

Universidade do Minho
Escola de Ciências

Maria Lúcia Gonçalves Pinheiro

Quaterniões: Cálculo numérico e simbólico



Universidade do Minho
Escola de Ciências

Maria Lúcia Gonçalves Pinheiro

Quaterniões: Cálculo numérico e simbólico

Mestrado em Matemática Computacional

Trabalho efectuado sob a orientação da
Professora Doutora Maria Irene Almeida Falcão

Outubro de 2006

É AUTORIZADA A REPRODUÇÃO PARCIAL DESTA TESE
APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO
ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

Agradecimentos

Começo por agradecer à minha orientadora, Professora Doutora Irene Falcão, cuja energia e determinação foram sempre fonte de admiração e de motivação. Obrigada pelo cuidado, pelas críticas e sugestões, pela disponibilidade e pelo acompanhamento regular com que sempre orientou este trabalho.

À minha princesa, Sofia, que nasceu no dia 12 de Setembro de 2004.

Ao meu marido, Mário, pela paciência, pela atenção e pelo apoio dado ao longo deste projecto.

Aos meus pais e irmã, que me inculcaram a importância da dedicação ao estudo e ao trabalho e que sempre estiveram presentes e disponíveis nos momentos mais difíceis.

Aos meus amigos, pela compreensão dos meus silêncios e ausências, pelas mensagens carinhosas, pela amizade incondicional.

Quaterniões: cálculo numérico e simbólico

Resumo

A descoberta dos quaterniões por Sir Hamilton em 1843 foi motivada pela esperança de criar um tipo de números hipercomplexos que estivesse relacionado com o espaço tridimensional da mesma forma que os números complexos estão relacionados com o plano.

Actualmente, os quaterniões são reconhecidos como uma ferramenta importante para modelar problemas na maioria das ciências aplicadas: visão por computador, robótica, navegação, realidade virtual, etc.

Neste trabalho apresentamos os principais conceitos e propriedades dos quaterniões e descrevemos de que forma os quaterniões podem ser reconhecidos como uma ferramenta única e poderosa para caracterizar rotações no espaço. As contribuições computacionais deste trabalho são as seguintes:

1. QuatCalc: calculadora, desenhada em MATLAB, para realizar operações aritméticas com quaterniões.
2. QuatRot: MATLAB *gui* para visualização de rotações no espaço, usando quaterniões.
3. Uma colecção de rotinas em MATLAB para realizar rotações 3D.

Quaternions: numerical and symbolical calculus

Abstract

The discovery of quaternions by Sir Hamilton in 1843 was motivated by the hope to create a type of hypercomplex numbers related to the three dimensional space like complex numbers are related to the plane.

Nowadays, quaternions are used as a tool for modelling problems in almost all applied sciences: computer vision, robotics, navigation, virtual reality, etc.

In this thesis we present the fundamental concepts and properties of quaternions and describe how quaternions can provide a unique and powerful tool for characterizing 3D rotations.

The computational contributions of this work are:

1. QuatCalc: a MATLAB calculator for quaternions, which was design to perform quaternion arithmetic.
2. QuatRot: a MATLAB gui for visualizing 3D rotations with quaternions.
3. A collection of MATLAB routines for 3D general rotations.

Conteúdo

1	Introdução	1
2	Os quaterniões	5
2.1	Resenha histórica	5
2.2	Definições e resultados básicos	10
2.3	Quaterniões e vectores	16
2.4	Representação matricial	19
2.5	Representação trigonométrica	24
2.6	Funções de variável quaterniônica	25
3	Quaterniões e Rotações	33
3.1	Representação de Rotações em \mathbb{R}^2	33
3.2	Representação de Rotações em \mathbb{R}^3	38
	Matrizes de rotação	38
	Ângulos de Euler	46
	Eixo-Ângulo	52
3.3	Rotações com quaterniões	55
3.4	Análise comparativa das representações	63
	Conversão entre representações	64
	Esforço computacional na rotação de vectores	71
3.5	Considerações adicionais	75

4	Cálculo numérico e simbólico com quaterniões	77
4.1	Software para Quaterniões	77
4.2	O MATLAB	81
4.3	A classe @quaternion	82
4.4	A calculadora QuatCalc	85
4.5	Rotinas para rotações 3D	89
4.6	QuatRotation - Visualização de rotações 3D	91

Capítulo 1

Introdução

“Quaternions came from Hamilton after his really good work had been done; and though beautifully ingenious, have been an unmixed evil to those who have touched them in any way.”
- Lord Kelvin

Os quaterniões descobertos por Sir Hamilton em 1843 são hoje um tema muito actual e a sua vasta aplicação abrange vários ramos das ciências: o tratamento de sinal, o tratamento de imagens, a mecânica quântica, a aeronáutica, a animação computacional, são apenas algumas das áreas que reconhecem a ferramenta importante que os quaterniões são, tanto para a modelação de um problema, como para a simplificação dos cálculos algébricos associados a esse problema.

A presente dissertação pretende ser um estudo das características dos quaterniões e do seu papel na implementação de rotações de objectos tridimensionais. Dois factos recentes proporcionaram o reaparecimento dos quaterniões nesta área: por um lado, o efeito do Gimbal Lock nos processos de rotação através do método tradicional dos ângulos de Euler, nomeadamente no módulo lunar Apollo, alarmou a comunidade aeronáutica e astronáutica, que procurou soluções para o problema, e por outro lado, o surgimento da era da computação gráfica e da animação computacional. Ken Shoemake apresentou em 1985, na conferência Siggraph, um artigo que despertou a comunidade computacional, ao expor as

deficiências dos ângulos de Euler e ao introduzir os quaterniões como alternativa e solução, estimulando uma vasta variedade de investigações e aplicações por diferentes comunidades científicas.

A estrutura desta tese assenta na fundamentação teórica dos quaterniões e das rotações, e de respectivas aplicações computacionais implementadas no MATLAB e está organizada do seguinte modo:

No Capítulo 2, após uma breve resenha histórica, são referidas diversas aplicações actuais dos quaterniões. São, ainda, introduzidos os conceitos básicos, definidas as operações elementares e apresentadas algumas formas de representação de quaterniões. Alguns resultados importantes e *versões quaterniónicas* de funções complexas elementares, constam também do conteúdo deste capítulo.

No terceiro capítulo descrevemos as características dos métodos mais utilizados na representação de rotações: matrizes de rotação, ângulos de Euler, eixo-ângulo e, finalmente, quaterniões. Apresentamos um balanço das vantagens e desvantagens de cada uma das representações, dando um ênfase especial aos quaterniões, e fazemos ainda uma análise do esforço computacional.

Face à necessidade de tirar partido das vantagens de cada representação e das implementações de algumas APIs (*Application Programming Interface*) gráficas existentes no mercado, as conversões entre processos de rotação são inevitáveis. Nesta sequência, apresentam-se as conversões entre os quaterniões e os métodos de representação mais frequentes, avaliando o custo em termos de número de operações envolvidas.

O último capítulo é totalmente dedicado à descrição do *software* disponível ou desenvolvido no âmbito deste projecto, para *operar* com quaterniões.

A atenção de várias comunidades científicas pela utilização de quaterniões, originou, naturalmente, o aparecimento de códigos nas mais diversas linguagens de programação

para manipulação de quatérnios. Assim, neste capítulo, fazemos referência ao *software* existente (livre, ou não) que permite a utilização de quatérnios, directa ou indirectamente, nas mais diversas áreas.

O desenvolvimento de *software* para tratamento numérico e/ou simbólico de quatérnios e, como aplicação importante, para descrição de rotações, é um dos objectivos principais do presente trabalho, cujos contributos computacionais são apresentados também neste capítulo. Mais precisamente, foi desenhada uma calculadora - a QuatCalc, em MATLAB, para tratamento numérico e simbólico de quatérnios e implementada uma colecção de rotinas que reflectem os temas abordados nesta tese. Em particular, inclui-se uma ferramenta - a QuatRotation - que facilita a visualização de rotações em três dimensões tornando-as mais intuitivas.

Todos os programas foram testados num computador pessoal com um processador Pentium 4 a 1.7 Ghz e 512 Mb de memória RAM.

A calculadora QuatCalc, na versão executável, e todas as funções em MATLAB desenvolvidas no âmbito deste trabalho são do domínio público e podem ser encontradas em <http://www.math.uminho.pt/~mmc/dissertacoes/quaternions.htm>.

Capítulo 2

Os quaterniões

Neste capítulo é feita uma breve introdução aos quaterniões. Em particular, são definidas as principais operações e correspondentes propriedades, sendo também apresentadas as diferentes formas de representação de quaterniões. O capítulo termina com a exposição de funções de variável quaterniônica: funções exponencial e logarítmica, funções trigonométricas e hiperbólicas. As operações/funções citadas foram implementadas numa calculadora de quaterniões que será descrita com pormenor no último capítulo.

2.1 Resenha histórica

Os quaterniões foram inventados, em 1843, pelo matemático irlandês Sir William Rowan Hamilton (*1805 - †1865), a partir dos trabalhos dos matemáticos Carl Friedrich Gauss e Leonhard Euler. Sir Hamilton estudava desde 1830 a interpretação geométrica da aritmética dos números complexos no plano e procurava obter resultados análogos no espaço a três dimensões. Em 1833, obteve como resultado que os números complexos formam uma álgebra de pares ordenados de números reais. Sir William Rowan Hamilton tentou estender este conceito a triplos de números, com um real e dois imaginários. Por mais de um década, esta questão preocupou Hamilton que escreveu ao filho [Ham67]:

"Every morning in the early part of the above-cited month [Oct. 1843] on my coming down to breakfast, your brother William Edwin and yourself used to ask me: Well, Papa,

can you multiply triplets? Whereto I was always obliged to reply, with a sad shake of the head: No, I can only add and subtract them".

Uma das motivações de Hamilton para procurar números complexos tridimensionais, era encontrar uma descrição de rotações no espaço, análoga ao caso complexo, onde a multiplicação corresponde a uma rotação e a uma mudança de escala.

No dia 16 do Outubro de 1843, enquanto passeava no *Royal Canal* em Dublin, Hamilton apercebeu-se que seriam necessários quatro números para descrever uma rotação seguida de uma mudança de escala: um número correspondente à mudança de escala, outro número para indicar o ângulo de rotação e os dois restantes para indicar o plano de rotação, encontrando assim a solução para o problema. Introduzindo uma estrutura não comutativa, Hamilton encontrou um fecho para a multiplicação de números da forma $w + ix + jy + kz$, fazendo

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

e chamou aos seus *números complexos* em \mathbb{R}^4 , **quaterniões**. Entusiasmado com a descoberta, Hamilton gravou, nessa altura, a fórmula fundamental da álgebra dos quaterniões, numa pedra da ponte de *Brougham* (hoje com o nome de *Broom Bridge*). Nenhum sinal desta gravação pode ser encontrado hoje, mas foi erguida uma placa no local, em 1956, comemorando a descoberta e exibindo a fórmula.

