. Visualizing two-dimensional phenomena in four-dimensional space: A computer graphics approach. In E. Wegman and D. Priest, editors, *Statistical Image Processing and Computer Graphics*, pages 187–202. Marcel Dekker, Inc., New York, 1986.

[6] Thomas F. Banchoff. *Beyond the Third Dimension: Geometry, Computer Graphics, and Higher Dimensions*. Scientific American Library, New York, NY, 1990.

[7] David Banks. Interactive display and manipulation of two-dimensional surfaces in four dimensional space. In *Symposium on Interactive 3D Graphics*, pages 197–207, New York, 1992. ACM.

[8] David Banks. Interactive manipulation and display of two-dimensional surfaces in four-dimensional space. In David Zeltzer, editor, *Computer Graphics (1992 Symposium on Interactive 3D Graphics)*, volume 25, pages 197–207, March 1992.

[9] David C. Banks. Illumination in diverse codimensions. In *Computer Graphics*, pages 327–334, New York, 1994. ACM. Proceedings of SIGGRAPH 1994; Annual Conference Series 1994.

[10] A. Barr, B. Currin, S. Gabriel, and J. Hughes. Smooth interpolation of orientations with angular velocity constraints using quaternions. In *Computer Graphics Proceedings, Annual Conference Series*, pages 313–320, 1992. Proceedings of SIGGRAPH '92.

[11] Richard L. Bishop. There is more than one way to frame a curve. *Amer. Math. Monthly*, 82(3):246–251, March 1975.

[12] Wilhelm Blaschke. *Kinematik und Quaternionen*. VEB Deutscher Verlag der Wissenschaften, Berlin, 1960.

[13] Jules Bloomenthal. Calculation of reference frames along a space curve. In Andrew Glassner, editor, *Graphics Gems*, pages 567–571. Academic Press, Cambridge, MA, 1990.

[14] Kenneth A. Brakke. The surface evolver. *Experimental Mathematics*, 1(2):141–165, 1992. The "Evolver" system, manual, and sample data files are available by anonymous ftp from geom.umn.edu, The Geometry Center, Minneapolis MN.

[15] D. W. Brisson, editor. *Hypergraphics: Visualizing Complex Relationships in Art, Science and Technology*, volume 24. Westview Press, 1978.

[16] S. A. Carey, R. P. Burton, and D. M. Campbell. Shades of a higher dimension. *Computer Graphics World*, pages 93–94, October 1987.

[17] Michael Chen, S. Joy Mountford, and Abigail Sellen. A study in interactive 3-d rotation using 2-d control devices. In *Proceedings of Siggraph 88*, volume 22, pages 121–130, 1988.

[18] H.S.M. Coxeter. *Regular Complex Polytopes*. Cambridge University Press, second edition, 1991.

[19] R. A. Cross and A. J. Hanson. Virtual reality performance for virtual geometry. In *Proceedings of Visualization '94*, pages 156–163. IEEE Computer Society Press, 1994.

[20] A.R. Edmonds. *Angular Momentum in Quantum Mechanics*. Princeton University Press, Princeton, New Jersey, 1957.

[21] N.V. Efimov and E.R. Rozendorn. *Linear Algebra and Multi-Dimensional Geometry*. Mir Publishers, Moscow, 1975.

[22] T. Eguchi, P. B. Gilkey, and A. J. Hanson. Gravitation, gauge theories and differential geometry. *Physics Reports*, 66(6):213–393, December 1980.

[23] L. P. Eisenhart. *A Treatise on the Differential Geometry of Curves and Surfaces*. Dover, New York, 1909 (1960).

[24] S. Feiner and C. Beshers. Visualizing n-dimensional virtual worlds with n-vision. *Computer Graphics*, 24(2):37–38, March 1990.

[25] S. Feiner and C. Beshers. Worlds within worlds: Metaphors for exploring n-dimensional virtual worlds. In *Proceedings of UIST '90, Snowbird, Utah*, pages 76–83, October 1990.

[26] Gerd Fischer. *Mathematische Modelle*, volume I and II. Friedr. Vieweg & Sohn, Braunschweig/Wiesbaden, 1986.

[27] H. Flanders. *Differential Forms*. Academic Press, New York, 1963.

[28] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics, Principles and Practice*. Addison-Wesley, second edition, 1990. page 227.

[29] A. R. Forsyth. *Geometry of Four Dimensions*. Cambridge University Press, 1930.

[30] George K. Francis. *A Topological Picturebook*. Springer Verlag, 1987.

[31] Herbert Goldstein. *Classical Mechanics*. Addison-Wesley, 1950.

[32] F.S. Grassia. Practical parameterization of rotations using the exponential map. *Journal of Graphics Tools*, 3(3):29–48, 1998.

[33] Alfred Gray. *Modern Differential Geometry of Curves and Surfaces*. CRC Press, Inc., Boca Raton, FL, second edition, 1998.

[34] Cindy M. Grimm and John F. Hughes. Modeling surfaces with arbitrary topology using manifolds. In *Computer Graphics Proceedings, Annual Conference Series*, pages 359–368, 1995. Proceedings of SIGGRAPH '95.

[35] W.R. Hamilton. *Lectures on Quaternions*. Cambridge University Press, 1853.

[36] A. J. Hanson. The rolling ball. In David Kirk, editor, *Graphics Gems III*, pages 51–60. Academic Press, Cambridge, MA, 1992.

[37] A. J. Hanson. A construction for computer visualization of certain complex curves. *Notices of the Amer.Math.Soc.*, 41(9):1156–1163, November/December 1994.

[38] A. J. Hanson. Geometry for n-dimensional graphics. In Paul Heckbert, editor, *Graphics Gems IV*, pages 149–170. Academic Press, Cambridge, MA, 1994.

[39] A. J. Hanson. Rotations for n-dimensional graphics. In Alan Paeth, editor, *Graphics Gems V*, pages 55–64. Academic Press, Cambridge, MA, 1995.

[40] A. J. Hanson. Constrained optimal framings of curves and surfaces using quaternion gauss maps. In *Proceedings of Visualization '98*, pages 375–382. IEEE Computer Society Press, 1998.

[41] A. J. Hanson. *Visualizing Quaternions*. Morgan Kaufmann, San Francisco, CA, 2006.

[42] A. J. Hanson and R. A. Cross. Interactive visualization methods for four dimensions. In *Proceedings of Visualization '93*, pages 196–203. IEEE Computer Society Press, 1993.

[43] A. J. Hanson and P. A. Heng. Visualizing the fourth dimension using geometry and light. In *Proceedings of Visualization '91*, pages 321–328. IEEE Computer Society Press, 1991.

[44] A. J. Hanson and P. A. Heng. Four-dimensional views of 3d scalar fields. In *Proceedings of Visualization '92*, pages 84–91. IEEE Computer Society Press, 1992.

[45] A. J. Hanson and P. A. Heng. Foursight. In *Siggraph Video Review*, volume 85. ACM Siggraph, 1992. Scene 11, Presented in the Animation Screening Room at SIGGRAPH '92, Chicago, Illinois, July 28–31, 1992.

[46] A. J. Hanson and P. A. Heng. Illuminating the fourth dimension. *Computer Graphics and Applications*, 12(4):54–62, July 1992.

[47] A. J. Hanson and H. Ma. Visualizing flow with quaternion frames. In *Proceedings of Visualization '94*, pages 108–115. IEEE Computer Society Press, 1994.

[48] A. J. Hanson and H. Ma. Quaternion frame approach to streamline visualization. *IEEE Trans. on Visualiz. and Comp. Graphics*, 1(2):164–174, June 1995.

[49] A. J. Hanson and H. Ma. Space walking. In *Proceedings of Visualization '95*, pages 126–133. IEEE Computer Society Press, 1995.

[50] A. J. Hanson, T. Munzner, and G. K. Francis. Interactive methods for visualizable geometry. *IEEE Computer*, 27(7):73–83, July 1994.

[51] A.J. Hanson. Quaternion gauss maps and optimal framings of curves and surfaces. Indiana University Computer Science Department Technical Report 518 (October, 1998).

[52] A.J. Hanson, K. Ishkov, and J. Ma. Meshview. A portable 4D geometry viewer written in OpenGL/Motif, available by anonymous ftp from ftp.cs.indiana.edu:pub/hanson.

[53] A.J. Hanson, K. Ishkov, and J. Ma. Meshview: Visualizing the fourth dimension. Overview of the MeshView 4D geometry viewer.

[54] John C. Hart, George K. Francis, and Louis H. Kauffman. Visualizing quaternion rotation. *ACM Trans. on Graphics*, 13(3):256–276, 1994.

[55] L. Herda, R. Urtasun, and P. Fua. Hierarchical Implicit Surface Joint Limits to Constrain Video-Based Motion Capture. In *ECCV*, Prague, Czech Republic, May 2004.

[56] L. Herda, R. Urtasun, A. Hanson, and P. Fua. An Automatic Method For Determining Quaternion Field Boundaries for Ball-and-Socket Joint Limits. In *Proceedings of the 5th International Conference on Automated Face and Gesture Recognition (FGR)*, pages 95–100, Washington, DC, May 2002. IEEE Computer Society.

[57] L. Herda, R. Urtasun, A. Hanson, and P. Fua. Automatic Determination of Shoulder Joint Limits using Experimentally Determined Quaternion Field Boundaries. *International Journal of Robotics Research*, 22(6):419–434, June 2003.

[58] D. Hilbert and S. Cohn-Vossen. *Geometry and the Imagination*. Chelsea, New York, 1952.

[59] John G. Hocking and Gail S. Young. *Topology*. Addison-Wesley, 1961.

[60] C. Hoffmann and J. Zhou. Some techniques for visualizing surfaces in four-dimensional space. *Computer-Aided Design*, 23:83–91, 1991.

[61] S. Hollasch. Four-space visualization of 4D objects. Master's thesis, Arizona State University, August 1991.

[62] H.B. Lawson Jr. and M.L. Michelsohn. *Spin Geometry*. Princeton University Press, 1989.

[63] B. Jüttler. Visualization of moving objects using dual quaternion curves. *Computers and Graphics*, 18(3):315–326, 1994.

[64] B. Jüttler and M.G. Wagner. Computer-aided design with saptial rational B-spline motions. *Journal of Mechanical Design*, 118:193–201, June 1996.

[65] Myoung-Jun Kim, Myung-Soo Kim, and Sung Yong Shin. A general construction scheme for unit quaternion curves with simple high order derivatives. In *Computer Graphics Proceedings, Annual Conference Series*, pages 369–376, 1995. Proceedings of SIGGRAPH '95.

[66] F.. Klock. Two moving coordinate frames for sweeping along a 3d trajectory. *Computer Aided Geometric Design*, 3, 1986.

[67] J.B. Kuipers. *Quaternions and Rotation Sequences*. Princeton University Press, 1999.

[68] J. Milnor. *Topology from the Differentiable Viewpoint*. The University Press of Virginia, Charlottesville, 1965.

[69] Hans Robert Müller. *Sphärische Kinematik*. VEB Deutscher Verlag der Wissenschaften, Berlin, 1962.

[70] G. M. Nielson. Smooth interpolation of orientations. In N.M. Thalman and D. Thalman, editors, *Computer Animation '93*, pages 75–93, Tokyo, June 1993. Springer-Verlag.

[71] Michael A. Noll. A computer technique for displaying n-dimensional hyperobjects. *Communications of the ACM*, 10(8):469–473, August 1967.

[72] Mark Phillips, Silvio Levy, and Tamara Munzner. Geomview: An interactive geometry viewer. *Notices of the Amer. Math. Society*, 40(8):985–988, October 1993. Available by anonymous ftp from geom.umn.edu, The Geometry Center, Minneapolis MN.

[73] D. Pletincks. Quaternion calculus as a basic tool in computer graphics. *The Visual Computer*, 5(1):2–13, 1989.

[74] Ravi Ramamoorthi and Alan H. Barr. Fast construction of accurate quaternion splines. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 287–292. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.

[75] John Schlag. Using geometric constructions to interpolate orientation with quaternions. In James Arvo, editor, *Graphics Gems II*, pages 377–380. Academic Press, 1991.

[76] Uri Shani and Dana H. Ballard. Splines as embeddings for generalized cylinders. *Computer Vision, Graphics, and Image Processing*, 27:129–156, 1984.

[77] K. Shoemake. Animating rotation with quaternion curves. In *Computer Graphics*, volume 19, pages 245–254, 1985. Proceedings of SIGGRAPH 1985.

[78] K. Shoemake. Animation with quaternions. Siggraph Course Lecture Notes, 1987.

[79] Ken Shoemake. Arcball rotation control. In Paul Heckbert, editor, *Graphics Gems IV*, pages 175–192. Academic Press, 1994.

[80] Ken Shoemake. Fiber bundle twist reduction. In Paul Heckbert, editor, *Graphics Gems IV*, pages 230–236. Academic Press, 1994.

[81] D.M.Y. Sommerville. *An Introduction to the Geometry of N Dimensions*. Reprinted by Dover Press, 1958.

[82] N. Steenrod. *The Topology of Fibre Bundles*. Princeton University Press, 1951. Princeton Mathematical Series 14.

[83] K. V. Steiner and R. P. Burton. Hidden volumes: The 4th dimension. *Computer Graphics World*, pages 71–74, February 1987.

[84] D. J. Struik. *Lectures on Classical Differential Geometry*. Addison-Wesley, 1961.

[85] P.G. Tait. *An Elementary Treatise on Quaternions*. Cambridge University Press, 1890.

[86] J. R. Weeks. *The Shape of Space*. Marcel Dekker, New York, 1985.

[87] S. Weinberg. *Gravitation and Cosmology: Principles and Applications of General Relativity*. John Wiley and Sons, 1972.

[88] E.T. Whittaker. *A Treatise on the Analytical Dynamics of Particles and Rigid Bodies*. Dover, New York, New York, 1944.