Na *Royal Irish Academy*, Hamilton apresentou uma teoria detalhada de um sistema algébrico não comutativo e publicou um conjunto de resultados, que deram origem, em 1853, ao livro: "*Lectures on Quaternions: containing a systematic statement of a new mathematical method*"[Ham53]. Este é, historicamente, o primeiro exemplo de uma álgebra não comutativa que nasceu subitamente e abriu as portas da álgebra abstracta.

Hamilton dedicou o resto da sua vida a desenvolver aplicações dos quaterniões à geometria, mecânica e física. Nesse período introduziu termos como *vector*, *versor*, *tensor*, *escalar*, que são familiares nos nossos dias. Como resultado deste trabalho foi ainda editado em 1866, a título póstumo, pelo seu filho William Edwin Hamilton, um trabalho em dois

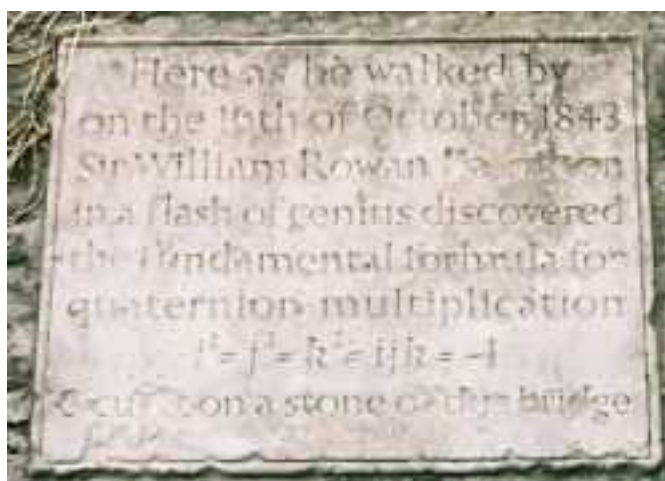


Figura 2.1: Placa comemorativa da descoberta dos quaterniões na ponte de Brougham

volumes: “*Elements of Quaternions*” [Ham69].

A perda da propriedade comutativa da multiplicação para sistemas numéricos foi de particular importância para as sucessivas investigações. Em 1843, Graves descobriu uma álgebra não associativa com oito elementos de base, os octoniões, e publicou o seu trabalho em 1848 [Gra48]. Os octoniões foram descobertos também, em 1845, pelo matemático inglês Arthur Cayley (*1821 - †1895) [Cay45], sendo também conhecidos como números de Cayley.

O próprio Hamilton introduziu em 1853, nas suas *Lectures on Quaternions*, quaterniões com coeficientes complexos, os *Biquaterniões*. Nesse mesmo texto, desenvolveu, ainda, uma nova generalização que já tinha iniciado num artigo nos *Transactions of Royal Irish Academy*, em 1848: os *Números Hipercomplexos*.

A aplicação dos quaterniões foi explorada de imediato. Em 1845, Cayley mostrou que os quaterniões podem representar orientações e em 1858 demonstrou que os quaterniões podem ser representados por meio de matrizes. Também o físico e matemático escocês, James Clerk Maxwell (*1831 - †1879) percebeu a utilidade dos quaterniões para o seu trabalho, utilizando-os na sua formulação de ondas electromagnéticas, em 1864, que mais

tarde conduziu ao primeiro telégrafo, desenvolvido em 1895 por Marconi, e que originou posteriormente a rádio e a televisão. Maxwell introduziu-os ainda no seu *Tratado de Electricidade e Magnetismo*, em 1885. A aplicação dos quaterniões em campos como a mecânica clássica e a teoria da relatividade foi identificada no início do século XX.

Durante algumas dezenas de anos os quaterniões foram largamente divulgados mas por alguma razão misteriosa, caíram no esquecimento sendo redescobertos ainda há pouco tempo pelo mundo da Engenharia Espacial e da Robótica. A capacidade dos quaterniões em representar rotações a três dimensões à volta de um eixo arbitrário motivou investigadores a empregar a álgebra dos quaterniões nas equações que definem a cinemática das rotações. Como resultado, emergiram novas áreas de aplicação que envolvem quaterniões, incluindo vários campos como a robótica, a mecânica orbital, as tecnologias aeroespaciais (e.g. trajectórias dos satélites) e sistemas de navegação inercial [Kui02].

Em 1985, Ken Shoemake introduziu os quaterniões na Computação Gráfica, com o objectivo de facilitar a animação rotacional e apresentou a sua pesquisa na conferência SIGGRAPH (*Special interest group on computer graphics*), com o artigo “*Animating Rotation with Quaternion Curves*” [Sho85], considerado desde a sua publicação, como uma referência. Shoemake apresentou também as vantagens da utilização dos quaterniões na obtenção de interpolações suaves de rotações e descreveu algoritmos de conversões entre matrizes, ângulos de Euler e quaterniões. Este trabalho deu origem a um interesse crescente pela aplicação dos quaterniões no campo da computação gráfica e na programação de jogos e desde aí a interpolação de orientações tem sido objecto de estudo de diversos trabalhos publicados nesta área.

Os programadores gráficos tomaram consciência do potencial dos quaterniões como operador de rotação poderoso. As recentes APIs gráficas disponibilizam funções para operar com quaterniões. Por exemplo, o *API Java-3D* tem classes para criar objectos quaterniônicos (`javax.vecmath.Quat4d`), o *Povray (Persistence of Vision Ray Tracer)* suporta quaterniões, o *Quickdraw-3D* fornece rotinas para operações com quaterniões, o MATLAB

disponibiliza uma ferramenta para simulação de sistemas aeroespaciais que opera com quaterniões, o *Aerospace Blockset*, e o *Mathematica* tem um package *Quaternions*. Os quaterniões são ainda utilizados em algoritmos de animação computacional, na programação de jogos, na interpolação de *keyframes* e na simulação do movimento de uma câmara no espaço tridimensional.

A união da *Walt Disney Pictures* com os estúdios de animação *Pixar* resultou numa nova era para a animação computacional. Na última década assistiu-se a uma nova forma de criação de filmes de animação, que deu origem a sucessivos sucessos cinematográficos: *Toy Story*(1995); *A Bug's Life*(1998); *Toy Story 2*(1999); *Monsters Inc.*(2001); *Finding Nemo*(2003) e *The Incredibles*(2004) [TA05]. Na implementação computacional destes filmes, as orientações são expressas por quaterniões. Nos jogos de computadores, cita-se por exemplo, a conhecida personagem *Lara Croft*, do jogo *Tomb Raider*, animada com quaterniões. Alguns periféricos de jogos de realidade virtual empregam quaterniões para parametrizar rotações, e.g., o *Logitech-3D mouse* tem um modo de saída para quaterniões. Assim, na computação gráfica, o domínio de aplicação dos quaterniões expandiu-se rapidamente em campos como a visualização, os fractais e a realidade virtual.

A computação digital tornou-se uma componente essencial da existência humana e influencia tudo, desde o entretenimento, às funcionalidades do carro familiar, aos métodos da investigação científica. Os quaterniões são ainda utilizados, como ferramenta matemática e algorítmica, em sistemas de simulação e assistência por computador de operações cirúrgicas [Bai96]. As aplicações aeroespaciais e os simuladores de voos também empregam quaterniões. Astronautas da NASA, bem como directores cinematográficos de Hollywood, usam os sistemas computacionais para dirigir e controlar as posições de objectos no espaço.

Como curiosidade, refira-se ainda que Andrew J. Hanson apresenta no seu livro [Kui02] dois exemplos simples - o *belt trick* e o *rolling ball* - que ilustram como as orientações de objectos de três dimensões estão presentes nas nossas vidas. O primeiro exemplo apresenta um “truque” realizado apenas com um cinto e cuja explicação prova, de certa forma, que os quaterniões são reais e não apenas uma abstracção. O segundo exemplo expõe um resultado interessante de uma sequência de rotações realizadas apenas com uma bola cuja

explicação emerge das propriedades dos quatérniões.

Face ao grande contributo dado por Hamilton, a Irlanda tem ao longo do tempo, prestado homenagem ao seu ilustre matemático e cientista, que passou grande parte da sua vida em Dublin. Em 1943, aquando do centenário da descoberta dos quatérniões, foi emitido um selo comemorativo com a fórmula fundamental dos quatérniões. Em 2005, celebrou-se o bicentenário do nascimento de William Rowan Hamilton e o governo Irlandês dedicou-lhe o ano, proclamando-o como: *“Hamilton Year 2005: Celebrating Irish Science and Technology”*. Neste enquadramento, dois símbolos emblemáticos foram produzidos: um selo comemorativo e uma moeda de colecção que representa a linguagem simbólica desenvolvida por Hamilton.

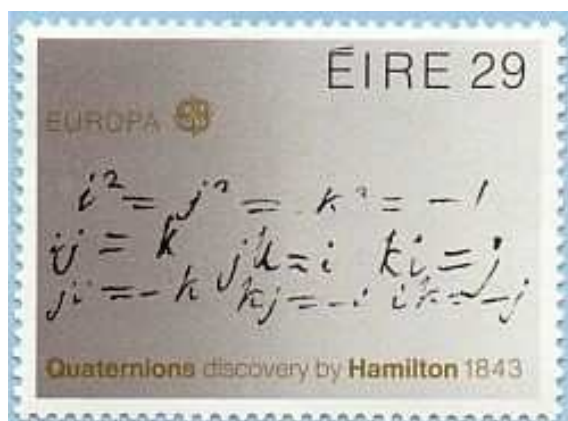


Figura 2.2: Selo comemorativo do centenário da descoberta dos quatérniões

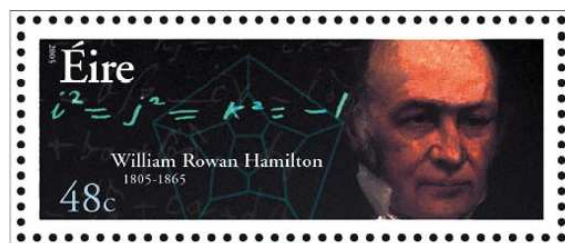


Figura 2.3: Selo comemorativo do bicentenário do nascimento de Hamilton

2.2 Definições e resultados básicos

O conjunto dos quatérniões é actualmente designado por \mathbb{H} , em homenagem a Hamilton. Como já foi referido anteriormente, este sistema define-se a partir de três números imaginários diferentes, i , j e k , em que:

$$i^2 = j^2 = k^2 = -1 \quad (2.1)$$

e

$$ij = -ji = k; \quad (2.2)$$

$$jk = -kj = i; \quad (2.3)$$

$$ki = -ik = j. \quad (2.4)$$

Usaremos ao longo deste trabalho a notação cartesiana

$$q = q_0 + \mathbf{q} = q_0 + q_1i + q_2j + q_3k,$$

com $q_0, q_1, q_2, q_3 \in \mathbb{R}$, ou a notação vectorial

$$q = (q_0, q_1, q_2, q_3),$$

identificando assim \mathbb{H} com \mathbb{R}^4 . À semelhança dos números complexos,

- q_0 representa a parte real (ou escalar, na terminologia de Hamilton) de q e denota-se por $\mathbf{Re}(q)$ (ou $\mathbf{Sc}(q)$).
- $\mathbf{q} := q_1i + q_2j + q_3k$ corresponde à parte imaginária (ou parte vectorial) do quaterniões, denotando-se também por $\mathbf{Im}(q)$ (ou $\mathbf{Vec}(q)$).
- q diz-se um **quaterniões puro** se $\mathbf{Re}(q) = 0$, ou, de forma equivalente, se $q = \mathbf{q}$.