Visualizing Quaternions

Andrew J. Hanson

Computer Science Department

Indiana University

Siggraph 2007 Tutorial

OUTLINE

I: (50 min) **Twisting Belts, Rolling Balls, and Locking Gimbals:**

Explaining Rotation Sequences with Quaternions

II: (50 min) **Quaternion Fields:**

Curves, Surfaces, and Volumes

Part I

Twisting Belts, Rolling Balls, and Locking Gimbals

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

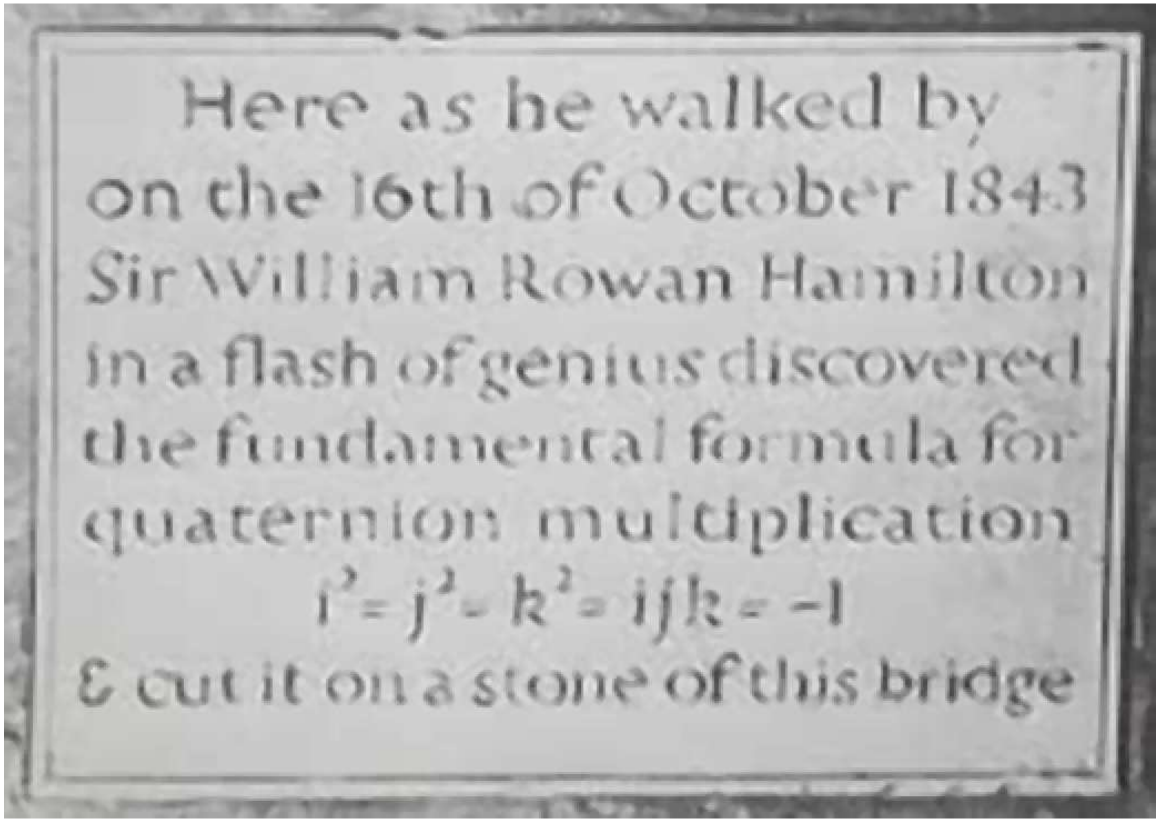
Hamilton's epiphany: 16 October 1843

"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



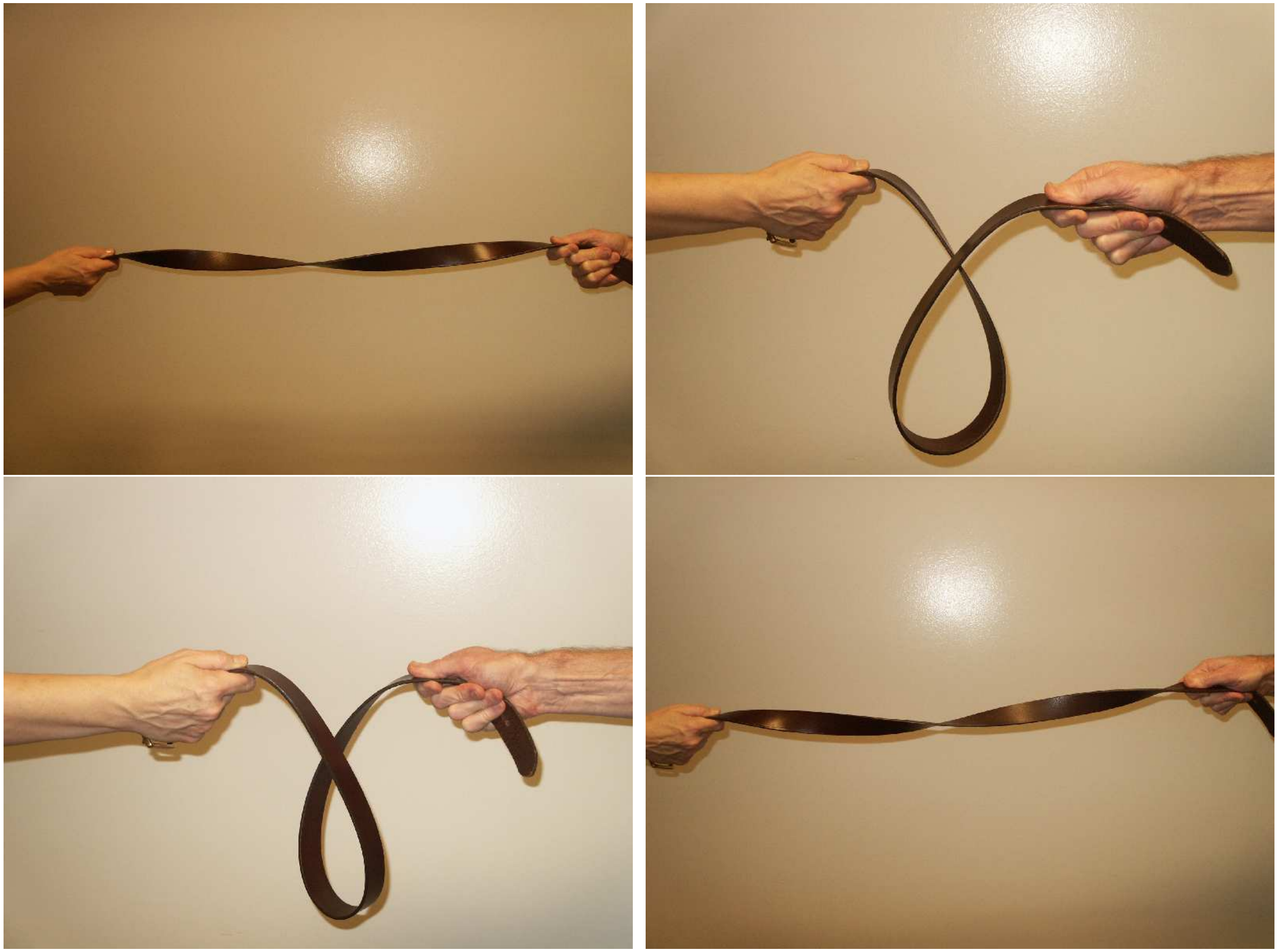
The plaque on Broome Bridge in Dublin, Ireland, commemorating the legendary location where Hamilton conceived of the idea of quaternions. (Hamilton apparently misspelled it as "Brougham Bridge" in his letter.)

The Belt Trick

Quaternion Geometry in our daily lives

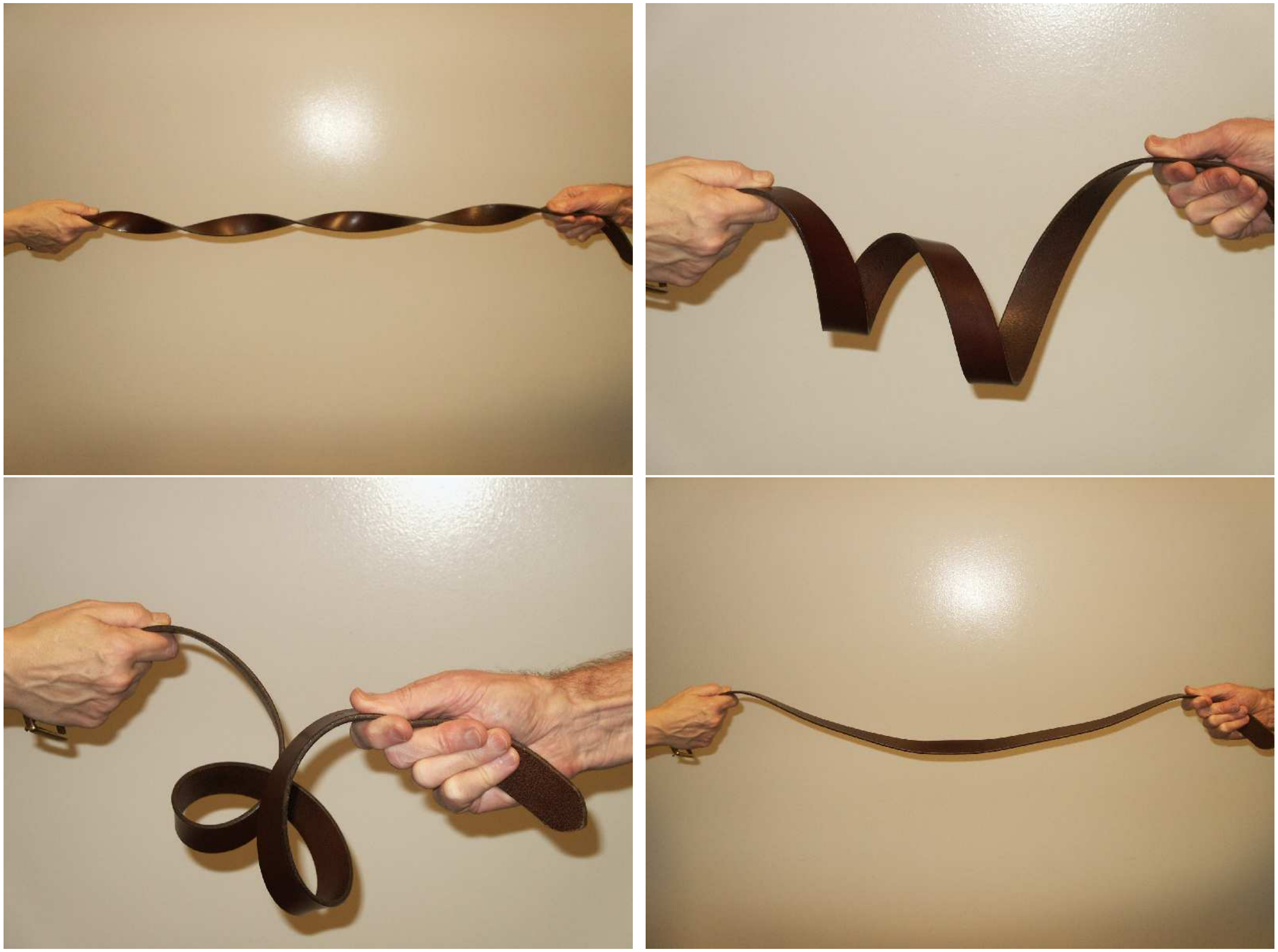
- Two people hold ends of a belt.
- Twist the belt either 360 degrees or 720 degrees.
- **Rule:** *Move belt ends any way you like but do not change orientation of either end.*
- Try to straighten out the belt.

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!*

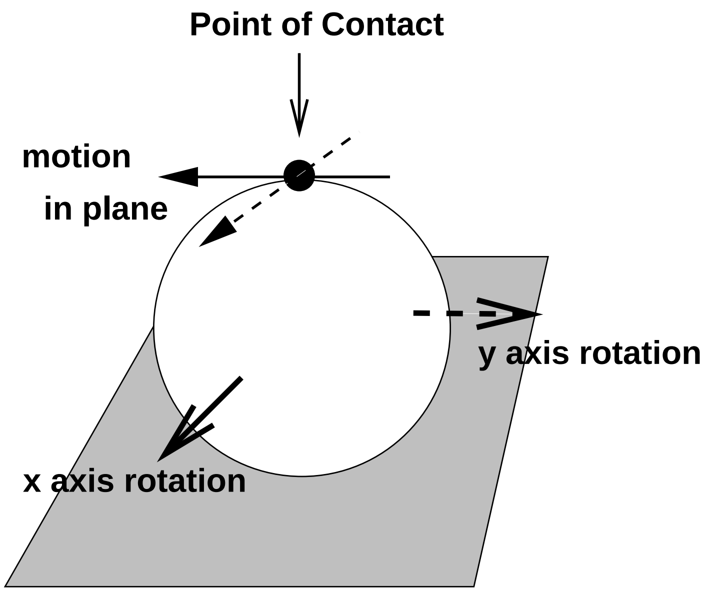
Rolling Ball Puzzle

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* $\longrightarrow$

   equator goes counterclockwise
6. *small counterclockwise circles* $\longrightarrow$

   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.**

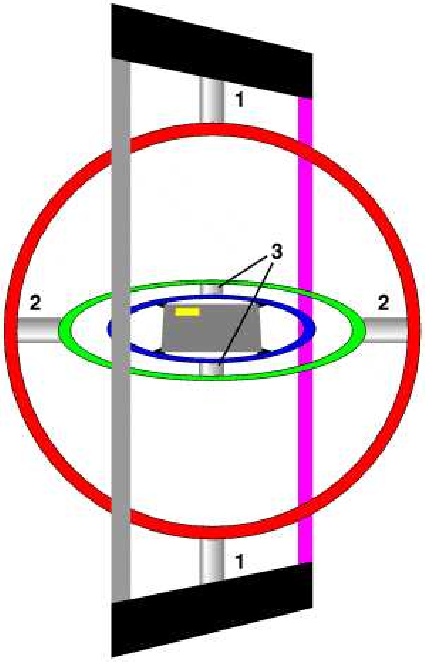


FIGURE 2

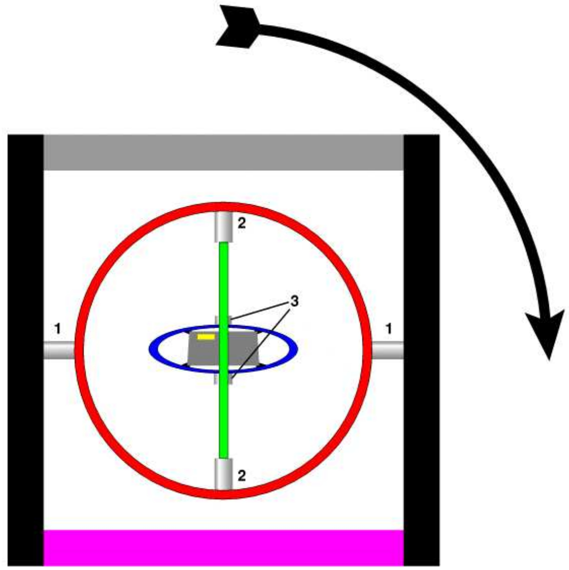


FIGURE 3

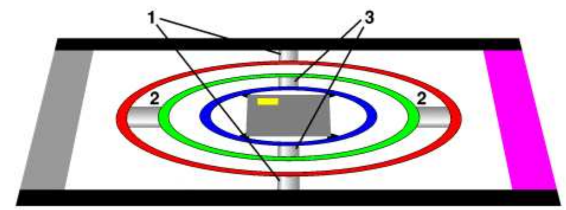


FIGURE 4

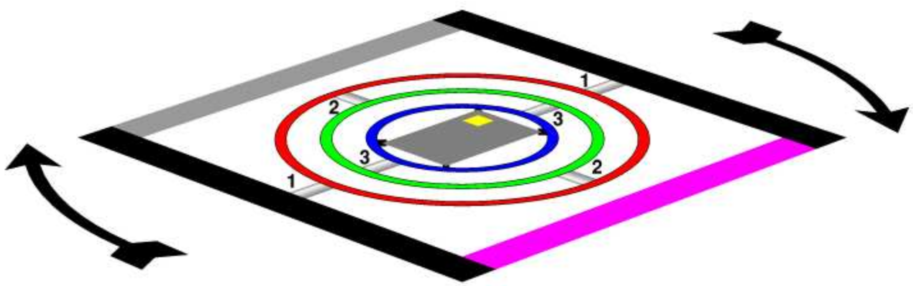
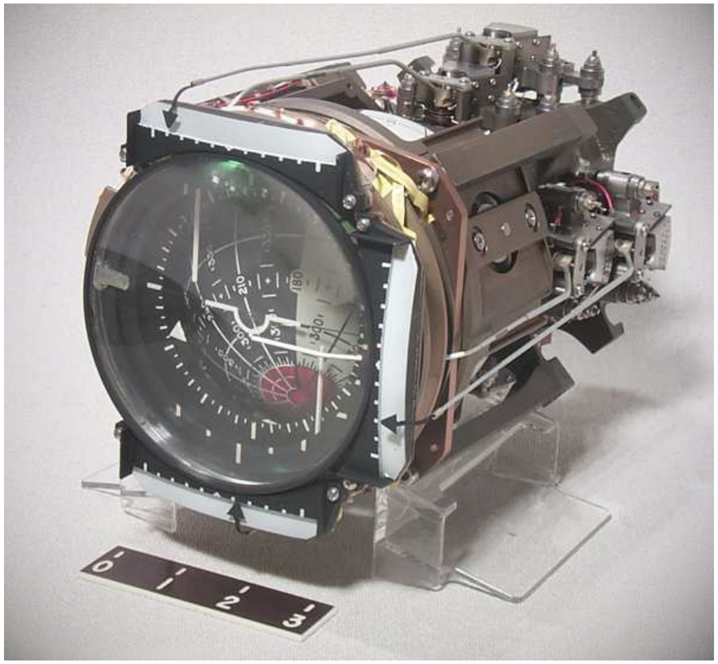


FIGURE 5

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

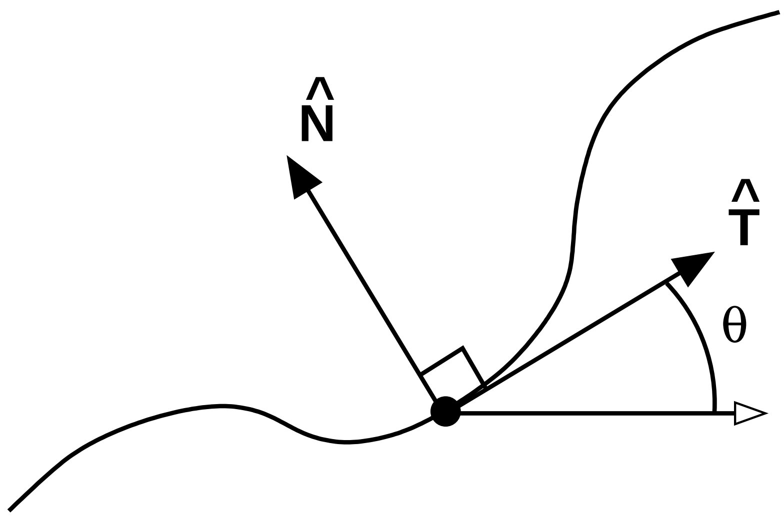
- 2D rotations $\leftrightarrow$ *complex numbers.*
- Why? $e^{i\theta}(x + iy) = (x' + iy')$

$$\begin{aligned} x' &= x \cos\theta - y \sin\theta \\ y' &= x \sin\theta + y \cos\theta \end{aligned}$$