Apresentamos de seguida, mais formalmente, as principais operações entre quaterniões da forma, $p = p_0 + \mathbf{p} = p_0 + p_1i + p_2j + p_3k$ e $q = q_0 + \mathbf{q} = q_0 + q_1i + q_2j + q_3k$.

Definição 2.1. Igualdade de quaterniões

$$p = q \quad \text{sse} \quad p_0 = q_0 \quad \text{e} \quad \mathbf{p} = \mathbf{q}.$$

Definição 2.2. *Adição e subtração de quaterniões*

$$\begin{aligned} p \pm q &= (p_0 + \mathbf{p}) \pm (q_0 + \mathbf{q}) \\ &= (p_0 + q_0) \pm (\mathbf{p} + \mathbf{q}). \end{aligned} \quad (2.5)$$

Definição 2.3. *Multiplicação de quaterniões*

$$\begin{aligned} p * q &= (p_0 + p_1i + p_2j + p_3k)(q_0 + q_1i + q_2j + q_3k) \\ &= (p_0q_0 - p_1q_1 - p_2q_2 - p_3q_3) + \\ &\quad (p_0q_1 + p_1q_0 + p_2q_3 - p_3q_2)i + \\ &\quad (p_0q_2 - p_1q_3 + p_2q_0 + p_3q_1)j + \\ &\quad (p_0q_3 + p_1q_2 - p_2q_1 + p_3q_0)k. \end{aligned} \quad (2.6)$$

A multiplicação por um escalar é facilmente introduzida, identificando $\alpha \in \mathbb{R}$ com o quaterniões $\alpha = \alpha_0 + \mathbf{0}$.

Definição 2.4. *Multiplicação de um escalar por um quaterniões*

$$\begin{aligned} \alpha * q &= (\alpha + \mathbf{0}) * (q_0 + \mathbf{q}), \quad \alpha \in \mathbb{R} \\ &= \alpha q_0 + \alpha q_1i + \alpha q_2j + \alpha q_3k. \end{aligned}$$

Sempre que não haja dúvidas quanto à operação de multiplicação envolvida, omitiremos, por simplificação de escrita, o símbolo $*$.

De seguida, são apresentados alguns resultados que se podem obter, de imediato, usando as definições introduzidas anteriormente.

Proposição 2.1. *Sejam $p, q, q' \in \mathbb{H}$ e $\alpha, \beta \in \mathbb{R}$. O produto de um escalar por um quaterniões goza das seguintes propriedades:*

1. $\alpha q = q\alpha$;
2. $p(\alpha q + \beta q') = \alpha pq + \beta pq'$;
3. $(\alpha q + \beta q')p = \alpha qp + \beta q'p$.

Conseqüentemente, o conjunto \mathbb{H} dos quaterniões, munido da operação de adição e produto por um escalar, constitui um *espaço vectorial real*.

Proposição 2.2. *Sejam $p, q, q' \in \mathbb{H}$. O produto de quaterniões goza das seguintes propriedades:*

1. $pq \neq qp$, em geral;
2. $(pq)q' = p(qq')$;
3. $p(q + q') = pq + pq'$;
4. $(q + q')p = qp + q'p$.

Conseqüentemente, \mathbb{H} é uma álgebra não comutativa, a *álgebra dos quaterniões*. A associatividade do produto de quaterniões torna a álgebra dos quaterniões numa álgebra associativa.

Ainda em relação à comutatividade, a proposição seguinte esclarece em que situações um dado quaternião comuta com qualquer outro.

Proposição 2.3. *Um quaternião p é real sse, para qualquer outro quaternião q , verificar $pq = qp$.*

Demonstração. É suficiente demonstrar que a partir da relação comutativa com um quaternião q , se obtém que p tem de ser um número real. Suponhamos, então, por hipótese, que $pq = qp$

Se $p = p_0 + p_1i + p_2j + p_3k$ e $q = i$, tem-se:

$$pq = pi = p_0i - p_1 + p_3j - p_2k,$$

$$qp = ip = p_0i - p_1 - p_3j + p_2k.$$

Assim, como $pi = ip$ tem-se que $2(p_2k - p_3j) = 0$, pelo que $p_2 = p_3 = 0$.

Após repetir este procedimento com o quaternião $q = j$, verifica-se que p_1 também tem que ser nulo. □

Definição 2.5. Conjugado de um quaterniões

O conjugado de $q = q_0 + \mathbf{q}$, denota-se por \bar{q} e define-se como:

$$\bar{q} = q_0 - \mathbf{q} = q_0 - q_1i - q_2j - q_3k. \quad (2.7)$$

Esta definição dá origem às seguintes propriedades, cujas demonstrações são imediatas:

Proposição 2.4. *Sejam $p, q \in \mathbb{H}$ e $\alpha \in \mathbb{R}$. Então,*

1. $\overline{\bar{q}} = q$;
2. $\overline{pq} = \bar{q}\bar{p}$;
3. $\overline{p+q} = \bar{p} + \bar{q}$;
4. $\overline{\alpha p} = \alpha\bar{p}$;
5. $p\bar{p} = \bar{p}p$.

Observe-se que a parte real de um quaterniões q , $\text{Re}(q)$, pode escrever-se como:

$$\text{Re}(q) = \frac{(q + \bar{q})}{2}.$$

Definição 2.6. Norma de um quaterniões

Seja $q \in \mathbb{H}$. Define-se norma (valor absoluto ou módulo) de q , como sendo o número não negativo,

$$\|q\| = \|q_0 + q_1i + q_2j + q_3k\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}. \quad (2.8)$$

Um pequeno cálculo revela que:

$$q\bar{q} = \bar{q}q = qq = \|q\|^2,$$

e, portanto,

$$\|q\| = \sqrt{q\bar{q}}. \quad (2.9)$$

Proposição 2.5. *Sejam $p, q \in \mathbb{H}$. Então,*

1. $\|p\| = \|\bar{p}\|$;

$$2. \|pq\| = \|p\| \|q\|.$$

Definição 2.7. Quaterniões unitários

Um quaterniões $q \in \mathbb{H}$ diz-se unitário se $\|q\| = 1$.

Ao longo deste texto, iremos usar, por simplificação de escrita,

- \mathbb{H}_1 para denotar o conjunto dos quaterniões unitários;
- $\overset{\circ}{\mathbb{H}}$ como o conjunto dos quaterniões não nulos;
- $\mathbb{H}_{\mathbb{R}^3}$ para representar o subconjunto dos quaterniões puros.

Proposição 2.6. Inverso de um quaterniões não nulo

Seja $q \in \overset{\circ}{\mathbb{H}}$. Então existe $q^{-1} \in \overset{\circ}{\mathbb{H}}$ tal que $qq^{-1} = q^{-1}q = 1$. Além disso, q^{-1} é único e é dado por:

$$q^{-1} = \frac{\bar{q}}{\|q\|^2}, \quad (2.10)$$

em que a divisão de um quaterniões por um escalar real corresponde à divisão por componente.

Proposição 2.7. Sejam $p, q \in \overset{\circ}{\mathbb{H}}$ e $p_1, q_1 \in \mathbb{H}_1$. Então, são válidas as seguintes propriedades:

1. $(p^{-1})^{-1} = p$;
2. $(pq)^{-1} = q^{-1}p^{-1}$;
3. $\|q\| = \|q\|^{-1}$;
4. $\|p_1q_1\| = 1$;
5. $p_1^{-1} = \bar{p}_1$.

Dada a não comutatividade do produto de quaterniões, a divisão de quaterniões define-se de dois modos. Usaremos o símbolo \backslash para denotar a divisão à esquerda e o símbolo $/$ para a divisão à direita. Assim, tem-se,

Definição 2.8. Divisão de quaterniões

Sejam $p \in \mathbb{H}$ e $q \in \overset{\circ}{\mathbb{H}}$. A divisão de p por q define-se como:

- *divisão à esquerda:* $p \backslash q = q^{-1}p$,
- *divisão à direita:* $p/q = pq^{-1}$.

Podemos facilmente concluir que o conjunto dos quaterniões constitui um anel não comutativo, com divisão. Segue-se que $(\mathbb{H}, +, \cdot)$ é um corpo não comutativo. Finalmente, podemos também provar que o conjunto \mathbb{H}_1 dos quaterniões unitários é um subgrupo de $\overset{\circ}{\mathbb{H}}$.

2.3 Quaterniões e vectores

Iremos agora evidenciar a relação entre a estrutura dos quaterniões e a álgebra clássica dos vectores em \mathbb{R}^4 . Relembremos que um quaternião $q = q_0 + iq_1 + jq_2 + kq_3 \in \mathbb{H}$ pode ser identificado com o vector de \mathbb{R}^4 , (q_0, q_1, q_2, q_3) .

Consideremos a base canónica do espaço vectorial euclidiano \mathbb{R}^4 :

$$e_0 = (1, 0, 0, 0), \quad e_1 = (0, 1, 0, 0), \quad e_2 = (0, 0, 1, 0), \quad e_3 = (0, 0, 0, 1).$$

Esquemáticamente, a multiplicação entre estes elementos pode definir-se, por analogia com a multiplicação entre os elementos da base $\{1, i, j, k\}$ de \mathbb{H} . Assim, tem-se:

	e_0	e_1	e_2	e_3
e_0	e_0	e_1	e_2	e_3
e_1	e_1	$-e_0$	e_3	$-e_2$
e_2	e_3	$-e_3$	$-e_0$	e_1
e_3	e_3	e_2	$-e_1$	$-e_0$

Tabela 2.1: Tabela de Cayley

Consideremos $p = p_0 + \mathbf{p} = p_0 + p_1i + p_2j + p_3k$ e $q = q_0 + \mathbf{q} = q_0 + q_1i + q_2j + q_3k \in \mathbb{H}$. Notando que os quaterniões puros \mathbf{p} e \mathbf{q} são vectores de \mathbb{R}^3 , a multiplicação de p e q pode ser representada usando o produto interno e o produto externo usuais de \mathbb{R}^3 :

Proposição 2.8. *Reescrita do produto de quaterniões*

$$pq = (p_0q_0 - \mathbf{p} \cdot \mathbf{q}) + p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q}. \quad (2.11)$$

Demonstração. De

$$\begin{aligned} pq &= (p_0 + \mathbf{p})(q_0 + \mathbf{q}) \\ &= p_0q_0 - (p_1q_1 + p_2q_2 + p_3q_3) + (p_0q_1 + q_0p_1 + p_2q_3 - p_3q_2)i + \\ &\quad (p_0q_2 + q_0p_2 + p_3q_1 - p_1q_3)j + (p_0q_3 + q_0p_3 + p_1q_2 - p_2q_1)k, \end{aligned}$$

obtém-se

$$\begin{aligned} pq &= (p_0q_0 - \mathbf{p} \cdot \mathbf{q}) \\ &\quad + p_0(q_1i + q_2j + q_3k) + q_0(p_1i + p_2j + p_3k) \\ &\quad + (p_2q_3 - p_3q_2)i + (p_3q_1 - p_1q_3)j + (p_1q_2 - p_2q_1)k \\ &= (p_0q_0 - \mathbf{p} \cdot \mathbf{q}) + p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q}. \end{aligned}$$

□

O resultado (2.11) deve-se a Rodrigues [Rod40], embora num contexto e com uma notação muito diferentes.

Com esta representação torna-se óbvio que a multiplicação de quaterniões não é comutativa, uma vez que o produto externo de vectores não é comutativo. Deste modo,

$$pq = qp \text{ sse } \mathbf{p} \times \mathbf{q} = 0.$$

Além disso, observe-se ainda que,

$$q^2 = (q_0^2 - \mathbf{q} \cdot \mathbf{q}) + 2q_0\mathbf{q},$$

uma vez que o produto externo de um vector pelo próprio é nulo. De (2.11) resulta ainda

Proposição 2.9. *Se $p, q \in \mathbb{H}_{\mathbb{R}^3}$, i.e. se $p = \mathbf{p}$ e $q = \mathbf{q}$ são dois quaterniões puros, então*

$$(a) \quad \mathbf{p} \times \mathbf{q} = \text{Vec}(\mathbf{pq}) = \frac{1}{2}(\mathbf{pq} - \mathbf{qp});$$

$$(b) \quad \mathbf{p} \cdot \mathbf{q} = \text{Re}(\overline{\mathbf{p}}\mathbf{q}) = \frac{1}{2}(\overline{\mathbf{p}}\mathbf{q} + \mathbf{q}\overline{\mathbf{p}}).$$

Demonstração. O primeiro resultado obtém-se, notando que

$$\text{Vec}(\mathbf{pq}) = \text{Vec}(-\mathbf{p} \cdot \mathbf{q} + \mathbf{p} \times \mathbf{q}) = \mathbf{p} \times \mathbf{q}$$

e

$$\begin{aligned} \frac{1}{2}(\mathbf{pq} - \mathbf{qp}) &= \frac{1}{2}(-\mathbf{p} \cdot \mathbf{q} + \mathbf{p} \times \mathbf{q} + \mathbf{q} \cdot \mathbf{p} - \mathbf{q} \times \mathbf{p}) \\ &= \frac{1}{2}(\mathbf{p} \times \mathbf{q} - \mathbf{q} \times \mathbf{p}) = \mathbf{p} \times \mathbf{q}. \end{aligned}$$

Além disso,

$$\text{Re}(\overline{\mathbf{p}}\mathbf{q}) = \text{Re}(-\overline{\mathbf{p}} \cdot \mathbf{q} + \overline{\mathbf{p}} \times \mathbf{q}) = -\overline{\mathbf{p}} \cdot \mathbf{q} = -(-p_1q_1 - p_2q_2 - p_3q_3) = \mathbf{p} \cdot \mathbf{q}.$$

Por outro lado,

$$\begin{aligned} \frac{1}{2}(\overline{\mathbf{p}}\mathbf{q} + \mathbf{q}\overline{\mathbf{p}}) &= \frac{1}{2}(-\overline{\mathbf{p}} \cdot \mathbf{q} + \overline{\mathbf{p}} \times \mathbf{q} - \mathbf{q} \cdot \overline{\mathbf{p}} + \mathbf{q} \times \overline{\mathbf{p}}) \\ &= \frac{1}{2}(-2\overline{\mathbf{p}} \cdot \mathbf{q}) = -\overline{\mathbf{p}} \cdot \mathbf{q} = \mathbf{p} \cdot \mathbf{q}, \end{aligned}$$

obtendo-se então (b). □

Nota 2.1. Outros produtos podem ainda ser definidos, cujas propriedades podem ser úteis em determinadas aplicações práticas. A título de exemplo referimos, para $p = p_0 + p_1i + p_2j + p_3k$ e $q = q_0 + q_1i + q_2j + q_3k$,

- Produto simétrico:

$$p \star q = \frac{pq + qp}{2}.$$

Este produto é comutativo, mas em geral não é associativo.

- Produto externo Euclidiano:

$$p \circ q = p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q} = \mathbf{Vec}(pq).$$

- Produto externo de Grassmann:

$$p \times q = \mathbf{p} \times \mathbf{q} = (p_2q_3 - p_3q_2)i + (p_3q_1 - p_1q_3)j + (p_1q_2 - p_2q_1)k.$$

2.4 Representação matricial

A representação de quaterniões como matrizes, em que a adição e a multiplicação de quaterniões correspondam à adição e multiplicação de matrizes (i.e., homomorfismos matrizes-quaterniões), pode ser realizada de duas formas: utilizando matrizes complexas 2×2 ou recorrendo a matrizes reais 4×4 .

Representação com matrizes complexas 2×2

Começemos por relembrar que os números complexos são isomorfos às matrizes reais da forma

$$\begin{bmatrix} a & b \\ -b & a \end{bmatrix}, \quad a, b \in \mathbb{R} \quad (2.12)$$

Qualquer número complexo $z = a + bi$ pode ser escrito como

$$z = a \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + b \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = aE + bI, \quad (2.13)$$

onde E é a matriz identidade e $I^2 = -E$.

A adição e a multiplicação de números complexos correspondem à adição e multiplicação das matrizes associadas.

Exemplo 2.1. Consideremos os complexos $-1 + 2i$ e $4 - i$. O produto destes números complexos pode ser expresso matricialmente como:

$$\begin{bmatrix} -1 & 2 \\ -2 & -1 \end{bmatrix} \begin{bmatrix} 4 & -1 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} -2 & 9 \\ -9 & -2 \end{bmatrix},$$

O resultado do produto corresponde ao número complexo $-2 + 9i$.

Consideremos agora um quaterniões $q = q_0 + q_1i + q_2j + q_3k$. Como $ij = k$, é possível escrever q na forma

$$q = z + wj, \quad (2.14)$$

onde $z = q_0 + q_1i$ e $w = q_2 + q_3i$. Esta representação é designada por *representação complexa de um quaterniões*.

O produto de $p = p_0 + p_1i + p_2j + p_3k = (p_0 + p_1i) + (p_2 + p_3i)j = z_1 + w_1j$ por $q = q_0 + q_1i + q_2j + q_3k = (q_0 + q_1i) + (q_2 + q_3i)j = z_2 + w_2j$ pode reescrever-se, em termos de variáveis complexas, do seguinte modo:

$$\begin{aligned} pq &= (z_1 + w_1j)(z_2 + w_2j) \\ &= z_1z_2 + w_1jw_2j + w_1jz_2 + z_1w_2j \\ &= (z_1z_2 - w_1\bar{w}_2) + (z_1w_2 + w_1\bar{z}_2)j. \end{aligned}$$

Esta representação do produto de quaterniões corresponde à multiplicação das matrizes:

$$\begin{bmatrix} z_1 & w_1 \\ -\bar{w}_1 & \bar{z}_1 \end{bmatrix} \begin{bmatrix} z_2 & w_2 \\ -\bar{w}_2 & \bar{z}_2 \end{bmatrix}.$$

Assim, cada quaterniões $q = q_0 + q_1i + q_2j + q_3k = z + wj$ pode ser associado com a matriz complexa

$$\begin{bmatrix} z & w \\ -\bar{w} & \bar{z} \end{bmatrix} = \begin{bmatrix} q_0 + q_1i & q_2 + q_3i \\ -q_2 + q_3i & q_0 - q_1i \end{bmatrix}, \quad (2.15)$$

obtendo-se assim a *representação matricial complexa* de um quaterniões.

Qualquer quaterniões pode ainda ser escrito como

$$q = q_0 + q_1I + q_2J + q_3K,$$

onde

$$E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad I = \begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix}, \quad J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad K = \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix}.$$

Note-se que, tal como seria de esperar,

$$I^2 = J^2 = K^2 = -E,$$

$$IJ = -JI = K,$$

$$JK = -KJ = I,$$

$$KI = -IK = J.$$

Exemplo 2.2. Os quaterniões $p = -1 + 2i + j - k$ e $q = 2 - i + 3j - k$ podem representar-se na forma complexa, respectivamente, como

$$p = (-1 + 2i) + (1 - i)j, \quad q = (2 - i) + (3 - i)j.$$

Por outro lado, as matrizes complexas correspondentes aos quaterniões p e q são, respectivamente:

$$P = \begin{bmatrix} -1 + 2i & 1 - i \\ -1 - i & -1 - 2i \end{bmatrix} \quad Q = \begin{bmatrix} 2 - i & 3 - i \\ -3 - i & 2 + i \end{bmatrix}.$$

O produto

$$P.Q = \begin{bmatrix} -1 + 2i & 1 - i \\ -1 - i & -1 - 2i \end{bmatrix} \begin{bmatrix} 2 - i & 3 - i \\ -3 - i & 2 + i \end{bmatrix} = \begin{bmatrix} -4 + 7i & 2 + 6i \\ -2 + 6i & -4 - 7i \end{bmatrix},$$

corresponde ao quaternião $-4 + 7i + (2 + 6i)j = -4 + 7i + 2j + 6k$, ou seja, ao resultado de pq .