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

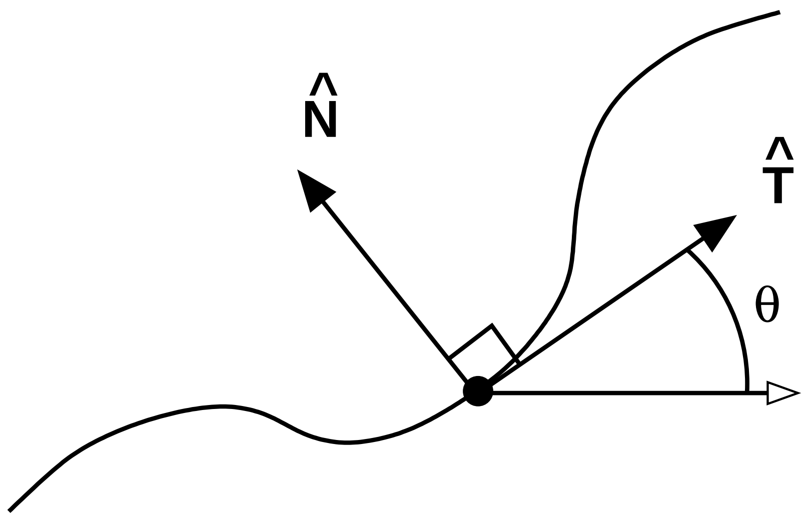
Frames in 2D

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



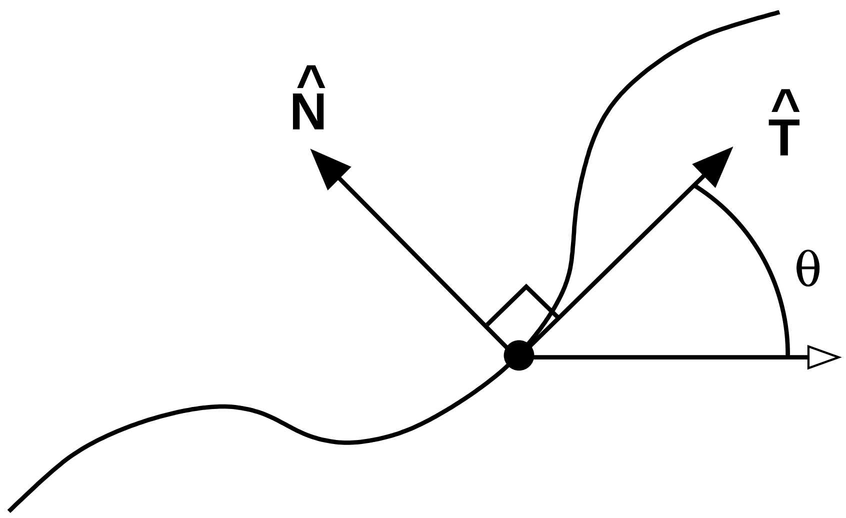
Frames in 2D

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:

$$R_2(\theta) = \left[\; \hat{\mathbf{T}} \;\; \hat{\mathbf{N}} \; \right]$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} .$$

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:

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

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

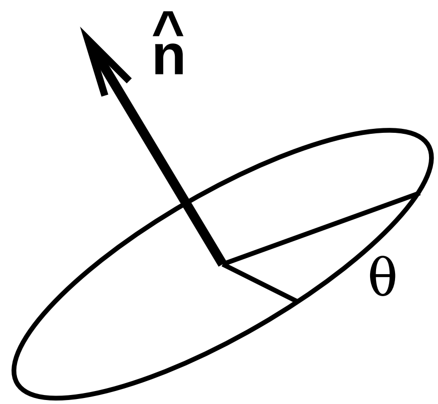
In 3D, repeat our trick: take square root of the frame, but now use *quaternions*:

- **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 $\mathbf{R}(\theta, \hat{\mathbf{n}})$:

$$\begin{bmatrix} a^2 - b^2 + (n_1)^2(2b^2) & 2b^2 n_1 n_2 - 2abn_3 & 2b^2 n_3 n_1 + 2abn_2 \\ 2b^2 n_1 n_2 + 2abn_3 & a^2 - b^2 + (n_2)^2(2b^2) & 2b^2 n_2 n_3 - 2abn_1 \\ 2b^2 n_3 n_1 - 2abn_2 & 2b^2 n_2 n_3 + 2abn_1 & a^2 - b^2 + (n_3)^2(2b^2) \end{bmatrix}$$

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 $R_3(q)$

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

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

$\boxed{q' * q = Q}$ 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 $\vec{\mathbf{q}}$:

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

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: $\mathbf{q} = (q_x,\, q_y,\, q_z)$

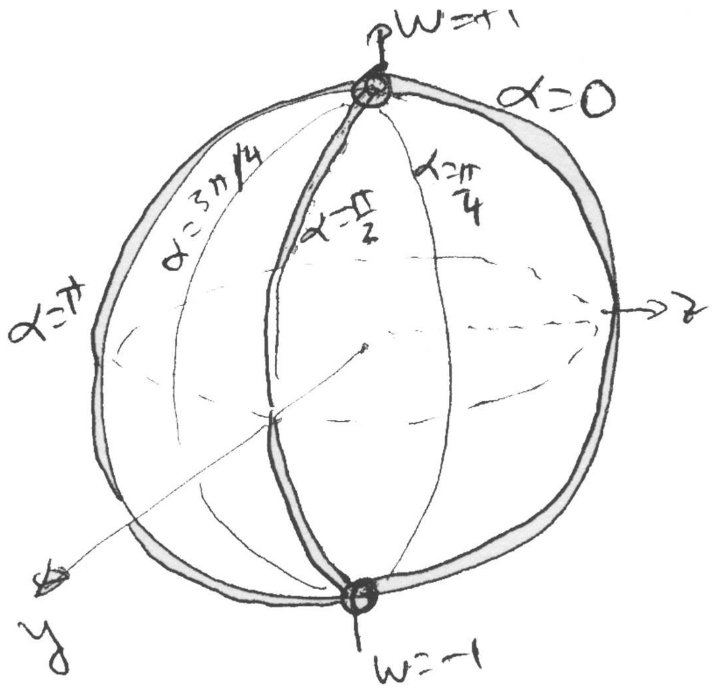
q_0 is KNOWN! We can also use any other triple: the fourth component is *dependent*.

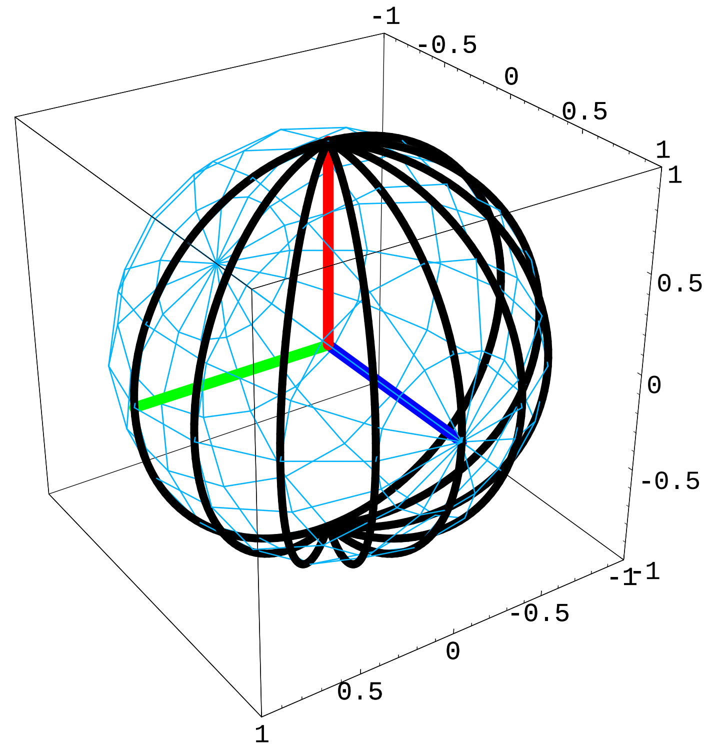
DEMO

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

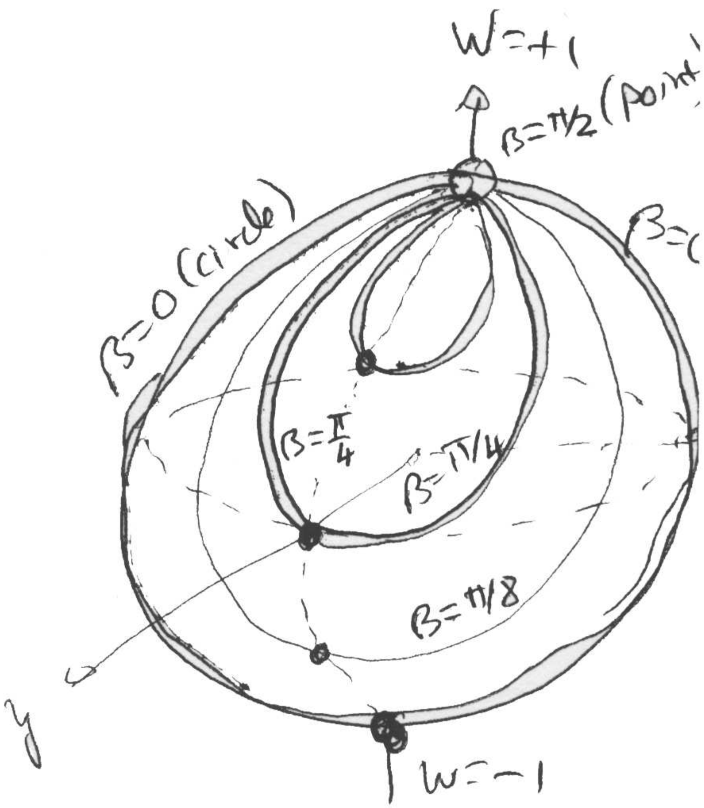
- 360° **Belt Trick in Quaternion Form.** DEMO:
- 720° **Belt Trick in Quaternion Form.**
- **Rolling Ball in Quaternion Form.** DEMO:
- **Gimbal Lock in Quaternion Form.**

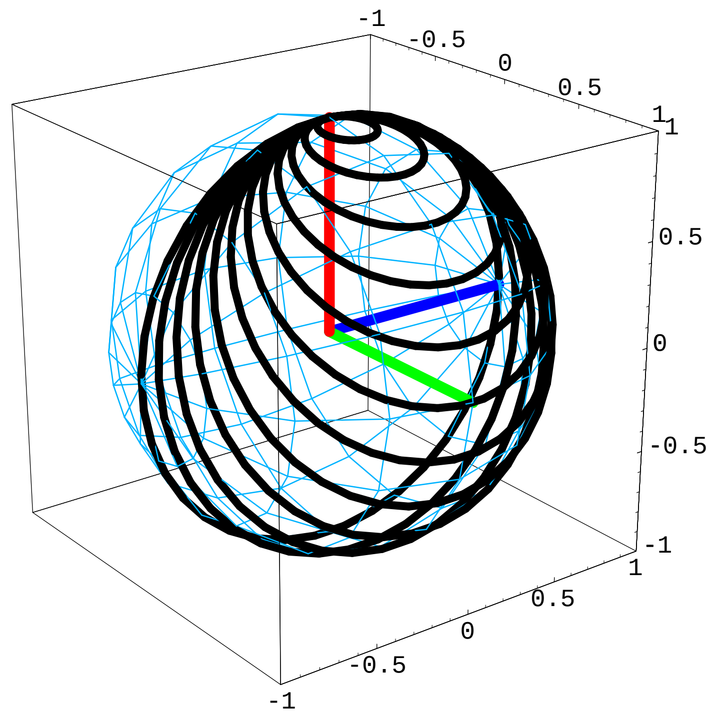
360° **Belt Trick in Quaternion Form**



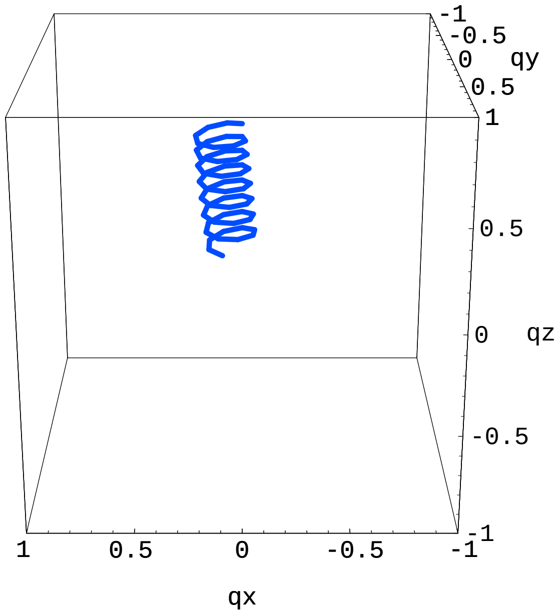


720° Belt Trick in Quaternion Form

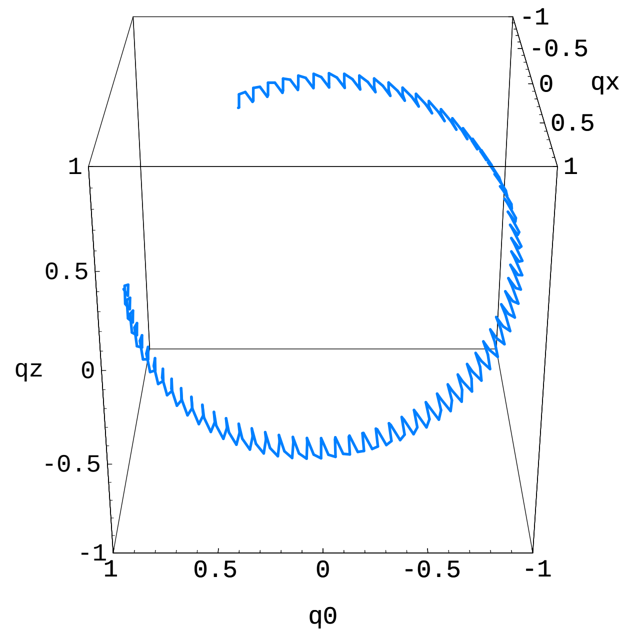




Rolling Ball in Quaternion Form

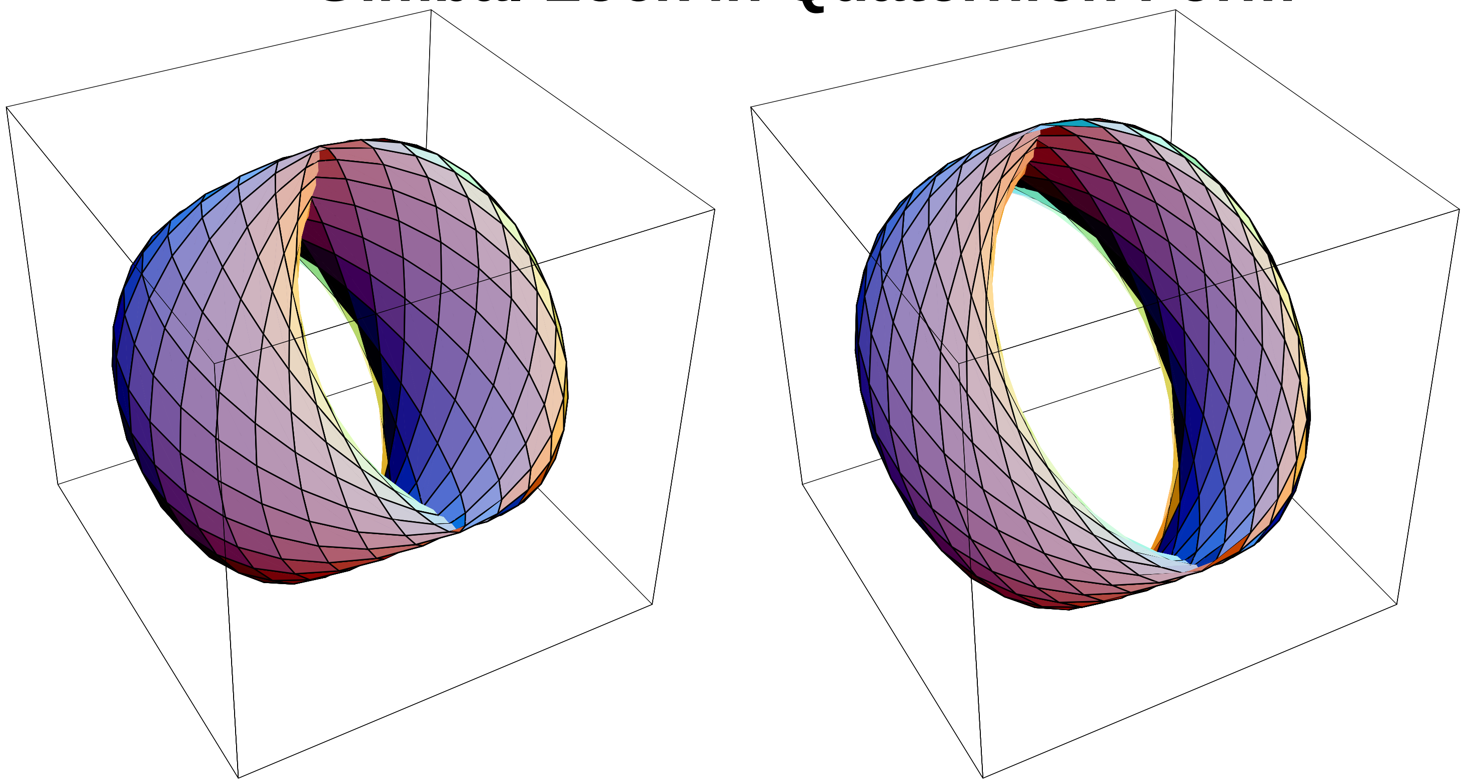


$\mathbf{q}$ vector-only plot.



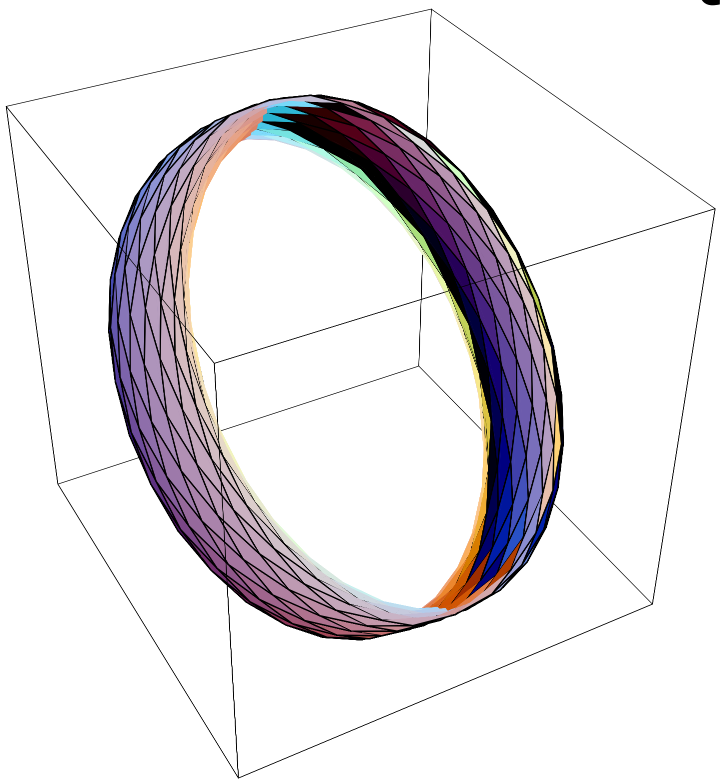
(q_0, q_x, q_z) plot

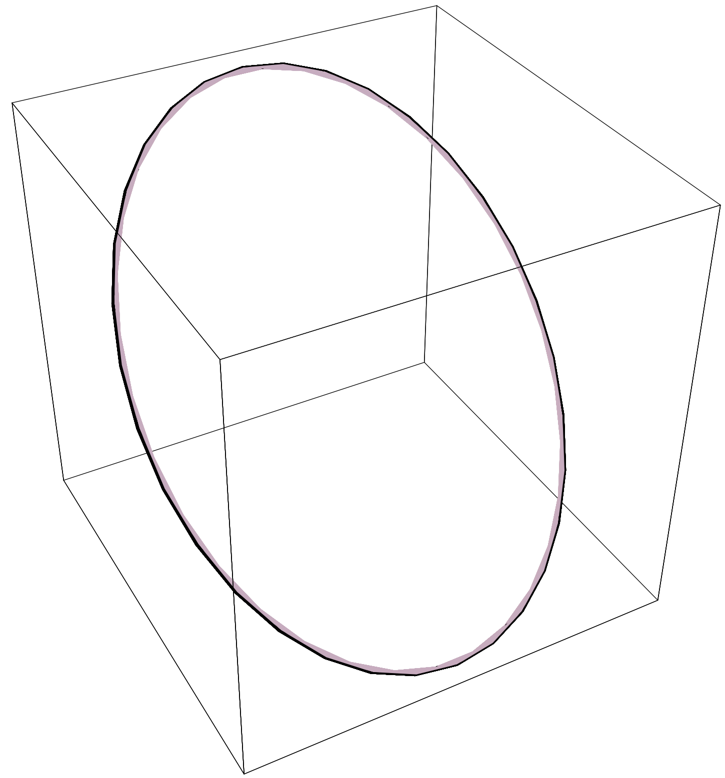
Gimbal Lock in Quaternion Form



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

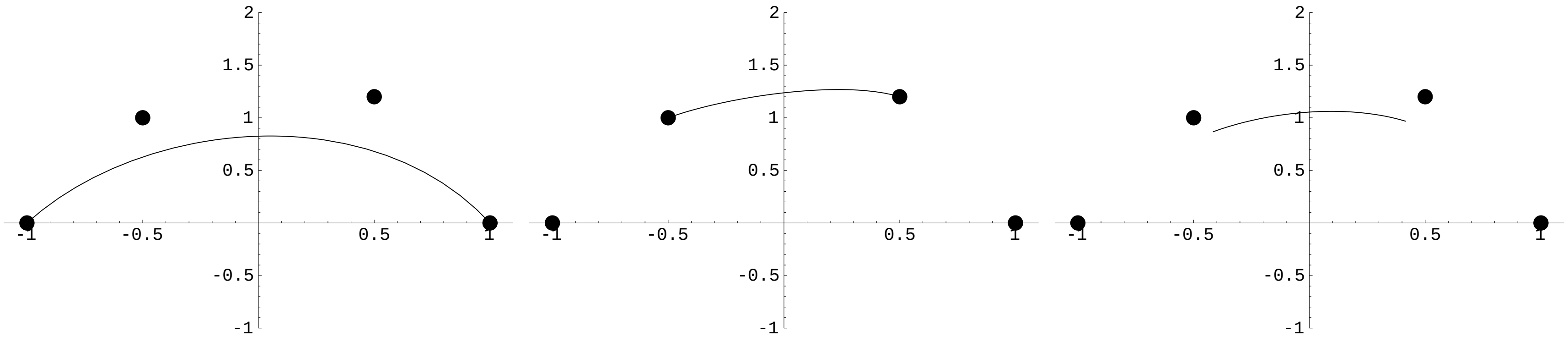
General quaternion spherical interpolation employs the “SLERP,” a constant angular velocity transition between two directions, $\hat{\mathbf{q}}_1$ and $\hat{\mathbf{q}}_2$:

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

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

Plane Interpolations

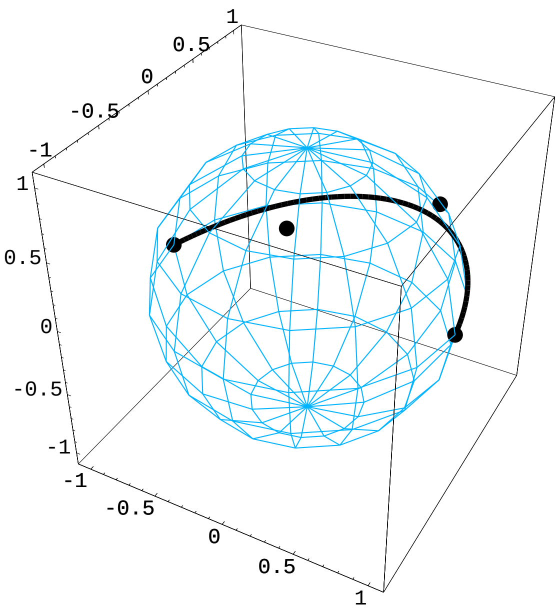
In Euclidean space, these three basic cubic splines look like this:



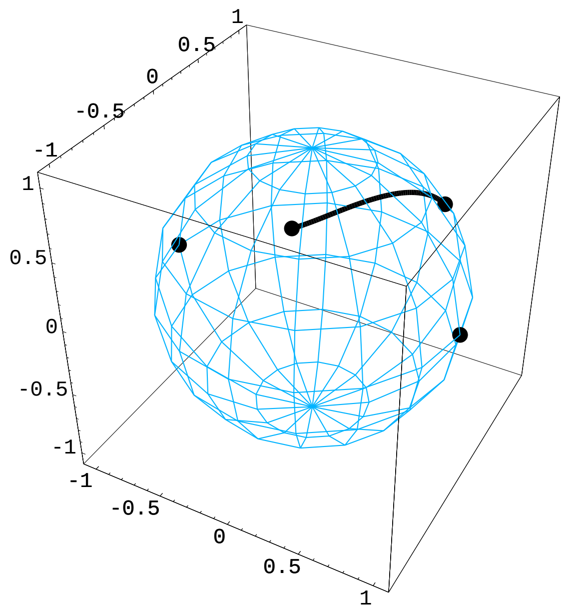
Bezier Catmull-Rom Uniform B

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*.

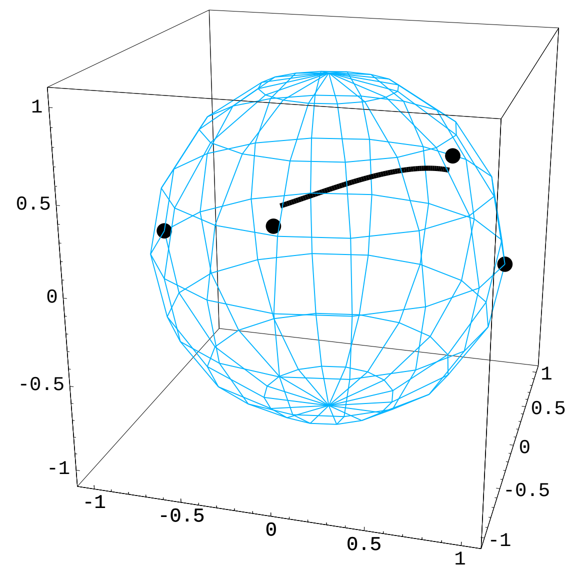
Spherical Interpolations



Bezier

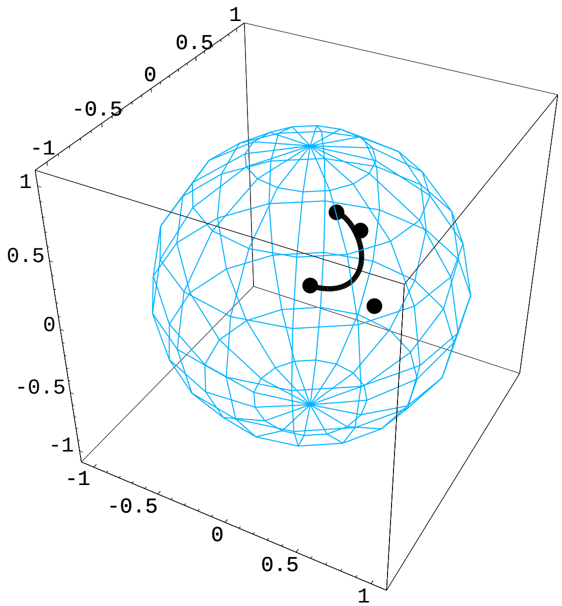


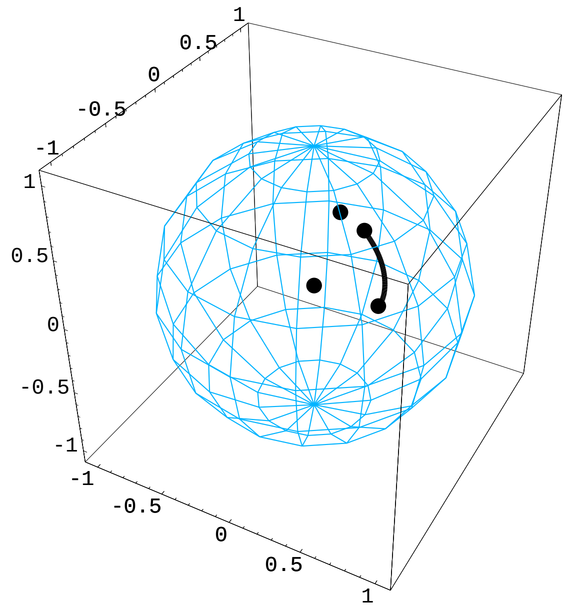
Catmull-Rom

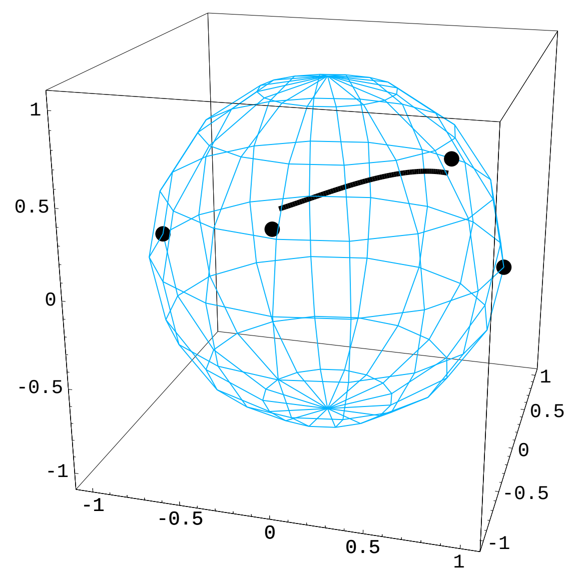


Uniform B

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),

Key to Quaternion Intuition

Fundamental Intuition: We know

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

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

Therefore

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

Summarize Quaternion Properties

- **Unit four-vector.** Take $q = (q_0, q_1, q_2, q_3) = (q_0, \vec{\mathbf{q}})$ to obey constraint $q \cdot q = 1$.

- **Multiplication rule.** The quaternion product q and p is $q * p = (q_0 p_0 - \vec{\mathbf{q}} \cdot \vec{\mathbf{p}},\ q_0 \vec{\mathbf{p}} + p_0 \vec{\mathbf{q}} + \vec{\mathbf{q}} \times \vec{\mathbf{p}})$, or, alternatively,

$$\begin{bmatrix} [q*p]_0 \\ [q*p]_1 \\ [q*p]_2 \\ [q*p]_3 \end{bmatrix} = \begin{bmatrix} q_0 p_0 - q_1 p_1 - q_2 p_2 - q_3 p_3 \\ q_0 p_1 + q_1 p_0 + q_2 p_3 - q_3 p_2 \\ q_0 p_2 + q_2 p_0 + q_3 p_1 - q_1 p_3 \\ q_0 p_3 + q_3 p_0 + q_1 p_2 - q_2 p_1 \end{bmatrix}$$

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

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

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

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.**

Visualizing Quaternions

Andrew J. Hanson
Computer Science Department
Indiana University

Siggraph 2007 Tutorial

OUTLINE

I: (50 min) **Twisting Belts, Rolling Balls, and Locking Gimbals:**
Explaining Rotation Sequences with Quaternions

II: (50 min) **Quaternion Fields:**
Curves, Surfaces, and Volumes

Part I

Twisting Belts, Rolling Balls, and Locking Gimbals

Explaining Rotation Sequences with Quaternions

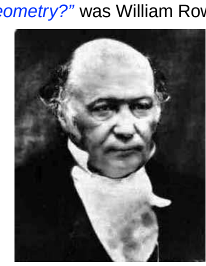
Where Did Quaternions Come From?

. . . 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

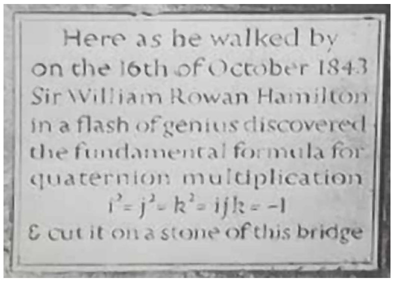
Hamilton's epiphany: 16 October 1843

"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



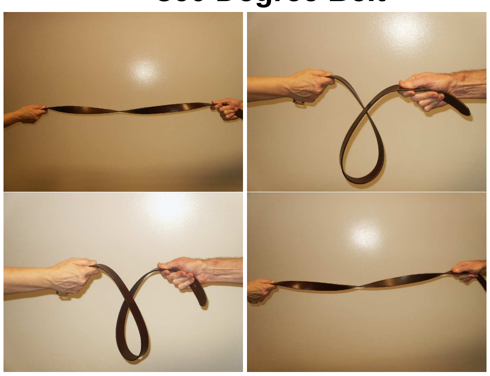
The plaque on Broome Bridge in Dublin, Ireland, commemorating the legendary location where Hamilton conceived of the idea of quaternions. (Hamilton apparently misspelled it as "Brougham Bridge" in his letter.)

The Belt Trick

Quaternion Geometry in our daily lives

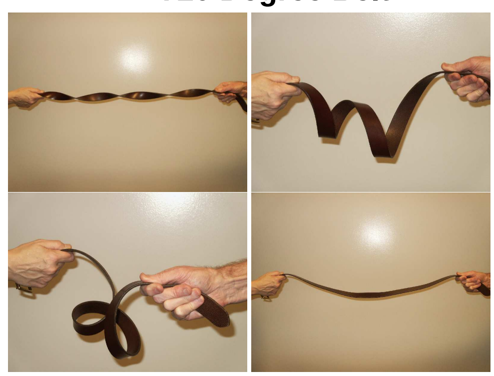
- Two people hold ends of a belt.
- Twist the belt either 360 degrees or 720 degrees.
- **Rule:** *Move belt ends any way you like but do not change orientation of either end.*
- Try to straighten out the belt.

360 Degree Belt



360 twist: stays twisted, can change DIRECTION!

720 Degree Belt



720 twist: CAN FLATTEN OUT WHOLE BELT!

The Beltless Trick

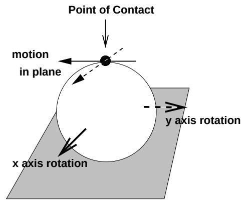
Quaternion Geometry is right in your hand!