De (2.14), conclui-se, de imediato que, se Q é a representação matricial de um quaternião q , então:

- se $q_2 = q_3 = 0$, i.e. se q é um número complexo, Q é uma matriz diagonal;
- $\|q\|^2 = \det(Q)$;
- a representação matricial de \bar{q} é \overline{Q}^T ;
- se $q \in \mathbb{H}_1$, i.e., se $\|q\| = 1$, então $\det(Q) = 1$ e $Q\overline{Q}^T = E$, i.e. Q é uma matriz unitária.

Representação por matrizes reais 4×4

Substituindo cada número complexo em (2.15) pela sua correspondente matriz real (2.12), obtém-se a *representação matricial real* de q , que denotamos por $R(q)$.

$$\begin{bmatrix} z & \vdots & w \\ \dots & \dots & \dots \\ -\bar{w} & \vdots & \bar{z} \end{bmatrix} = \begin{bmatrix} q_0 & q_1 & \vdots & q_2 & q_3 \\ -q_1 & q_0 & \vdots & -q_3 & q_2 \\ \dots & \dots & \dots & \dots & \dots \\ -q_2 & q_3 & \vdots & q_0 & -q_1 \\ -q_3 & -q_2 & \vdots & q_1 & q_0 \end{bmatrix} =: R(q). \quad (2.16)$$

A álgebra dos quaterniões reais \mathbb{H} pode ser gerada pelas matrizes reais:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

De (2.16) conclui-se, de imediato, que:

- $R(\bar{q}) = R(q)^T$;
- $\det(R(q)) = \|q\|^4$;
- se $q \in \mathbb{H}_1$, i.e., se $\|q\| = 1$, então $\det(R(q)) = 1$ e $QQ^T = I$, i.e. Q é uma matriz ortogonal.

Facilmente se verifica que o produto de dois quaterniões p e q definido em (2.6) pode ser reescrito como o produto de um vector linha por uma matriz 4×4 . Note-se que, neste caso, há que distinguir entre o produto de p por q à direita - pq - e à esquerda - qp . A matriz

$R(q)$ permite expressar o produto à direita e daí a sua designação (R - *right*). Assim,

$$pq = pR(q) = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \end{bmatrix} \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix}. \quad (2.17)$$

Já o produto à esquerda pode ser obtido usando a matriz abaixo, que se denota por $L(q)$ (L - *left*).

$$qp = pL(q) = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \end{bmatrix} \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix}. \quad (2.18)$$

A matriz $L(q)$ obtém-se da matriz $R(q)$, transpondo a submatriz:

$$\begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & q_0 & q_3 & -q_2 \\ \cdot & -q_3 & q_0 & q_1 \\ \cdot & q_2 & -q_1 & q_0 \end{bmatrix}.$$

Designando por \mathbf{q} , o vector coluna $[q_1 \ q_2 \ q_3]^T$, podemos expressar as matrizes $R(q)$ e $L(q)$ como:

$$R(q) = \begin{bmatrix} q_0 & \mathbf{q}^T \\ -\mathbf{q} & q_0 I + K(\mathbf{q}) \end{bmatrix} \quad \text{e} \quad L(q) = \begin{bmatrix} q_0 & \mathbf{q}^T \\ -\mathbf{q} & q_0 I - K(\mathbf{q}) \end{bmatrix},$$

onde $K(\mathbf{q})$ é a matriz anti-simétrica de ordem 3

$$K(\mathbf{q}) := \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}, \quad (2.19)$$

que expressa o produto externo de \mathbf{q} por um vector arbitrário \mathbf{v} , i.e. $\mathbf{q} \times \mathbf{v} = K(\mathbf{q})\mathbf{v}$.

Facilmente se verifica que as matrizes $R(q)$ e $L(q)$ possuem as seguintes propriedades:

- $L(q)^T L(q) = L(q)L(q)^T = q^T q I$;
- $R(q)^T R(q) = R(q)R(q)^T = q^T q I$;
- $pR(q) = qL(p)$;
- $qR(q)^T = qL(q)^T = q^T q e_1$, onde $e_1 = [1 \ 0 \ 0 \ 0]^T$;
- $L(q)L(p) = L(L(q)p)$;
- $R(q)R(p) = R(R(q)p)$;
- $L(q)R(p)^T = R(p)^T L(q)$.

Finalmente, note-se que a matriz associada a um quaternião puro é a matriz anti-simétrica,

$$L(q) = \begin{bmatrix} 0 & q_1 & q_2 & q_3 \\ -q_1 & 0 & -q_3 & q_2 \\ -q_2 & q_3 & 0 & -q_1 \\ -q_3 & -q_2 & q_1 & 0 \end{bmatrix},$$

com as propriedades:

$$L(q)^T = -L(q) \quad (e) \quad R(q)^T = -R(q).$$

2.5 Representação trigonométrica

À semelhança dos números complexos, é possível escrever um quaternião $q = q_0 + q_1 i + q_2 j + q_3 k = q_0 + \mathbf{q}$ na forma trigonométrica (ou forma polar).

Teorema 2.1. *Se $q \in \overset{\circ}{\mathbb{H}}$, então*

$$q = \|q\| \left(\cos \theta + \frac{\mathbf{q}}{\|\mathbf{q}\|} \operatorname{sen} \theta \right),$$

onde $\theta = \operatorname{arccot} \left(\frac{q_0}{\|\mathbf{q}\|} \right)$.

Demonstração. Seja $\theta = \operatorname{arccot}\left(\frac{q_0}{\|\mathbf{q}\|}\right)$. Usando as igualdades

$$\operatorname{sen}^2 \theta = \frac{1}{1 + \cot^2 \theta} \quad e \quad \cos^2 \theta = \frac{\cot^2 \theta}{1 + \cot^2 \theta},$$

resulta

$$\operatorname{sen}^2 \theta = \frac{1}{1 + \left(\frac{q_0}{\|\mathbf{q}\|}\right)^2} \quad e \quad \cos^2 \theta = \frac{\left(\frac{q_0}{\|\mathbf{q}\|}\right)^2}{1 + \left(\frac{q_0}{\|\mathbf{q}\|}\right)^2}$$

e, conseqüentemente,

$$\begin{aligned} \cos \theta + \frac{\mathbf{q}}{\|\mathbf{q}\|} \operatorname{sen} \theta &= \frac{\left(\frac{q_0}{\|\mathbf{q}\|}\right)}{\sqrt{1 + \left(\frac{q_0}{\|\mathbf{q}\|}\right)^2}} + \frac{\mathbf{q}}{\|\mathbf{q}\|} \frac{1}{\sqrt{1 + \left(\frac{q_0}{\|\mathbf{q}\|}\right)^2}} = \frac{q_0 + \mathbf{q}}{\|\mathbf{q}\| \sqrt{1 + \left(\frac{q_0}{\|\mathbf{q}\|}\right)^2}} \\ &= \frac{q}{\sqrt{q_0^2 + \|\mathbf{q}\|^2}} = \frac{q}{\|q\|}. \end{aligned}$$

□

Nota 2.2. Se $q \in \mathbb{H}_1$, i.e., se $\|q\| = 1$, então

$$q = \cos \theta + \frac{\mathbf{q}}{\|\mathbf{q}\|} \operatorname{sen} \theta.$$

Exemplo 2.3. Se $q = \sqrt{3} + 2i - 2j - k$, então $\|q\| = 2\sqrt{3}$, $\|\mathbf{q}\| = 3$ e $\theta = \operatorname{arccot}\left(\frac{\sqrt{3}}{3}\right) = \frac{\pi}{3}$.

A representação trigonométrica de q é:

$$q = 2\sqrt{3} \left(\cos \frac{\pi}{3} + \frac{2i - 2j - k}{3} \operatorname{sen} \frac{\pi}{3} \right).$$

2.6 Funções de variável quaterniônica

A teoria das funções quaterniônicas de variável quaterniônica é um campo de reconhecido interesse e tem recebido ao longo dos últimos anos significativa atenção, (ver, por exemplo, [Fue28], [BDS82], [Sud79]). Não está no âmbito deste trabalho o estudo, em detalhe, deste tipo de funções. No último capítulo iremos precisar de *versões quaterniônicas* de algumas funções elementares complexas. A generalização destas funções pode ser

feita de várias formas. Apresentamos aqui as mais frequentemente usadas em aplicações práticas, nomeadamente na animação gráfica. Uma lista mais exaustiva de propriedades e correspondentes demonstrações relativas às funções que vamos introduzir podem ser vistas em [GS97].

Consideremos então um quaternião q na forma cartesiana

$$q = q_0 + q_1i + q_2j + q_3k = q_0 + \mathbf{q}.$$

Definição 2.9. Função exponencial

A função exponencial e^q define-se pela série de potências:

$$e^q = \sum_{k=0}^{\infty} \frac{q^k}{k!}.$$

Note-se que, se $q \in \mathbb{C}$ ou $q \in \mathbb{R}$, obtemos, respectivamente, a função exponencial complexa ou real.

Definição 2.10. A função

$$\omega(q) = \begin{cases} \frac{q}{\|q\|}, & \text{se } q \neq 0, \\ 0, & \text{se } q = 0. \end{cases} \quad (2.20)$$

designa-se habitualmente por **senal** de q e retorna um quaternião unitário com a mesma direcção de q . A notação $\text{sgn}(q)$ é também muito frequente.

Definição 2.11. A função argumento de q define-se como,

$$\arg(q) = \begin{cases} \arccos\left(\frac{q_0}{\|q\|}\right), & \text{se } q \neq 0, \\ \text{indefinido}, & \text{se } q = 0. \end{cases} \quad (2.21)$$

Proposição 2.10. A função exponencial admite a representação:

$$e^q = e^{q_0}(\cos \| \mathbf{q} \| + \omega(\mathbf{q}) \text{sen } \| \mathbf{q} \|).$$

Para um quaterniões puro da forma $q = \omega(\mathbf{q})\theta$, $\theta \in \mathbb{R}$,

$$e^q = \cos \theta + \omega(\mathbf{q}) \operatorname{sen} \theta.$$

e, conseqüentemente, $\|e^q\| = 1$. Neste caso, obtém-se uma função de $\mathbb{H}_{\mathbb{R}^3}$ em \mathbb{H}_1 .

Proposição 2.11. *Se p e q são quaterniões tais que $pq = qp$ então*

$$e^{p+q} = e^p e^q.$$

Nota 2.3. A comutatividade é uma condição suficiente mas não necessária para o estabelecimento da proposição anterior. De facto, se $p = i$ e $q = j$, então $e^p = e^i = \cos 1 + i \operatorname{sen} 1$ e $e^q = e^j = \cos 1 + j \operatorname{sen} 1$ e

$$\begin{aligned} e^p \cdot e^q &= (\cos 1 + i \operatorname{sen} 1)(\cos 1 + j \operatorname{sen} 1) \\ &= (\cos 1)^2 + \cos 1 \operatorname{sen} 1 i + \cos 1 \operatorname{sen} 1 j + (\operatorname{sen} 1)^2 \approx 1 + 0.4546i + 0.4546j. \\ e^{p+q} = e^{i+j} &= \cos \sqrt{2} + \frac{i+j}{\sqrt{2}} \operatorname{sen} \sqrt{2} \approx 0.1559 + \frac{i+j}{\sqrt{2}} \cdot 0.9878. \end{aligned}$$

Por outro lado, se $p = 3\pi i$ e $q = 4\pi j$ então $e^{p+q} = \cos 5\pi + \frac{3\pi i + 4\pi j}{5\pi} \operatorname{sen} 5\pi = -1$ e

$$e^p = \cos 3\pi = -1 \quad e \quad e^q = \cos 4\pi = 1.$$

Concluindo, verifica-se que $e^{p+q} = e^p e^q$.

A função exponencial verifica as seguintes propriedades:

Proposição 2.12. *Para qualquer quaterniões $q = q_0 + \mathbf{q}$, tem-se:*

(a) $e^q \neq 0$;

(b) $e^{-q} e^q = 1$;

(c) $e^{\omega(\mathbf{q})\pi} = -1$;

(d) $e^{k\mathbf{q}} = (e^{\mathbf{q}})^k \quad (k \in \mathbb{N}) \quad (\text{Fórmula de Moivre});$

(e) $\|e^q\| = e^{q_0}$.

Definição 2.12. Função logaritmo

Seja $q = q_0 + \mathbf{q}$ um quatérnião tal que $\mathbf{q} \neq 0$ ou $q = q_0 > 0$. Então a função logarítmica $\log q$ define-se por:

$$\log q = \log \|q\| + \omega(\mathbf{q})\arg(q).$$

Observe-se que, se $q \in \mathbb{H}_1$, i.e. se $q = \cos \theta + \omega(\mathbf{q}) \operatorname{sen} \theta$ então:

$$\log q = \log(\cos \theta + \omega(\mathbf{q}) \operatorname{sen} \theta) = \omega(\mathbf{q})\theta.$$

ou seja, obtém-se uma função definida de \mathbb{H}_1 em $\mathbb{H}_{\mathbb{R}^3}$.

Saliente-se ainda que, se q é um quatérnião unitário, $\log q$ não é, geralmente, um quatérnião unitário.

Proposição 2.13. Seja $pq = qp$. Então

$$\log pq = \log p + \log q. \quad (2.22)$$

Nota 2.4. À semelhança da função exponencial, a comutatividade é também uma condição suficiente mas não necessária para a validade da proposição anterior, pelo que se $pq \neq qp$, a regra (2.22) nem sempre é verdadeira. Por exemplo,

$$\log i = i\frac{\pi}{2}, \quad \log j = j\frac{\pi}{2}, \quad \log k = k\frac{\pi}{2},$$

mas

$$\begin{aligned} \log ij &= \log k = k\frac{\pi}{2}, \\ \log i + \log j &= i\frac{\pi}{2} + j\frac{\pi}{2} = (i + j)\frac{\pi}{2}. \end{aligned}$$

Definição 2.13. Potenciação

A função potencia p^q define-se por:

$$p^q = e^{q \log p}.$$

As fórmulas de Moivre, conhecidas em \mathbb{C} , podem ser generalizadas para os quaterniões [LS87].

Teorema 2.2. Fórmula de Moivre

Seja $q \in \mathbb{H}$, $\mathbf{q} \neq 0$, $n \in \mathbb{N}$, então a seguinte fórmula é válida:

$$q^n = (\|q\| \cos \theta + \omega(\mathbf{q}) \operatorname{sen} \theta)^n = \|q\|^n (\cos n\theta + \omega(\mathbf{q}) \operatorname{sen} n\theta).$$

Proposição 2.14. Se $q = \|q\|(\cos \theta + \omega(\mathbf{q}) \operatorname{sen} \theta) \in \mathbb{H}$, $\mathbf{q} \neq 0$ então

$$\sqrt[n]{q} = \sqrt[n]{\|q\|} \left(\cos \frac{\theta + 2k\pi}{n} + \omega(\mathbf{q}) \operatorname{sen} \frac{\theta + 2k\pi}{n} \right), \quad k = 0, \dots, n-1.$$

A restrição indicada na proposição anterior deve-se ao facto de haver quaterniões, com parte vectorial nula, que possuem um número infinito de raízes. Em \mathbb{C} , -1 possui duas raízes quadradas: i e $-i$. Em \mathbb{H} , -1 tem um número infinito de raízes quadradas. De facto, todo o quaternião puro e unitário é raiz quadrada de -1 , uma vez que

$$q^2 = (q_1i + q_2j + q_3k)^2 = -(q_1^2 + q_2^2 + q_3^2) = -1.$$

Exemplo 2.4. Para o quaternião $q = 2i - 2j + k$, tem-se que $\|q\| = 3$, $\omega(\mathbf{q}) = \frac{2i - 2j + k}{3}$ e $\theta = \frac{\pi}{2}$, i.e.

$$q = 3 \left(\cos \frac{\pi}{2} + \frac{2i - 2j + k}{3} \operatorname{sen} \frac{\pi}{2} \right).$$

Então

$$\sqrt{q} = \sqrt{3} \left(\cos \frac{\theta + 2k\pi}{2} + \omega(\mathbf{q}) \operatorname{sen} \frac{\theta + 2k\pi}{2} \right), \quad k = 0, 1.$$

Assim, para $k=0$, tem-se:

$$r_1 = \sqrt{3} \left(\cos \frac{\theta}{2} + \omega(\mathbf{q}) \operatorname{sen} \frac{\theta}{2} \right) = \frac{\sqrt{6}}{2} + \frac{2\sqrt{6}}{6}i - \frac{2\sqrt{6}}{6}j + \frac{\sqrt{6}}{6}k.$$

e para $k=1$,

$$r_2 = \sqrt{3} \left(\cos \frac{\theta + 2\pi}{2} + \omega(\mathbf{q}) \operatorname{sen} \frac{\theta + 2\pi}{2} \right) = -\frac{\sqrt{6}}{2} - \frac{2\sqrt{6}}{6}i + \frac{2\sqrt{6}}{6}j - \frac{\sqrt{6}}{6}k.$$

Definição 2.14. *Funções trigonométricas e hiperbólicas*

Seja $q = q_0 + \mathbf{q}$ um quaterniões. Então as funções trigonométricas e hiperbólicas definem-se por:

$$\begin{aligned} \operatorname{sen} q &= \frac{e^{q\omega(\mathbf{q})} - e^{-q\omega(\mathbf{q})}}{2} \omega(\mathbf{q}), & \cos q &= \frac{e^{q\omega(\mathbf{q})} + e^{-q\omega(\mathbf{q})}}{2}. \\ \operatorname{senh} q &= \frac{e^q - e^{-q}}{2}, & \operatorname{cosh} q &= \frac{e^q + e^{-q}}{2}. \end{aligned}$$

Facilmente se verifica que:

$$\operatorname{sen} q = -\operatorname{sen}(-q) \quad e \quad \cos q = \cos(-q).$$

As proposições seguintes são consequências imediatas da definição anterior.

Proposição 2.15. *As funções trigonométricas $\operatorname{sen} q$ e $\cos q$ permitem as seguintes representações:*

$$\begin{aligned} \operatorname{sen} q &= \frac{e^{\|\mathbf{q}\|} + e^{-\|\mathbf{q}\|}}{2} \operatorname{sen} q_0 + \omega(\mathbf{q}) \frac{e^{\|\mathbf{q}\|} - e^{-\|\mathbf{q}\|}}{2} \cos q_0 \\ &= \operatorname{cosh} \|\mathbf{q}\| \operatorname{sen} q_0 + \omega(\mathbf{q}) \operatorname{senh} \|\mathbf{q}\| \cos q_0, \\ \cos q &= \frac{e^{-\|\mathbf{q}\|} + e^{\|\mathbf{q}\|}}{2} \cos q_0 + \omega(\mathbf{q}) \frac{e^{-\|\mathbf{q}\|} - e^{\|\mathbf{q}\|}}{2} \operatorname{sen} q_0 \\ &= \operatorname{cosh} \|\mathbf{q}\| \cos q_0 - \omega(\mathbf{q}) \operatorname{senh} \|\mathbf{q}\| \operatorname{sen} q_0. \end{aligned}$$

Proposição 2.16. *As funções hiperbólicas $\operatorname{senh} q$ e $\operatorname{cosh} q$ permitem as seguintes representações:*

$$\begin{aligned}\sinh q &= \frac{e^{q_0} - e^{-q_0}}{2} \cos \|\mathbf{q}\| + \omega(\mathbf{q}) \frac{e^{q_0} + e^{-q_0}}{2} \operatorname{sen} \|\mathbf{q}\| \\ &= \sinh q_0 \cos \|\mathbf{q}\| + \omega(\|\mathbf{q}\|) \cosh q_0 \operatorname{sen} \|\mathbf{q}\|,\end{aligned}$$

$$\begin{aligned}\cosh q &= \frac{e^{q_0} + e^{-q_0}}{2} \cos \|\mathbf{q}\| + \omega(\mathbf{q}) \frac{e^{q_0} - e^{-q_0}}{2} \operatorname{sen} \|\mathbf{q}\| \\ &= \cosh q_0 \cos \|\mathbf{q}\| + \omega(\|\mathbf{q}\|) \sinh q_0 \operatorname{sen} \|\mathbf{q}\|.\end{aligned}$$

Proposição 2.17. *Sejam $p, q \in \mathbb{H}$. Então,*

- (a) $\cosh q = \cos(\omega(\mathbf{q})q)$;
- (b) $\sinh q = \omega(\mathbf{q}) \operatorname{sen}(\omega(\mathbf{q})q)$;
- (c) $\cosh \mathbf{q} = \cos \|\mathbf{q}\|$;
- (d) $\sinh \mathbf{q} = -\omega(\mathbf{q}) \operatorname{sen} \|\mathbf{q}\|$;
- (e) $\operatorname{sen}(p + q) = \operatorname{sen} p \cos q + \cos p \operatorname{sen} q$;
- (f) $\cos(p + q) = \cos p \cos q - \operatorname{sen} p \operatorname{sen} q$;
- (g) $\operatorname{sen}(k\pi + \mathbf{q}) = \omega(\mathbf{q})(-1)^k \sinh \mathbf{q} \quad (k \in \mathbb{N})$;
- (h) $\cos(k\pi + \mathbf{q}) = (-1)^k \cosh \mathbf{q}$.