- 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* $\longrightarrow$
   equator goes counterclockwise
6. *small counterclockwise circles* $\longrightarrow$
   equator goes clockwise

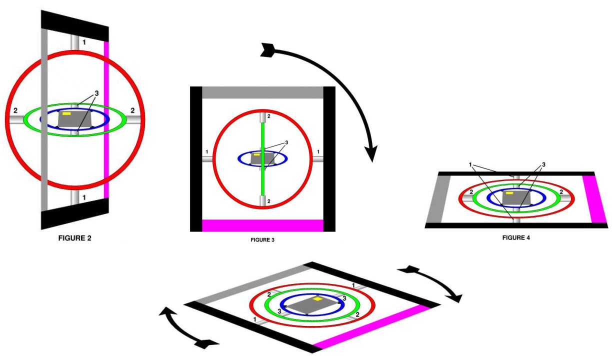
Rolling Ball Scenario



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

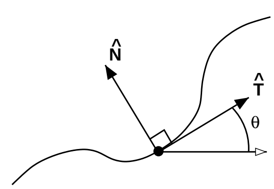
- 2D rotations $\leftrightarrow$ *complex numbers.*
- Why? $e^{i\theta}(x + iy) = (x' + iy')$

$$\begin{aligned} x' &= x \cos\theta - y \sin\theta \\ y' &= x \sin\theta + y \cos\theta \end{aligned}$$

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

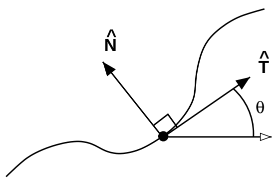
Frames in 2D

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



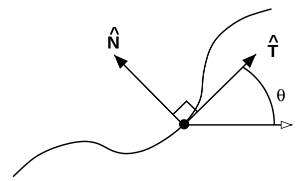
Frames in 2D

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:

$$R_2(\theta) = \left[\, \hat{\mathbf{T}} \ \ \hat{\mathbf{N}} \,\right]$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} .$$

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:

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

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 $(a + ib) = \exp\left(i\,\theta/2\right)$ we see that rotation is complex multiplication!

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

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

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

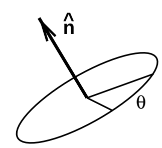
In 3D, repeat our trick: take square root of the frame, but now use *quaternions*:

- **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 :

$$\begin{bmatrix} c + (n_1)^2(1-c) & n_1n_2(1-c) - sn_3 & n_3n_1(1-c) + sn_2 \\ n_1n_2(1-c) + sn_3 & c + (n_2)^2(1-c) & n_3n_2(1-c) - sn_1 \\ n_1n_3(1-c) - sn_2 & n_2n_3(1-c) + sn_1 & c + (n_3)^2(1-c) \end{bmatrix}$$

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 $\mathbf{R}(\theta, \hat{\mathbf{n}})$:

$$\begin{bmatrix} a^2 - b^2 + (n_1)^2(2b^2) & 2b^2n_1n_2 - 2abn_3 & 2b^2n_3n_1 + 2abn_2 \\ 2b^2n_1n_2 + 2abn_3 & a^2 - b^2 + (n_2)^2(2b^2) & 2b^2n_2n_3 - 2abn_1 \\ 2b^2n_3n_1 - 2abn_2 & 2b^2n_2n_3 + 2abn_1 & a^2 - b^2 + (n_3)^2(2b^2) \end{bmatrix}$$

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 $R_3(q)$

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

37

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*.

38

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:

39

Quaternions and Rotations . . .

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

$$\begin{bmatrix} Q_0 = [q' * q]_0 \\ Q_1 = [q' * q]_1 \\ Q_2 = [q' * q]_2 \\ Q_3 = [q' * q]_3 \end{bmatrix} = \begin{bmatrix} q'_0q_0 - q'_1q_1 - q'_2q_2 - q'_3q_3 \\ q'_0q_1 + q'_1q_0 + q'_2q_3 - q'_3q_2 \\ q'_0q_2 + q'_2q_0 + q'_3q_1 - q'_1q_3 \\ q'_0q_3 + q'_3q_0 + q'_1q_2 - q'_2q_1 \end{bmatrix}$$

This is Quaternion Multiplication.

40

Algebra of Quaternions
= 3D Rotations!

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

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

41

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:

$$\begin{aligned} q' * q \ = \ (&q'_0q_0 - q'_1q_1 - q'_2q_2 - q'_3q_3, \\ &q'_0q_1 + q'_1q_0 + q'_2q_3 - q'_3q_2, \\ &q'_0q_2 + q'_2q_0 + q'_3q_1 - q'_1q_3, \\ &q'_0q_3 + q'_3q_0 + q'_1q_2 - q'_2q_1) \end{aligned}$$

42

Algebra of Quaternions . . .

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

$$q' * q \;=\; (q'_0 q_0 - \vec{\mathrm{q}}' \cdot \vec{\mathrm{q}},\; q'_0\,\vec{\mathrm{q}} + q_0\,\vec{\mathrm{q}}' + \vec{\mathrm{q}}' \times \vec{\mathrm{q}})$$

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: $\mathbf{q} = (q_x,\, q_y,\, q_z)$

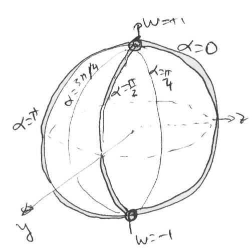
q_0 is KNOWN! We can also use any other triple: the fourth component is *dependent*.

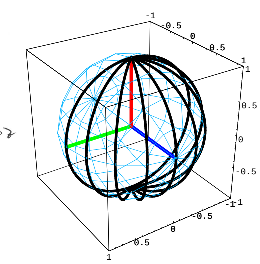
DEMO

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

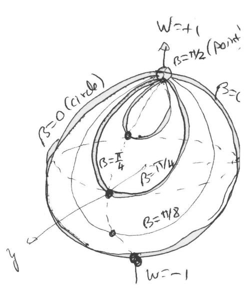
- 360° **Belt Trick in Quaternion Form.** DEMO:
- 720° **Belt Trick in Quaternion Form.**
- **Rolling Ball in Quaternion Form.** DEMO:
- **Gimbal Lock in Quaternion Form.**

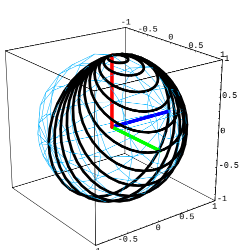
360° **Belt Trick in Quaternion Form**



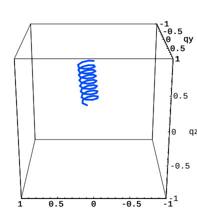


720° **Belt Trick in Quaternion Form**

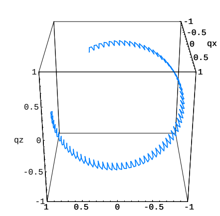




Rolling Ball in Quaternion Form

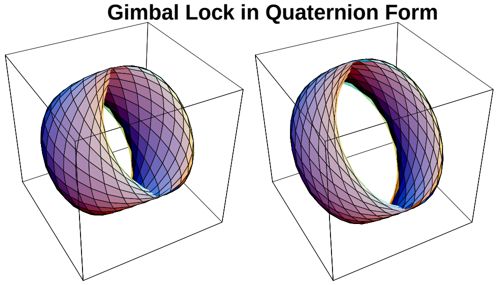


$\mathbf{q}$ vector-only plot.



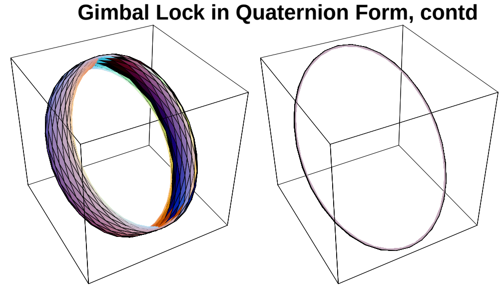
(q_0, q_x, q_z) plot

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** $\phi \to \pi/2$

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

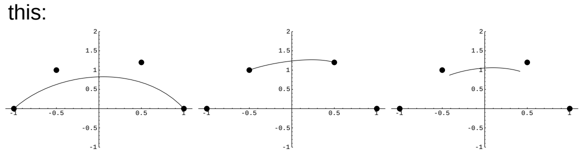
General quaternion spherical interpolation employs the "SLERP," a constant angular velocity transition between two directions, $\hat{q}_1$ and $\hat{q}_2$:

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

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

Plane Interpolations

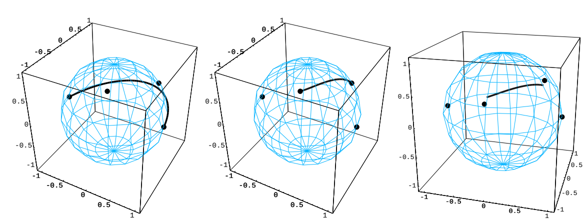
In Euclidean space, these three basic cubic splines look like this:



Bezier Catmull-Rom Uniform B

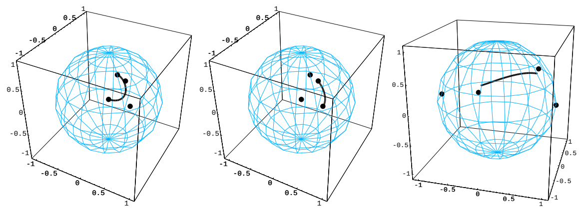
The differences are in the derivatives: Bezier has to start matching all over at every fourth point; Catmull-Rom matches the first derivative; and B-spline is the cadillac, matching all derivatives but *no control points*.

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 $\vec{\mathrm{q}} = \hat{\mathbf{n}}\sin(\theta/2)$ 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),

Key to Quaternion Intuition

Fundamental Intuition: We know

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

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

Therefore

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

Summarize Quaternion Properties

- **Unit four-vector.** Take $q = (q_0, q_1, q_2, q_3) = (q_0, \vec{\mathrm{q}})$ to obey constraint $q \cdot q = 1$.

- **Multiplication rule.** The quaternion product q and p is $q * p = (q_0 p_0 - \vec{\mathrm{q}} \cdot \vec{\mathrm{p}},\ q_0 \vec{\mathrm{p}} + p_0 \vec{\mathrm{q}} + \vec{\mathrm{q}} \times \vec{\mathrm{p}})$,
  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

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

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

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.**

Visualizing Quaternions

Part II

Quaternion Fields

Curves, Surfaces, and Volumes

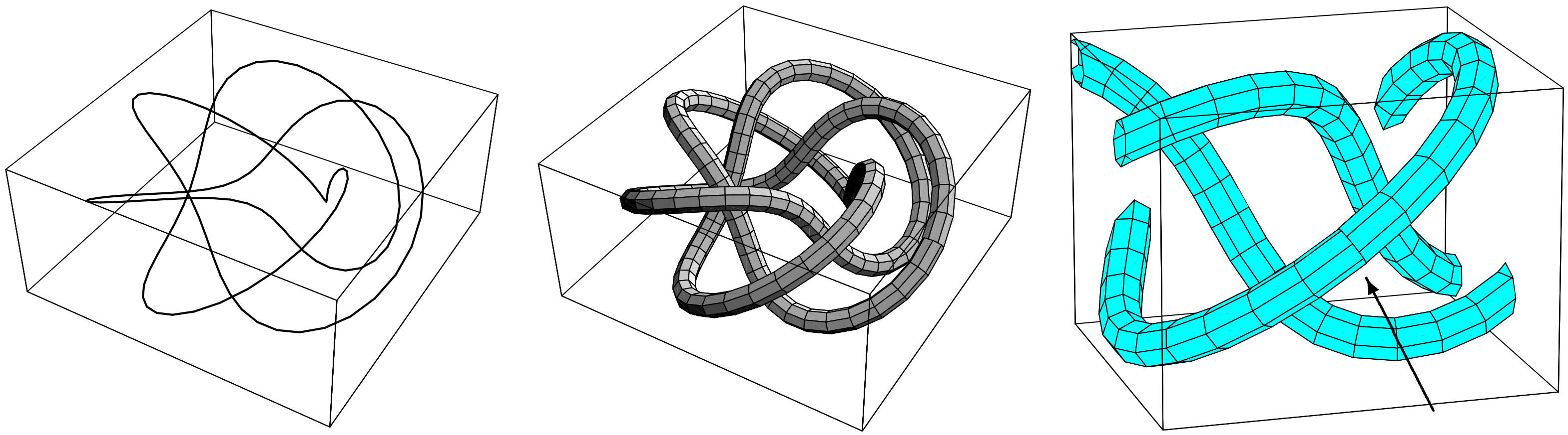
OUTLINE

- **Quaternion Curves:** generalize the Frenet Frame, optimize in quaternion space
- **Quaternion Surfaces:** generalize Gauss map, optimize in quaternion space
- **Quaternion Volumes:** visualize degrees of freedom of a joint

What are Frames used For?

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

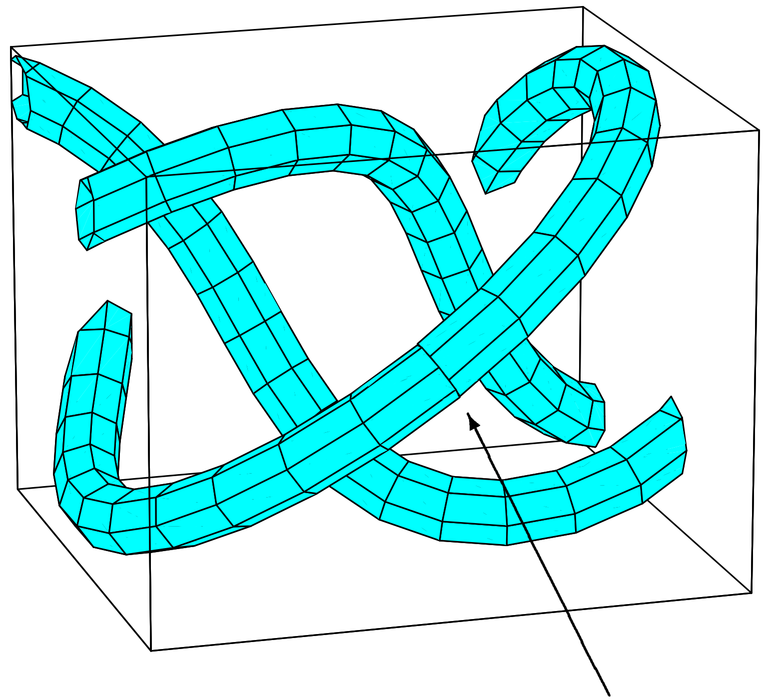
Motivating Problem: Framing Curves



The (3,5) torus knot.

- Line drawing ≈ useless.
- Tubing based on parallel transport, **not periodic.**
- Closeup of the non-periodic mismatch.

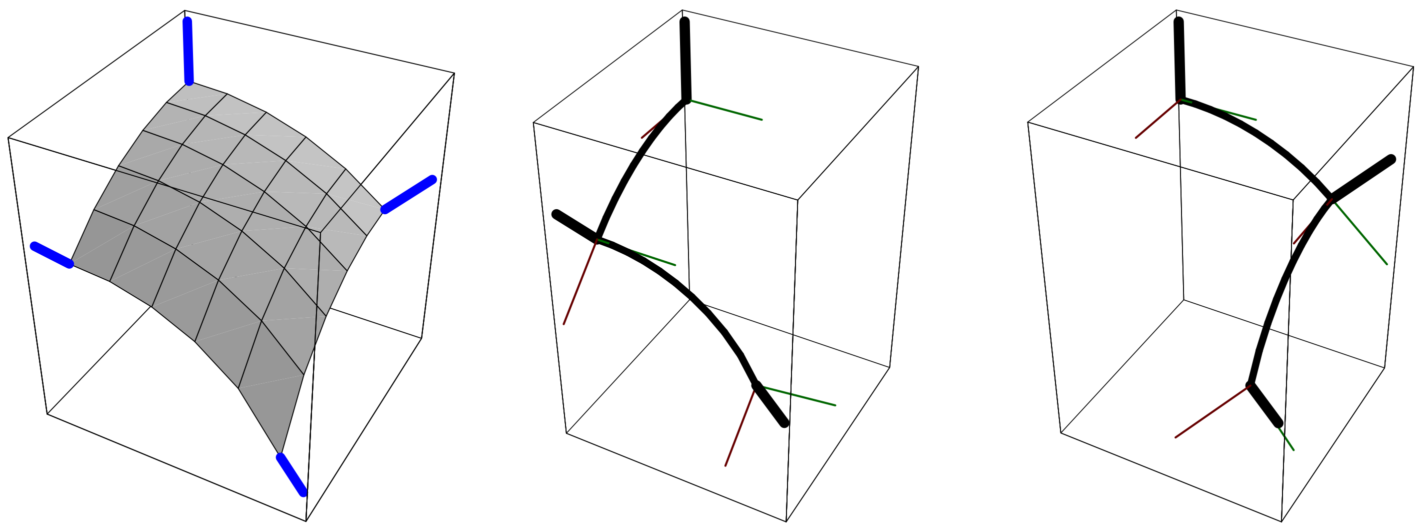
Motivating Problems: Curves



Closeup of the non-periodic mismatch.

Can't apply texture.

Motivating Problems: Surfaces



A smooth 3D surface patch: two ways to get bottom frame.

No unique orthonormal frame is derivable from the parameterization.

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.