Deste modo, conclui-se que $\cos q$ e $\operatorname{sen} q$ não têm outros zeros para além dos bem conhecidos reais. Assim, torna-se claro que as funções tangente e cotangente podem ser introduzidas.

Definição 2.15. *Seja $q = q_0 + \mathbf{q}$ um quaterniões tal que $\|\mathbf{q}\| \neq 0$. Então as funções tangente e cotangente definem-se por:*

$$\tan q = \operatorname{sen} q / \cos q, \quad \cos q \neq 0 \quad e \quad \cot q = \cos q / \operatorname{sen} q, \quad \operatorname{sen} q \neq 0.$$

Note-se que, na definição anterior, a divisão à direita pode ser substituída pela divisão à esquerda, porque neste caso, a ordem do produto é irrelevante.

Inversa das funções hiperbólicas

$$\begin{aligned}\operatorname{arcsenh} q &= \log(q + \sqrt{q^2 + 1}), \\ \operatorname{arccosh} q &= \log(q + \sqrt{q^2 - 1}), \\ \operatorname{arctanh} q &= \frac{\log(1 + q) - \log(1 - q)}{2}.\end{aligned}$$

Inversa das funções trigonométricas

$$\begin{aligned}\operatorname{arcsen} q &= -\omega(\mathbf{q})\operatorname{arcsenh}(q\omega(\mathbf{q})), \\ \operatorname{arccos} q &= -\omega(\mathbf{q})\operatorname{arccosh}(q), \\ \operatorname{arctan} q &= -\omega(\mathbf{q})\operatorname{arctanh}(q\omega(\mathbf{q})).\end{aligned}$$

Capítulo 3

Quaterniões e Rotações

As rotações são uma das possíveis transformações no espaço, que correspondem a mudanças de orientação e de posição. O primeiro estudo sobre rotações foi publicado por Euler em 1776. Desde então a parametrização de rotações finitas tem sido objecto de contínua análise devido à sua extensa aplicação em diversas áreas já referenciadas, como a pesquisa aeronáutica e aeroespacial, a computação gráfica, a animação computacional, a robótica e a visão por computador.

Neste capítulo, faz-se uma breve referência a rotações no plano, seguida de uma análise mais detalhada de rotações em \mathbb{R}^3 . Em particular, apresentam-se quatro representações usuais de rotações: matrizes de rotação, ângulos de Euler, eixo-ângulo e quaterniões. Para cada representação, efectua-se uma descrição do processo de rotação, define-se a concatenação de rotações e conclui-se com um balanço das vantagens e desvantagens. As representações são comparadas em função de características como o desempenho, o espaço necessário para armazenamento, a estabilidade numérica e a facilidade de utilização. Das representações citadas é dado um ênfase particular aos quaterniões, que motivaram a realização deste trabalho, e cuja importância tem sido reconhecida, face às vantagens que apresentam, em relação às restantes parametrizações. A interpolação de rotações, embora muito importante, não está no âmbito deste trabalho. No entanto, e face à aplicação prática que tem, não pode ser dissociada da análise de uma representação pelo que será ponderada nas vantagens e desvantagens de cada parametrização.

Como referências principais para o tema e que serviram de suporte a este trabalho, apontamos os livros de Kuipers [Kui02] e Hanson [Han01], os artigos de Shoemake [Sho85], Dam et al [DKL98] e Eberly [Ebe04].

3.1 Representação de Rotações em \mathbb{R}^2

Começamos por distinguir duas perspectivas sobre as rotações no plano e analisar o efeito produzido sobre as coordenadas dos pontos (ou vectores) envolvidos na rotação. A primeira é a rotação do sistema de coordenadas - *sistema de coordenadas móvel* - em

relação a pontos fixos no plano; a segunda corresponde à rotação de pontos (ou vectores) sobre um *sistema de coordenadas fixo*.

- *Sistema de coordenadas móvel*

Consideremos um ponto $P(x_1, y_1)$ num sistema de coordenadas XOY conforme mostra a Figura 3.1.

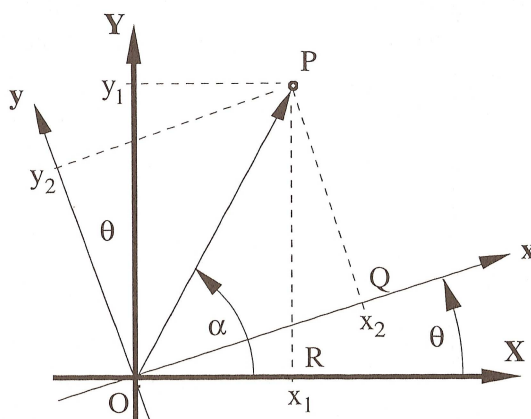


Figura 3.1: Eixos coordenados móveis

Após efectuar uma rotação de ângulo θ (positivo), do sistema de coordenadas XOY, em relação à origem, obtém-se um novo sistema de coordenadas xOy. Sejam (x_2, y_2) as coordenadas do ponto P no novo sistema. Iremos determinar a relação entre as novas coordenadas do ponto P e as iniciais.

Seja r o comprimento do vector \overrightarrow{OP} e α o ângulo entre o vector e o eixo positivo O_X . Uma vez que o ângulo entre o vector \overrightarrow{OP} e o novo eixo O_x é $\alpha - \theta$, obtém-se,

$$\begin{aligned} x_1 &= r \cos \alpha & x_2 &= r \cos(\alpha - \theta) = r \cos \alpha \cos \theta + r \operatorname{sen} \alpha \operatorname{sen} \theta \\ y_1 &= r \operatorname{sen} \alpha & y_2 &= r \operatorname{sen}(\alpha - \theta) = r \operatorname{sen} \alpha \cos \theta - r \cos \alpha \operatorname{sen} \theta. \end{aligned}$$

e conclui-se, de imediato, que a relação entre as coordenadas é:

$$x_2 = x_1 \cos \theta + y_1 \operatorname{sen} \theta, \quad (3.1)$$

$$y_2 = y_1 \cos \theta - x_1 \operatorname{sen} \theta. \quad (3.2)$$

- *Sistema de coordenadas fixo*

Na derivação anterior, pensa-se no ponto P (ou vector \overrightarrow{OP}) como estando fixo enquanto que os eixos de coordenadas sofrem uma rotação de ângulo θ , em relação à origem. No entanto, outra situação é pensar que os eixos estão fixos, enquanto que um ponto P_1 de coordenadas (x_1, y_1) , se *desloca* para um ponto P_2 de coordenadas (x_2, y_2) , como consequência de uma rotação de ângulo θ em relação à origem, como mostra a Figura 3.2. Neste caso, tem-se:

$$\begin{aligned} x_1 &= r \cos \alpha & e & & x_2 &= r \cos(\alpha + \theta) = r \cos \alpha \cos \theta - r \sin \alpha \sin \theta \\ y_1 &= r \sin \alpha & & & y_2 &= r \sin(\alpha + \theta) = r \sin \alpha \cos \theta + r \cos \alpha \sin \theta. \end{aligned}$$

e a relação entre as coordenadas é:

$$x_2 = x_1 \cos \theta - y_1 \sin \theta, \quad (3.3)$$

$$y_2 = y_1 \cos \theta + x_1 \sin \theta. \quad (3.4)$$

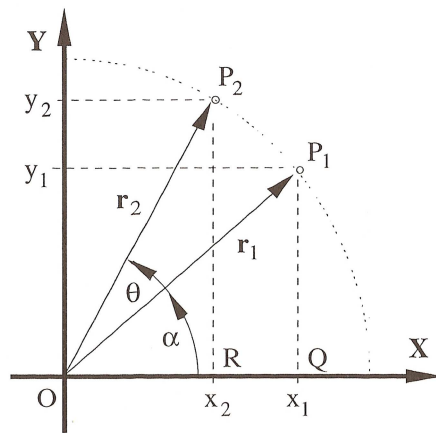


Figura 3.2: Eixos coordenados fixos

Note-se que as relações entre coordenadas das duas perspectivas apresentadas diferem apenas nos sinais de $\sin \theta$. O exemplo seguinte ilustra o significado desta diferença.

Exemplo 3.1. Consideremos um ponto P de coordenadas $(1, 1)$ no sistema XOY, como se mostra na Figura 3.3.

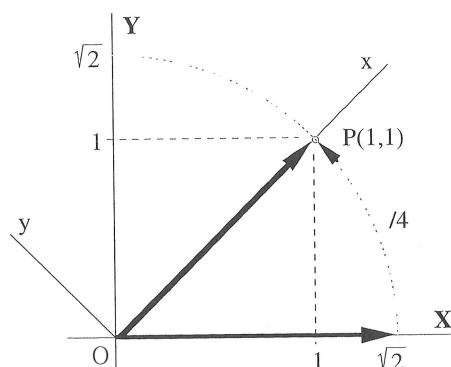


Figura 3.3: Exemplo das duas perspectivas de eixos coordenados

De (3.1) e (3.2) resulta que, se realizarmos uma rotação do sistema de coordenadas no sentido positivo, por um ângulo $\theta = \pi/4$, as coordenadas resultantes para o ponto P, no novo sistema, são $(\sqrt{2}, 0)$. Assim, no sistema de coordenadas xOy o ponto P está sobre o eixo O_x . Se, no entanto, for efectuada a rotação do vector \overrightarrow{OP} , na perspectiva *sistema fixo*, as coordenadas resultantes para P , usando (3.3) e (3.4), são $(0, \sqrt{2})$. Nesta situação o ponto P está sobre o eixo O_Y .

As equações (3.1) e (3.2), da perspectiva *sistema de coordenadas móvel*, podem ser escritas na forma matricial:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & \text{sen } \theta \\ -\text{sen } \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}.$$

Denotando por \mathbf{x} e \mathbf{y} , respectivamente, os vectores coluna $\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$ e $\begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$ e por A a matriz

$$A = \begin{bmatrix} \cos \theta & \text{sen } \theta \\ -\text{sen } \theta & \cos \theta \end{bmatrix}, \quad (3.5)$$

então as equações (3.1) e (3.2) reescrevem-se matricialmente, de forma única, como

$$\mathbf{y} = A\mathbf{x}.$$

Nesta equação, A é chamada *matriz de rotação* ou *operador de rotação*, que transforma \mathbf{x} em \mathbf{y} . De imediato se conclui que, para a perspectiva *sistema de coordenadas fixo*, a

matriz de rotação é, precisamente, A^T .