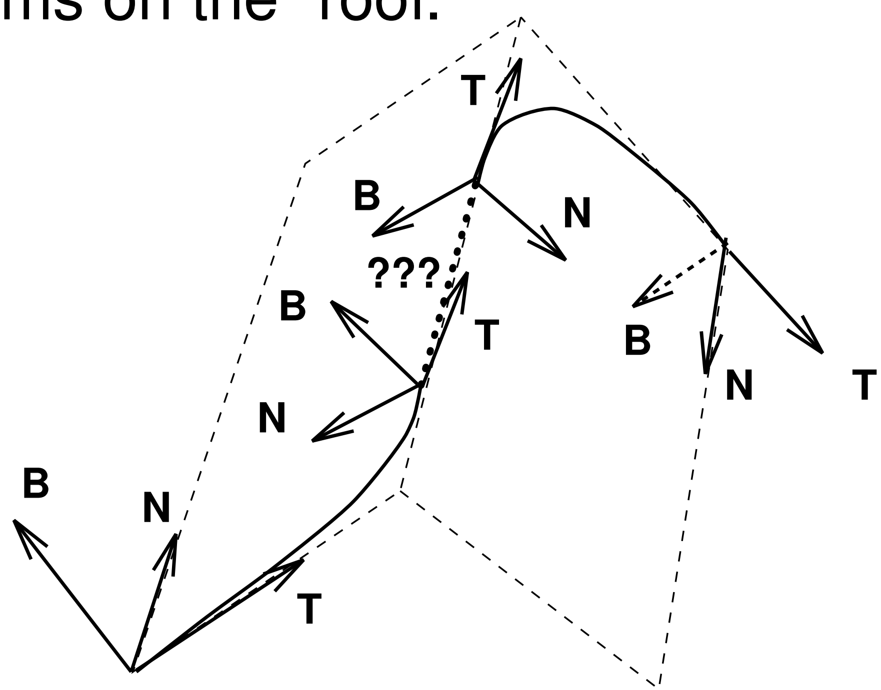
Parallel Transport frame (Bishop): $\sigma = 0$ to get minimal turning. NON-LOCAL = an INTEGRAL.

3D curve frames, contd

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

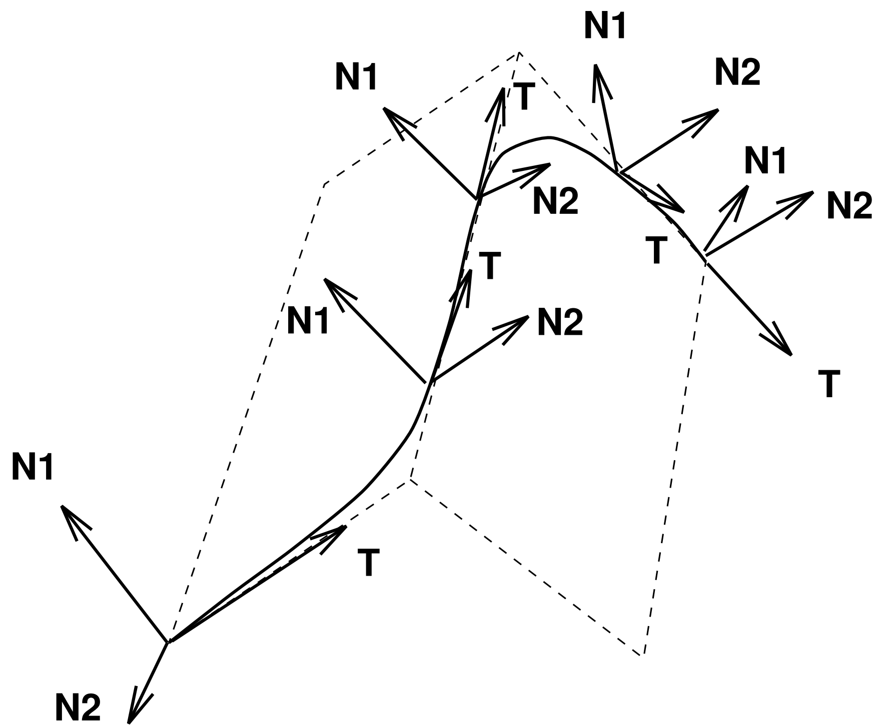
$$\mathrm{B}(t) = \frac{\mathbf{x}'(t) \times \mathbf{x}''(t)}{\|\mathbf{x}'(t) \times \mathbf{x}''(t)\|}$$

but has problems on the “roof.”

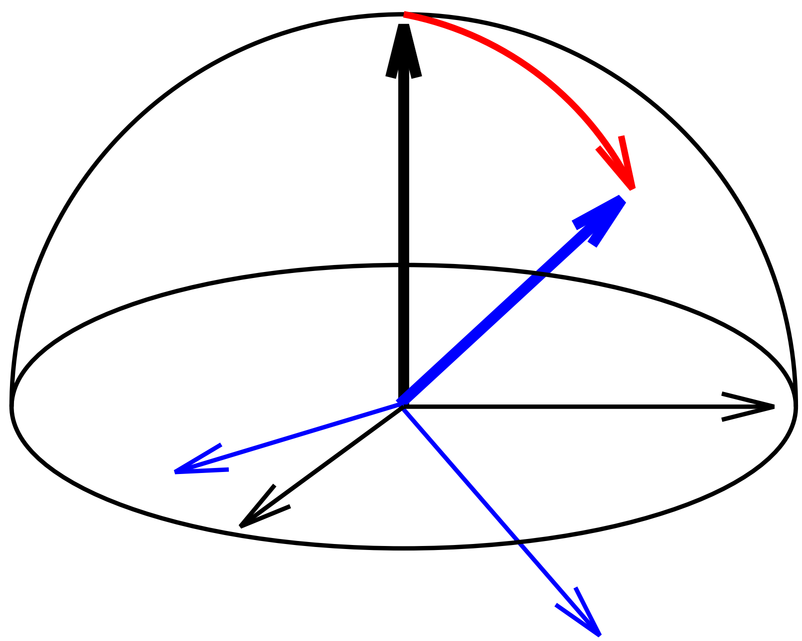


3D curve frames, contd

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*.

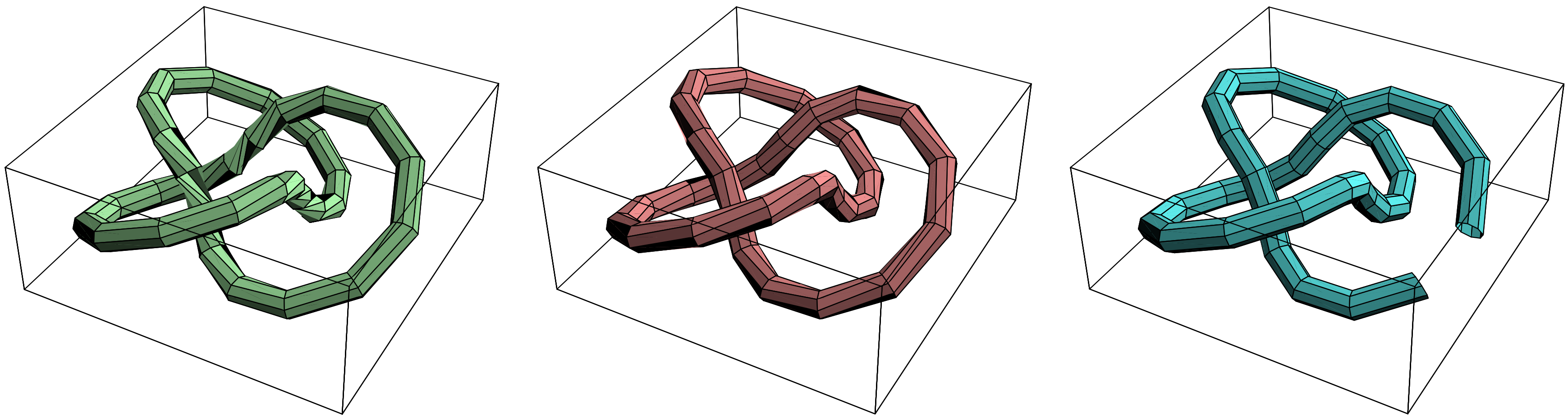
Observations:

- **Tubing and Generalized Cones.** Any of these frames can be used to solve the *tubing problem*.

- **Minimality.** The PT frame appears to be unique frame with *minimum total rotation*. Examine later in quaternion space.

- **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.

Sample Curve Tubings and their Frames



Frenet　　Geodesic Reference　　Parallel Transport

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

3D Frames to Quaternion Frames

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

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

- **Rotation Correspondence.** $q = (\cos\frac{\theta}{2}, \hat{\mathbf{n}}\sin\frac{\theta}{2})$, with $\hat{\mathbf{n}}$ a unit 3-vector, $\hat{\mathbf{n}}\cdot\hat{\mathbf{n}} = 1$. $R(\theta, \hat{\mathbf{n}})$ is usual 3D rotation by θ in the plane perpendicular to $\hat{\mathbf{n}}$.

- **Extract quaternion:** Either directly from sequence of quaternion multiplications, or indirectly from $R_3(q)$.

Quaternion Frame Evolution

Just as in 2D, let columns of $R_3(q)$ be a **9-part frame**: $(\mathbf{T}, \mathbf{N}, \mathbf{B})$.

Derivatives of the i-th column R_i in quaternion coordinates have the form:

$$\boxed{\dot{R}_i = 2W_i \cdot [\dot{q}(t)]}$$

$$\text{e.g. } W_1 = \begin{bmatrix} q_0 & q_1 & -q_2 & -q_3 \\ q_3 & q_2 & q_1 & q_0 \\ -q_2 & q_3 & -q_0 & q_1 \end{bmatrix}$$

where $i = 1, 2, 3$ and rows form mutually orthonormal basis.

Quaternion Frame Evolution . . .

When we simplify by eliminating W_i . . .

we find the *square root* of the 3D frame eqns!

Tait (1890) derived the quaternion equation that makes **all 9 3D frame equations reduce to:** $\boxed{\dot{q} = (1/2)q * (0, \mathbf{k})}$ or:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & k_2 & -k_1 & -\sigma \\ -k_2 & 0 & \sigma & -k_1 \\ k_1 & -\sigma & 0 & -k_2 \\ \sigma & k_1 & k_2 & 0 \end{bmatrix} \cdot \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

Properties of Tait's quaternion frame equations:

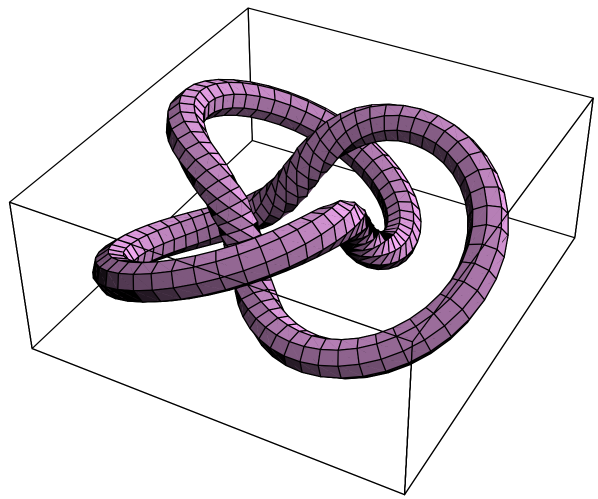
- Antisymmetry $\Rightarrow$ $q(t) \cdot \dot{q}(t) = 0$ as required to keep constant unit radius on 3-sphere.
- *Nine equations and six constraints* become *four equations and one constraint*, keeping quaternion on the 3-sphere. $\Rightarrow$ **Good for computer implementation.**
- MATHEMATICA code implementing this differential equation is provided.

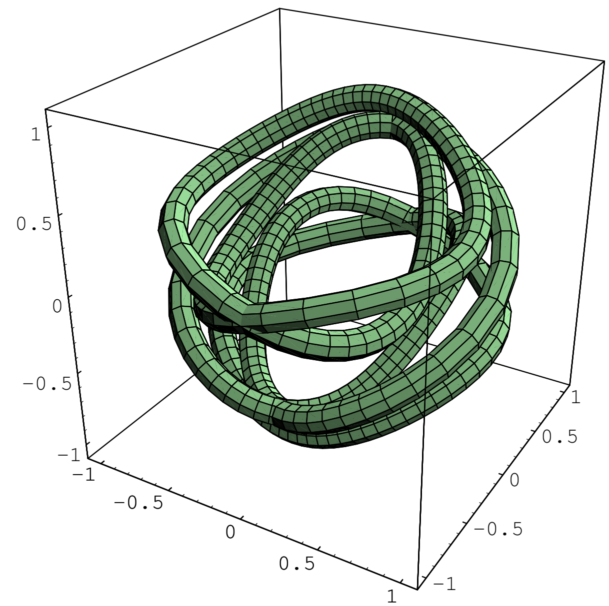
Quaternion Frames . . .

- Analogous treatment (given in Hanson Tech Note in Course Notes) applies also to the Weingarten equations, allowing a direct quaternion treatment of the classical differential geometry of *surfaces* as well.

Example of a Quaternion Frame Curve

Left Curve = torus knot tubed with Frenet frame; Right Curve is projection from 4D of (twice around) quaternion Frenet frames:

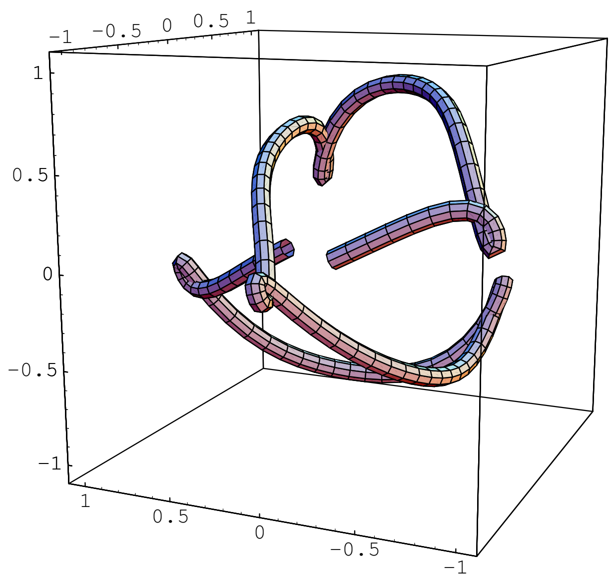


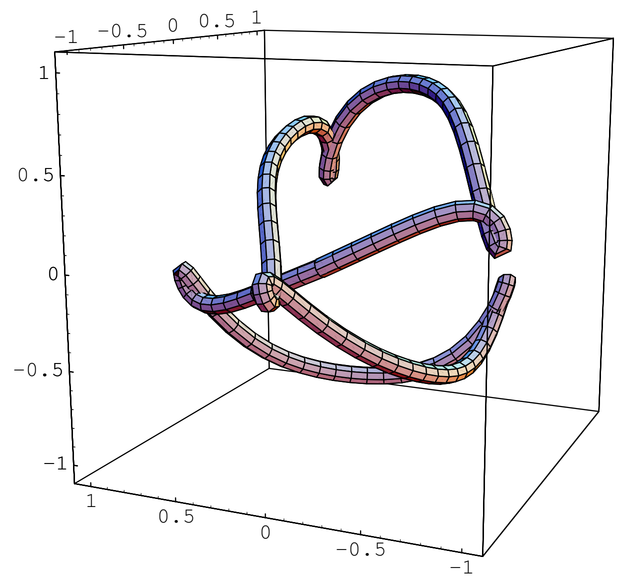


see Notes: Hanson and Ma, “Quaternion Frame Approach to Streamline Visualization,” *IEEE Trans. on Visualiz. and Comp. Graphics*, **1**, No. 2, pp. 164–174 (June, 1995).

Minimizing Quaternion Length Solves Periodic Tube

Quaternion space optimization of the non-periodic parallel transport frame of the (3,5) torus knot.

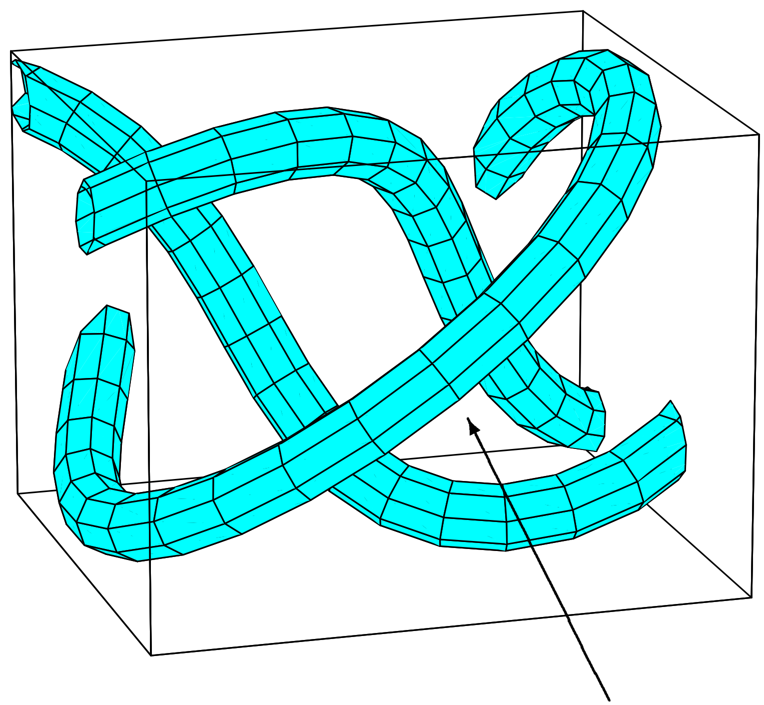




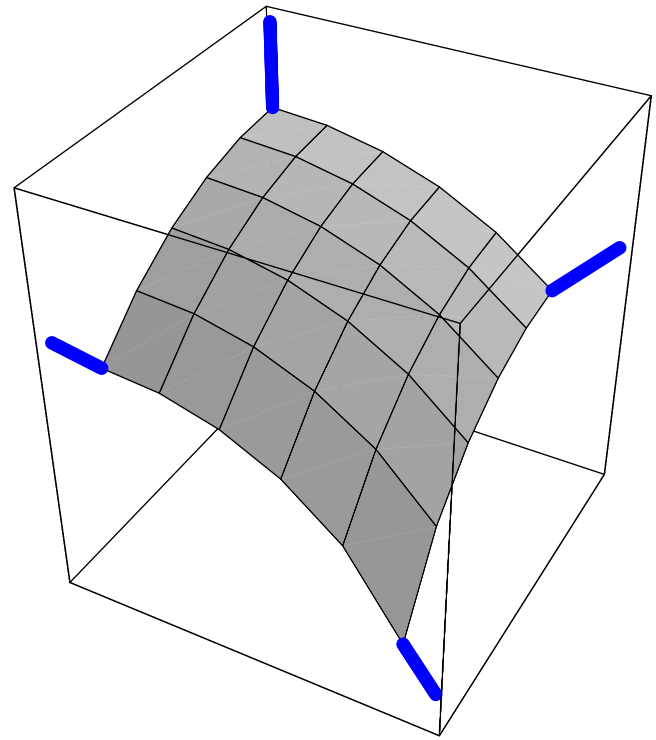
see Notes: "Constrained Optimal Framings of Curves and Surfaces using Quaternion Gauss Maps," *Proceedings of Visualization '98*, pp. 375–382 (1998).

Minimizing Quaternion Length Works

Result of Quaternion space optimization of the (3,5) torus knot frame.

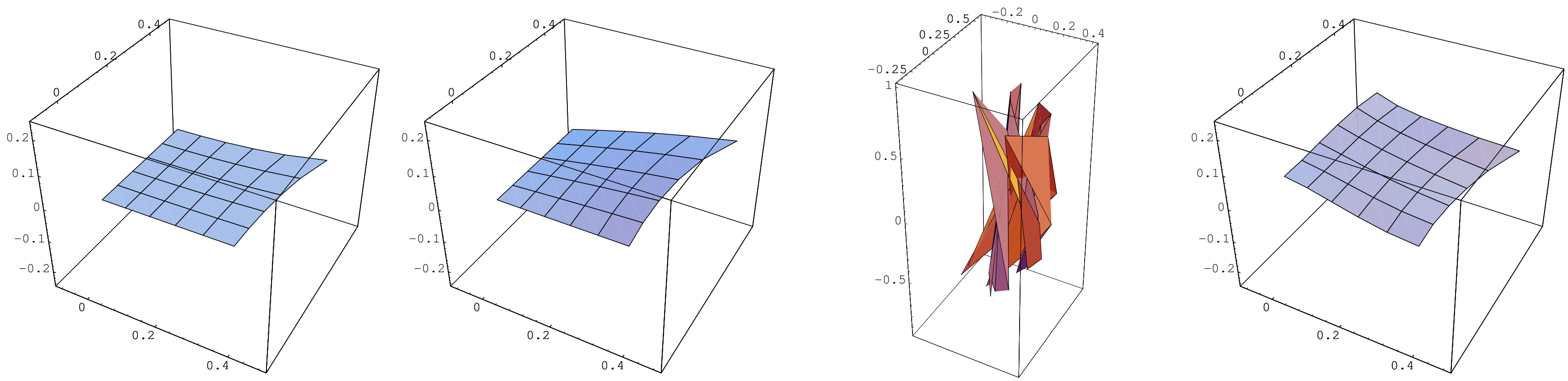


Return to Frames on Surface Patch



Remember: no unique way to disambiguate bottom frame.

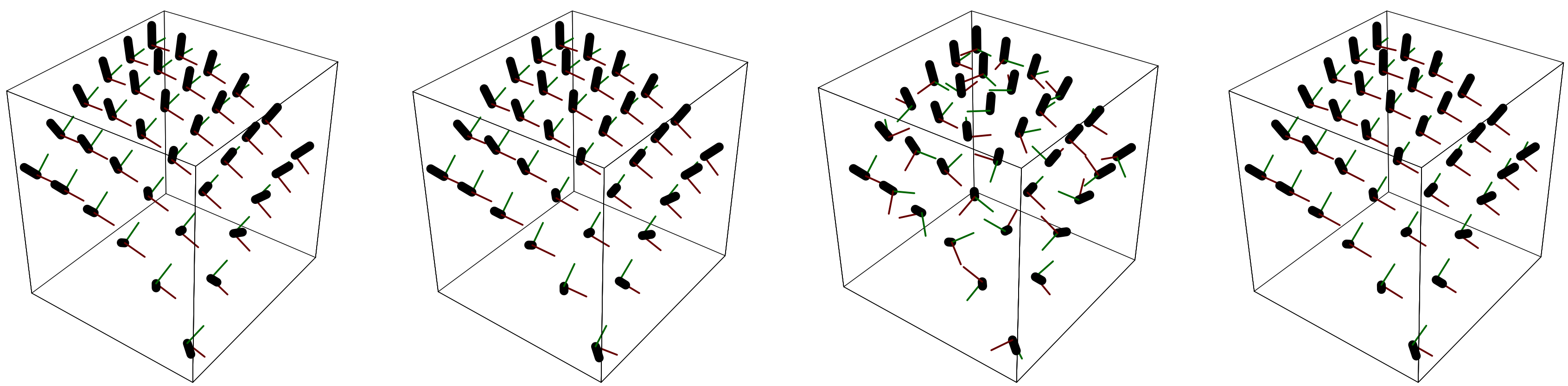
Can also Optimize Quaternion Frames on Patch:



(a) (b) (c) (d)

Quaternion frames for (a) Geodesic Ref. (b) One edge Parallel Transport. (c) Random. (d) Minimal area result.

3D Frames for Patch



(a) (b) (c) (d)

Quaternion frames for (a) Geodesic Ref. (b) One edge Parallel Transport. (c) Random. (d) Minimal area result.

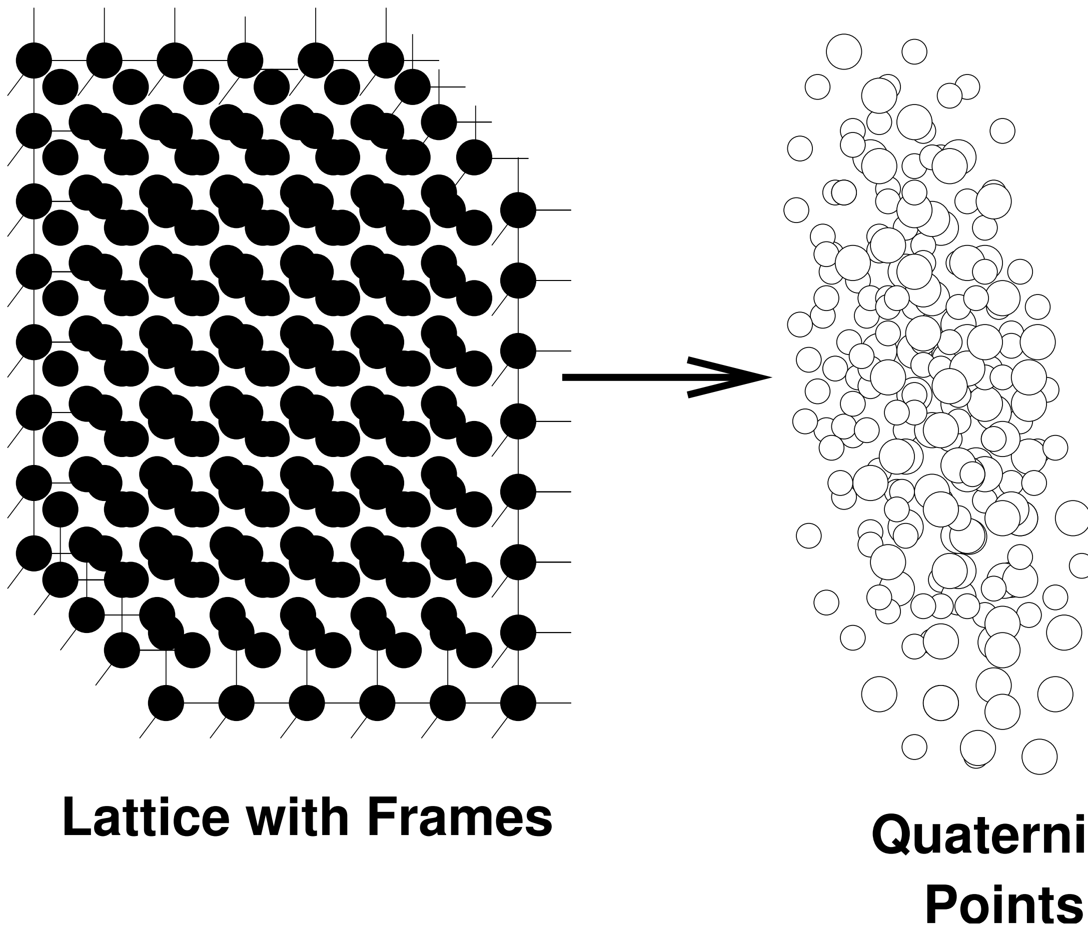
Quaternion Volumes

Last possible orientation field = Volumes:

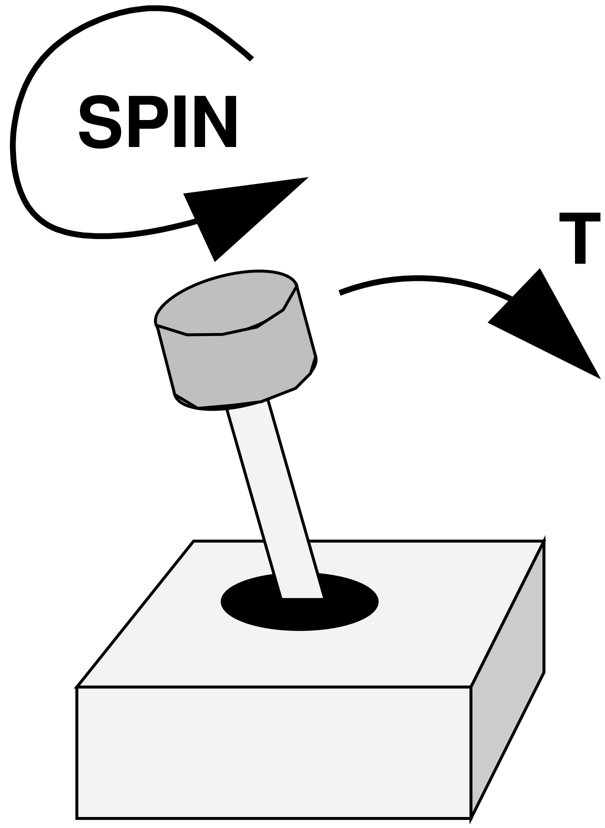
- Collections of oriented objects in a volume.
- 3 degree-of-freedom control monitoring
- 3 degree-of-freedom biological and robotic joints

$\Rightarrow$ all map to **Quaternion Volumes**

Quaternion Volumes

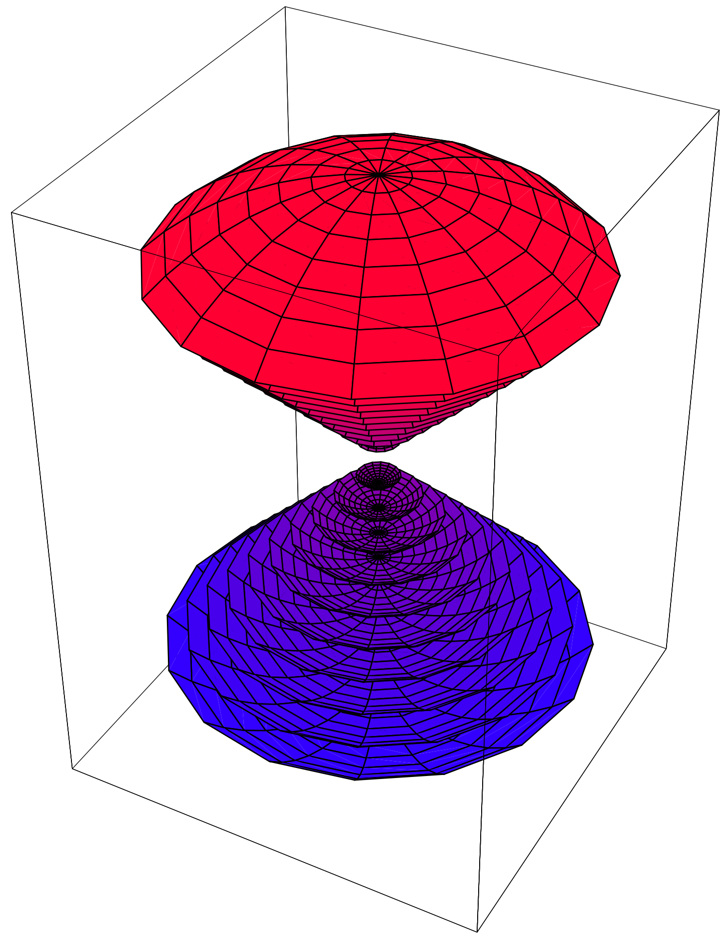


Joystick as Quaternion volume



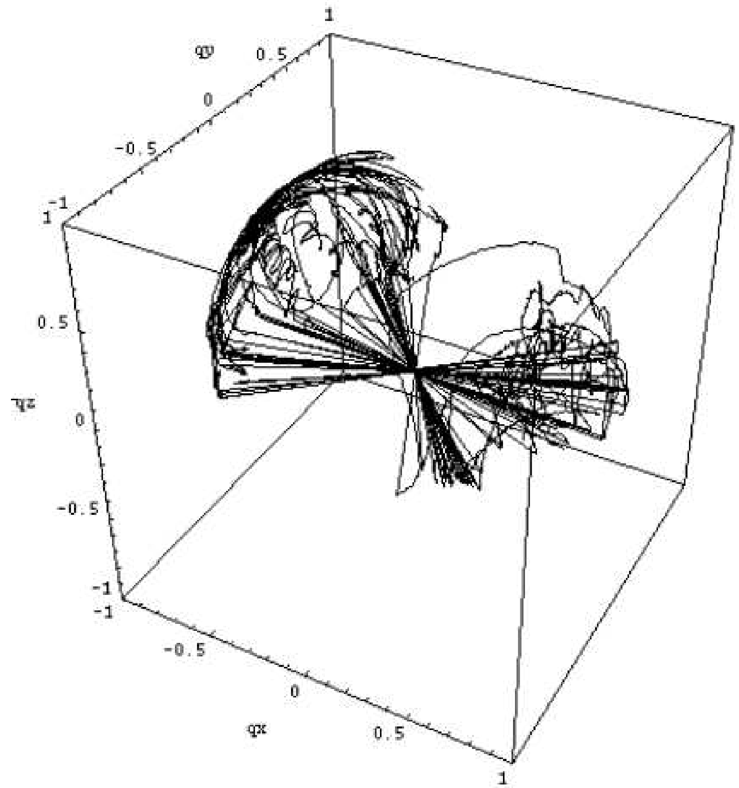
Motion of joystick maps to quaternion volume.

Joystick as quaternion volume



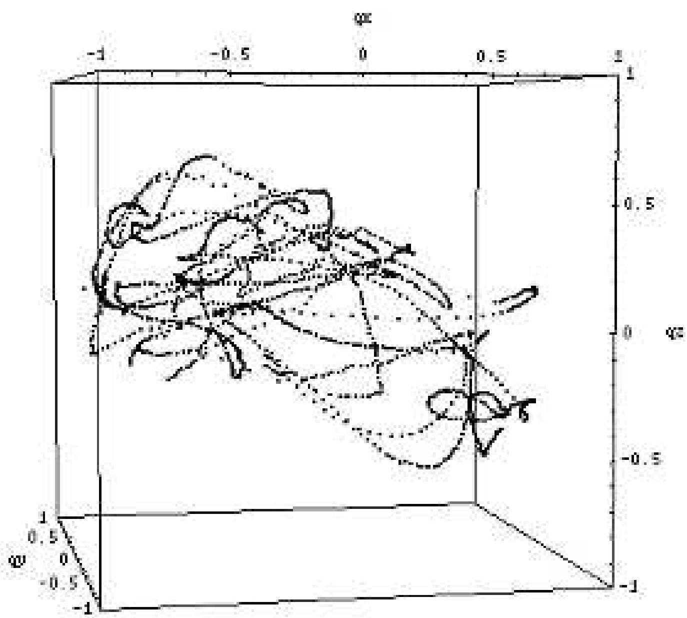
"Solid cone" describes the joystick access space as a quaternion volume

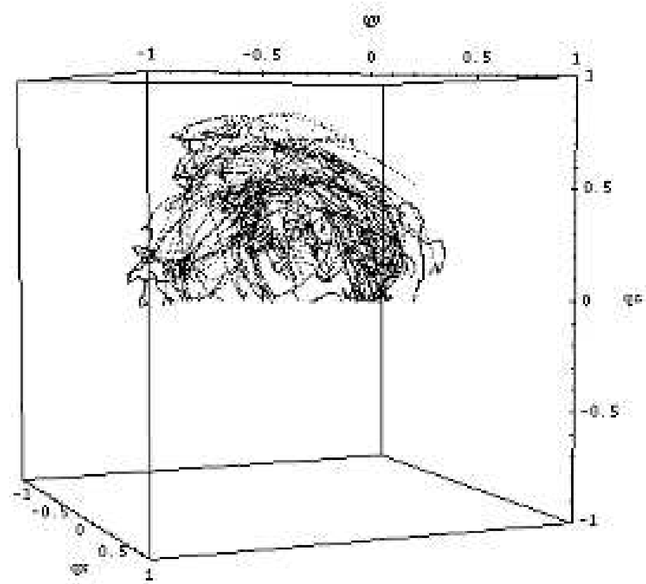
Quaternion volumes: Shoulder data



Quaternion shoulder joint data before correction for doubling.

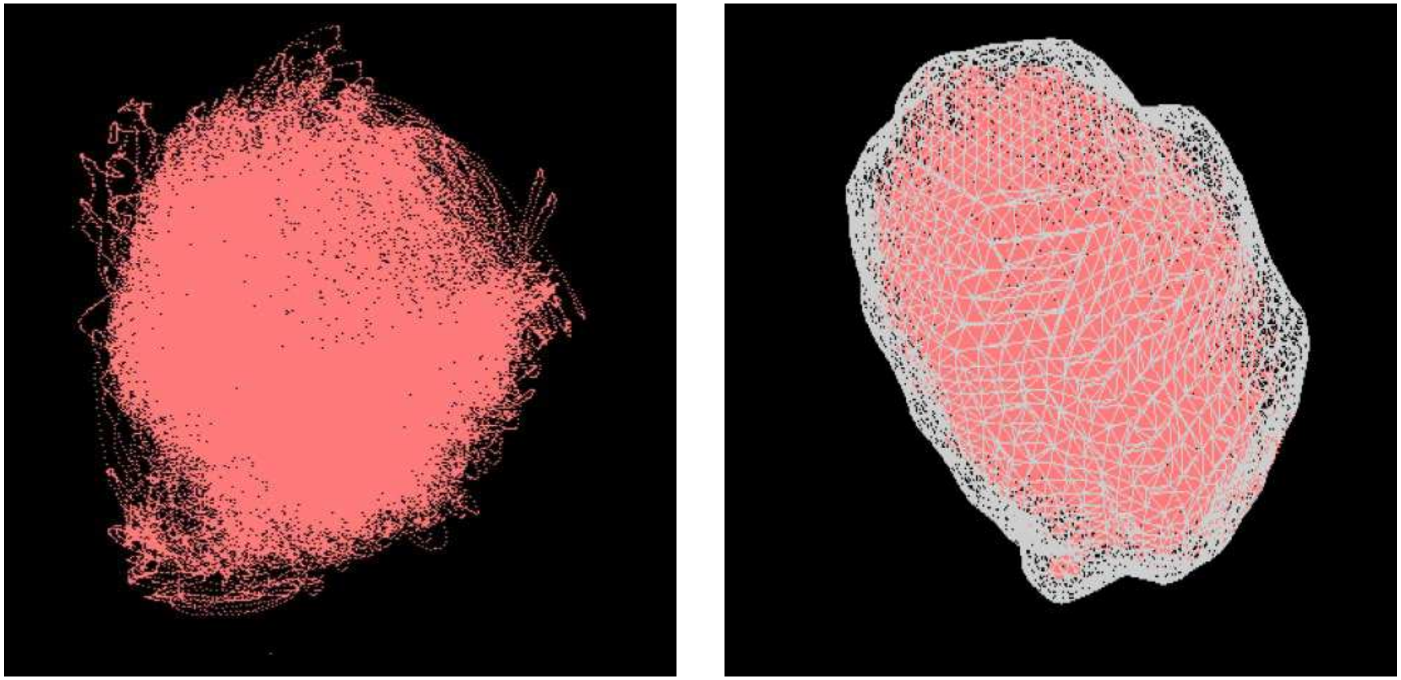
Quaternion volumes: Shoulder data





Shoulder data with neighbors forced to be in same hemisphere of quaternion space as their predecessors.

Quaternion volumes: Shoulder data



(a) (b)

(a) A dense sample of shoulder orientation data in quaternion space.

(b) Implicit surface model fitted to the data. (Herda et al.)

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!

SUMMARY

- **Quaternions nicely represent frame sequences.**
- **Curve frames $\Rightarrow$ quaternion curves.**
- **Surface patch frames $\Rightarrow$ quaternion surface patches.** (Tubes, Proteins, . . .)
- **Minimizing quaternion length or area finds parallel transport "minimal turning" set of frames.**
- **Volume sampled frames $\Rightarrow$ quaternion volumes.**

Use Quaternions for Global Picture of any orientation sequence or collection!

Visualizing Quaternions

Part II

Quaternion Fields

Curves, Surfaces, and Volumes

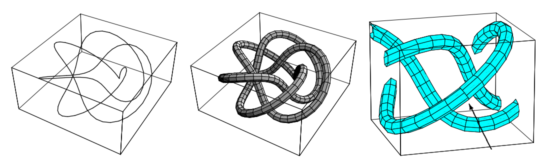
OUTLINE

- **Quaternion Curves:** generalize the Frenet Frame, optimize in quaternion space
- **Quaternion Surfaces:** generalize Gauss map, optimize in quaternion space
- **Quaternion Volumes:** visualize degrees of freedom of a joint

What are Frames used For?

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

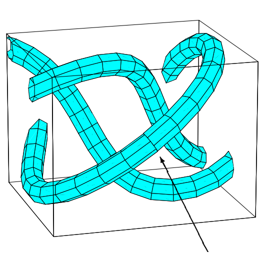
Motivating Problem: Framing Curves



The (3,5) torus knot.

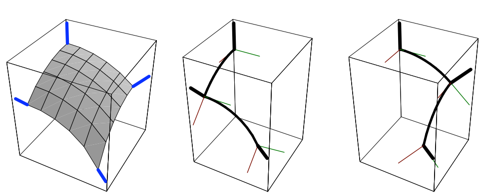
- Line drawing ≈ useless.
- Tubing based on parallel transport, **not periodic.**
- Closeup of the non-periodic mismatch.

Motivating Problems: Curves



Closeup of the non-periodic mismatch.
Can't apply texture.

Motivating Problems: Surfaces



A smooth 3D surface patch: two ways to get bottom frame.

No unique orthonormal frame is derivable from the parameterization.

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.

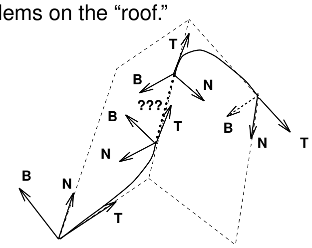
Parallel Transport frame (Bishop): $\sigma = 0$ to get minimal turning. NON-LOCAL = an INTEGRAL.

3D curve frames, contd

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

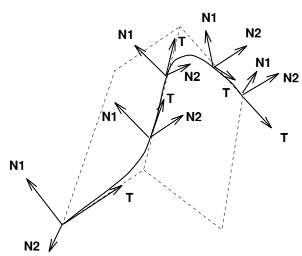
$$\mathbf{B}(t) = \frac{\mathbf{x}'(t) \times \mathbf{x}''(t)}{\|\mathbf{x}'(t) \times \mathbf{x}''(t)\|}$$

but has problems on the "roof."

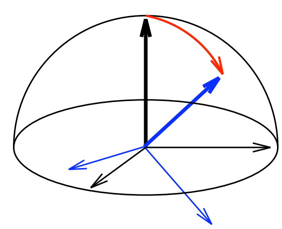


3D curve frames, contd

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).

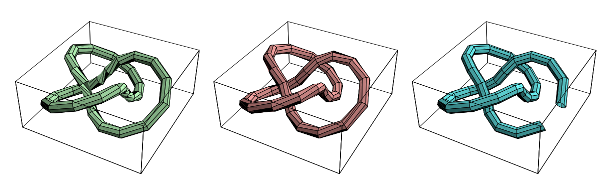
MAIN VALUE: A foolproof reference frame for *sliding rings*.

Frames in 3D, contd

Observations:

- **Tubing and Generalized Cones.** Any of these frames can be used to solve the *tubing problem*.
- **Minimality.** The PT frame appears to be unique frame with *minimum total rotation*. Examine later in quaternion space.
- **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.

Sample Curve Tubings and their Frames



Frenet Geodesic Reference Parallel Transport

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

3D Frames to Quaternion Frames

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

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

- **Rotation Correspondence.** $q = (\cos\frac{\theta}{2}, \hat{\mathbf{n}}\sin\frac{\theta}{2})$, with $\hat{\mathbf{n}}$ a unit 3-vector, $\hat{\mathbf{n}}\cdot\hat{\mathbf{n}} = 1$. $R(\theta, \hat{\mathbf{n}})$ is usual 3D rotation by θ in the plane perpendicular to $\hat{\mathbf{n}}$.
- **Extract quaternion:** Either directly from sequence of quaternion multiplications, or indirectly from $R_3(q)$.

Quaternion Frame Evolution

Just as in 2D, let columns of $R_3(q)$ be a **9-part frame**: $(\mathbf{T}, \mathbf{N}, \mathbf{B})$.

Derivatives of the i-th column R_i in quaternion coordinates have the form:

$$\boxed{\dot{R}_i = 2W_i \cdot [\dot{q}(t)]}$$

$$\text{e.g. } W_1 = \begin{bmatrix} q_0 & q_1 & -q_2 & -q_3 \\ q_3 & q_2 & q_1 & q_0 \\ -q_2 & q_3 & -q_0 & q_1 \end{bmatrix}$$

where $i = 1, 2, 3$ and rows form mutually orthonormal basis.

Quaternion Frame Evolution . . .

When we simplify by eliminating W_i . . .

we find the *square root* of the 3D frame eqns!

Tait (1890) derived the quaternion equation that makes **all 9 3D frame equations reduce to:** $\boxed{\dot{q} = (1/2)q * (0, \mathrm{k})}$ or:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & k_2 & -k_1 & -\sigma \\ -k_2 & 0 & \sigma & -k_1 \\ k_1 & -\sigma & 0 & -k_2 \\ \sigma & k_1 & k_2 & 0 \end{bmatrix} \cdot \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

Quaternion Frames . . .

Properties of Tait's quaternion frame equations:

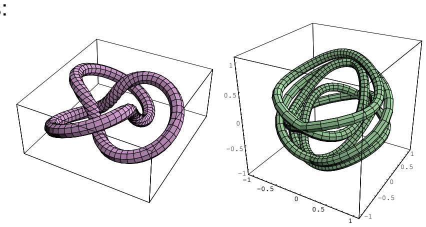
- Antisymmetry $\Rightarrow$ $q(t) \cdot \dot{q}(t) = 0$ as required to keep constant unit radius on 3-sphere.
- *Nine equations and six constraints* become *four equations and one constraint*, keeping quaternion on the 3-sphere. $\Rightarrow$ **Good for computer implementation.**
- MATHEMATICA code implementing this differential equation is provided.

Quaternion Frames . . .

- Analogous treatment (given in Hanson Tech Note in Course Notes) applies also to the Weingarten equations, allowing a direct quaternion treatment of the classical differential geometry of *surfaces* as well.

Example of a Quaternion Frame Curve

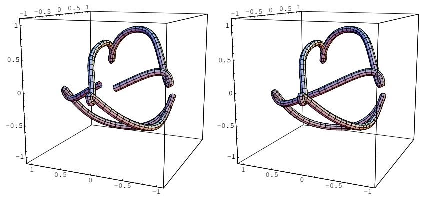
Left Curve = torus knot tubed with Frenet frame; Right Curve is projection from 4D of (twice around) quaternion Frenet frames:



see Notes: Hanson and Ma, "Quaternion Frame Approach to Streamline Visualization," *IEEE Trans. on Visualiz. and Comp. Graphics*, **1**, No. 2, pp. 164–174 (June, 1995).

Minimizing Quaternion Length Solves Periodic Tube

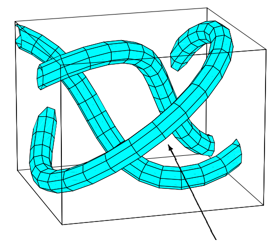
Quaternion space optimization of the non-periodic parallel transport frame of the (3,5) torus knot.



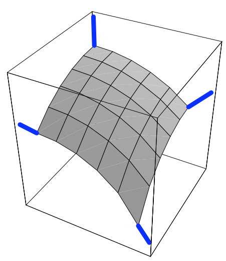
see Notes: "Constrained Optimal Framings of Curves and Surfaces using Quaternion Gauss Maps," *Proceedings of Visualization '98*, pp. 375–382 (1998).

Minimizing Quaternion Length Works

Result of Quaternion space optimization of the (3,5) torus knot frame.

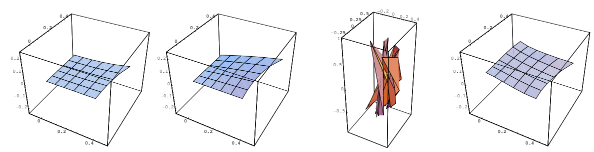


Return to Frames on Surface Patch



Remember: no unique way to disambiguate bottom frame.

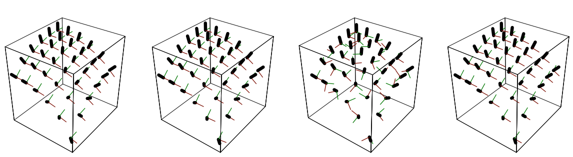
Can also Optimize Quaternion Frames on Patch:



(a) (b) (c) (d)

Quaternion frames for (a) Geodesic Ref. (b) One edge Parallel Transport. (c) Random. (d) Minimal area result.

3D Frames for Patch



(a) (b) (c) (d)

Quaternion frames for (a) Geodesic Ref. (b) One edge Parallel Transport. (c) Random. (d) Minimal area result.

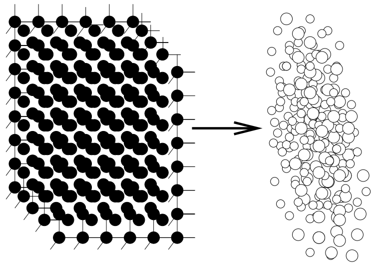
Quaternion Volumes

Last possible orientation field = Volumes:

- Collections of oriented objects in a volume.
- 3 degree-of-freedom control monitoring
- 3 degree-of-freedom biological and robotic joints

$\Rightarrow$ all map to Quaternion Volumes

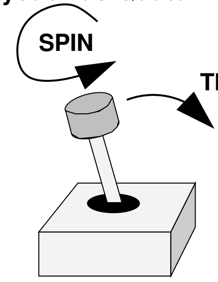
Quaternion Volumes



Lattice with Frames

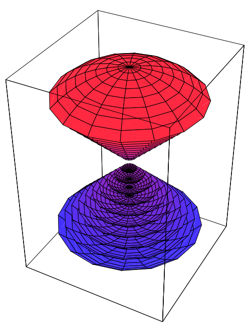
Quaternion Points

Joystick as Quaternion volume



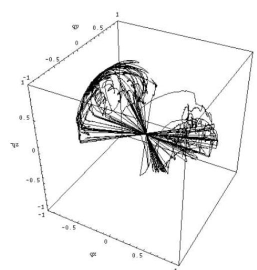
Motion of joystick maps to quaternion volume.

Joystick as quaternion volume



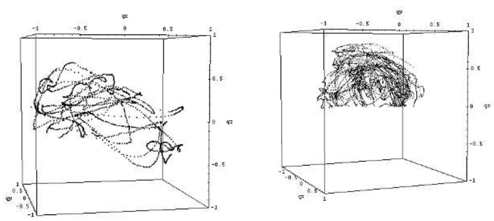
“Solid cone” describes the joystick access space as a quaternion volume

Quaternion volumes: Shoulder data



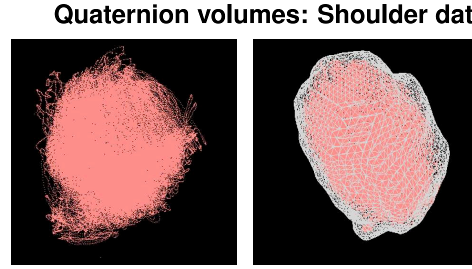
Quaternion shoulder joint data before correction for doubling.

Quaternion volumes: Shoulder data



Shoulder data with neighbors forced to be in same hemisphere of quaternion space as their predecessors.

Quaternion volumes: Shoulder data



(a) (b)

(a) A dense sample of shoulder orientation data in quaternion space.

(b) Implicit surface model fitted to the data. (Herda et al.)

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!

SUMMARY

- **Quaternions nicely represent frame sequences.**
- **Curve frames ⇒ quaternion curves.**
- **Surface patch frames ⇒ quaternion surface patches.** (Tubes, Proteins, ...)
- **Minimizing quaternion length or area finds parallel transport "minimal turning" set of frames.**
- **Volume sampled frames ⇒ quaternion volumes.**

Use Quaternions for Global Picture of any orientation sequence or collection!