Facilmente se verifica também que a matriz de rotação A (e, naturalmente, a sua transposta) é uma matriz ortogonal com determinante igual a 1. Além disso, pode provar-se ([Kui02]) que o recíproco também é válido, i.e. toda a matriz 2×2 ortogonal e com determinante 1 é uma matriz de rotação. Por outras palavras, as matrizes ortogonais 2×2 cujo determinante é 1, normalmente designadas *matrizes ortogonais especiais*, representam todas as rotações no plano. O conjunto destas matrizes, munido com a operação de multiplicação, forma um grupo, habitualmente denotado por \mathbf{SO}_2 .

Identificando a matriz de rotação (3.5) com o número complexo $\cos \theta + i \operatorname{sen} \theta$ (cf. com a matriz (2.12)), obtém-se

$$\mathbf{y} = (\cos \theta + i \operatorname{sen} \theta)\mathbf{x},$$

i.e. o conjunto dos números complexos unitários pode ser identificado com o conjunto das rotações no plano.

Note-se que, do ponto de vista do espaço necessário para armazenamento no computador e do número de operações envolvidas, a representação complexa é computacionalmente mais vantajosa que a representação matricial.

Finalmente, consideremos o resultado da sequência de duas rotações. Se efectuarmos uma rotação no *sistema de coordenadas móvel* por um ângulo α seguida de uma rotação do sistema de coordenadas resultante por um ângulo β , facilmente se verifica que o resultado é uma rotação do sistema de coordenadas inicial por um ângulo $\alpha + \beta$. Em particular, se R_α for a matriz da primeira rotação e R_β a matriz da segunda, a matriz de rotação da sequência das duas rotações é o produto das duas matrizes de rotação, R_β e R_α , i.e.

$$R_\beta R_\alpha = R_{\alpha+\beta},$$

donde se conclui também que a ordem pela qual se efectuam as rotações é irrelevante; a composição de rotações no plano é comutativa.

3.2 Representação de Rotações em \mathbb{R}^3

Nesta secção iremos estender o conceito de rotação, as suas diversas representações e respectivas características ao espaço tridimensional. As rotações em três dimensões são mais complexas do que as correspondentes em duas dimensões porque é necessário especificar um eixo de rotação em vez de um ponto de rotação. Em \mathbb{R}^2 , o eixo de rotação é sempre perpendicular ao plano xOy enquanto que em \mathbb{R}^3 o eixo de rotação pode ter qualquer orientação espacial. Iremos analisar as rotações à volta dos três eixos principais O_x, O_y, O_z assim como as rotações à volta de um eixo arbitrário.

Assume-se o sistema de coordenadas dito da mão direita, pelo que todas as rotações ditas positivas são no sentido contrário ao dos ponteiros do relógio. As duas perspectivas de representação de rotações no plano apresentadas na secção anterior - *sistema de coordenadas fixo* ou *sistema de coordenadas móvel* - são também consideradas em \mathbb{R}^3 . Note-se que na computação gráfica, os dois tipos de rotações são geralmente utilizadas, em função dos eixos estarem ligados ao écran ou ao objecto gráfico que está a ser manipulado.

As rotações em \mathbb{R}^3 podem ser representadas, matemática e computacionalmente, de diversas formas, das quais salientamos as que descrevemos de seguida:

- matrizes de rotação;
- ângulos de Euler;
- eixo-ângulo;
- quaterniões.

Matrizes de rotação

Nesta subsecção as rotações serão apresentadas na perspectiva *sistema de coordenadas móvel*. Começemos por notar que uma rotação 2D no plano XOY descrita anteriormente, corresponde a uma rotação 3D, em torno do eixo O_z . De facto, a rotação de um ponto

P de coordenadas (x_1, y_1, z_1) , em torno do eixo O_z , não depende da coordenada z_1 desse ponto e todos os pontos do eixo O_z são pontos fixos da rotação. De (3.1) e (3.2), conclui-se, de imediato, que as coordenadas (x_2, y_2, z_2) de P no sistema obtido após uma rotação de ângulo θ em torno do eixo O_z , satisfazem

$$x_2 = x_1 \cos \theta + y_1 \operatorname{sen} \theta,$$

$$y_2 = y_1 \cos \theta - x_1 \operatorname{sen} \theta,$$

$$z_2 = z_1.$$

Matricialmente, estas equações podem ser expressas como:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & \operatorname{sen} \theta & 0 \\ -\operatorname{sen} \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}. \quad (3.6)$$

As rotações à volta dos outros eixos definem-se de modo análogo. Mais precisamente, as matrizes de rotação em torno do eixo O_x e O_y são, respectivamente:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \operatorname{sen} \theta \\ 0 & -\operatorname{sen} \theta & \cos \theta \end{bmatrix}, \quad (3.7)$$

$$\begin{bmatrix} \cos \theta & 0 & -\operatorname{sen} \theta \\ 0 & 1 & 0 \\ \operatorname{sen} \theta & 0 & \cos \theta \end{bmatrix}. \quad (3.8)$$

Saliente-se que, tal como no caso 2D, o grupo das matrizes ortogonais especiais de ordem 3, \mathbf{SO}_3 representa o conjunto de todas as rotações em \mathbb{R}^3 (ver, p.e. [Kui02]).

Rotações sucessivas à volta do mesmo eixo

Consideremos uma sequência de duas rotações à volta do mesmo eixo, a primeira por um ângulo α seguida por uma segunda, de ângulo β . Facilmente se verifica que, conforme

acontece em \mathbb{R}^2 , a matriz de rotação da composição é obtida pela multiplicação das matrizes de rotações individuais, i.e.,

$$R_\alpha R_\beta = R_{\alpha+\beta}.$$

Qualquer sequência de duas rotações sucessivas à volta do mesmo eixo, corresponde a uma rotação à volta desse eixo por um ângulo resultante da soma dos ângulos das rotações da sequência.

Sequências de rotações em \mathbb{R}^3

Observe-se agora a sequência de duas rotações, digamos

$$R = R_2 R_1,$$

em que as rotações não são realizadas à volta do mesmo eixo de coordenadas. O resultado desta sequência é ainda uma rotação à volta de um eixo, que usualmente não é um eixo de coordenadas, por um certo ângulo. Assim, iremos determinar um operador de rotação para esta sequência, o *eixo fixo* de rotação e o ângulo de rotação à volta desse eixo.

Geometricamente é claro que, em \mathbb{R}^3 , numa rotação em torno de um eixo que passa pela origem do referencial, o próprio eixo não se altera sob essa rotação, i.e., se \mathbf{v} designar o eixo de rotação que se pretende determinar e R designar a matriz de rotação, então

$$\mathbf{v} = R\mathbf{v}.$$

Da igualdade anterior conclui-se que \mathbf{v} é um vector próprio de R associado ao valor próprio real 1. Se $R = (r_{ij})$, uma das possíveis expressões para o vector próprio associado ao valor próprio 1, ou equivalentemente, para o eixo de rotação é

$$\mathbf{v} = (v_1, v_2, v_3),$$

$$\text{onde } v_1 = r_{12}r_{23} - (r_{22} - 1)r_{13},$$

$$v_2 = r_{21}r_{13} - (r_{11} - 1)r_{23},$$

$$v_3 = (r_{11} - 1)(r_{22} - 1) - r_{12}r_{21}.$$

Por fim iremos determinar a magnitude e direcção da rotação à volta do eixo fixo. Pode provar-se (ver, por exemplo [Kui02]) que o ângulo θ associado a uma rotação em torno de um eixo fixo é dado por

$$\theta = \arccos \frac{\text{tr}(R) - 1}{2}, \quad (3.9)$$

onde $\text{tr}(R)$ é o traço da matriz de rotação R .

Uma matriz de rotação pode ser facilmente produzida, se a base do espaço que sofreu a rotação é conhecida, porque corresponde à matriz cujas colunas formam a base. Apresentamos de seguida um exemplo:

Exemplo 3.2. Consideremos num sistema de coordenadas XYZ , uma rotação à volta do vector $\mathbf{v} = (1, 1, 1)$, por um ângulo $\theta = 2\pi/3$.

Geometricamente é claro que tal rotação resulta num novo sistema coordenado xyz no qual o novo eixo O_x coincide com o eixo inicial O_Y , o novo eixo O_y coincide com o eixo inicial O_Z e o novo eixo O_z coincide com o eixo inicial O_X .

Procuremos a matriz R que representa esta rotação. Uma vez que

$$(1, 0, 0) \mapsto (0, 0, 1),$$

$$(0, 1, 0) \mapsto (1, 0, 0),$$

$$(0, 0, 1) \mapsto (0, 1, 0),$$

a matriz R é dada por

$$R = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Podemos agora calcular as coordenadas de qualquer vector $\mathbf{v} = (x, y, z)$ calculando o produto $R\mathbf{v}$.

Suponhamos agora que dada a matriz R , queremos saber que tipo de transformação representa em \mathbb{R}^3 . Verificando que R é ortogonal e que o determinante é 1, podemos afirmar

que R representa uma rotação. É imediato concluir que $(1,1,1)$ é um vector próprio de R associado ao valor próprio 1, pelo que acabámos de encontrar o eixo de rotação.

Finalmente o traço da matriz R é 0, pelo que o ângulo de rotação é dado por

$$\alpha = \arccos(-1/2) = 2\pi/3.$$

As matrizes de rotação em \mathbb{R}^3 apresentam duas novas características em relação às matrizes de rotação em \mathbb{R}^2 :

- Com excepção de alguns casos especiais as rotações em \mathbb{R}^3 não comutam pelo que, para duas matrizes de rotação R_1 e R_2 :

$$R_1R_2 \neq R_2R_1.$$

- As matrizes de rotação em \mathbb{R}^3 têm um valor próprio real e, conseqüentemente, todas as matrizes de rotação (e os seus produtos) podem ser escritas em termos de uma matriz de rotação final, que deixa inalterada uma determinada direcção fixa.

Uma vez que R não altera o eixo, esta matriz expressa todas as possíveis rotações por um dado ângulo à volta de uma direcção.

Denotemos por $R(\theta, \mathbf{v})$ a matriz correspondente a uma rotação de ângulo θ , em torno de um eixo \mathbf{v} . Esta matriz pode ser construída do seguinte modo:

O eixo fixo \mathbf{v} à volta do qual se pretende realizar a rotação pode escrever-se como

$$\mathbf{v} = (\cos \alpha \sin \beta, \sin \alpha \sin \beta, \cos \beta), \text{ com } 0 \leq \alpha < 2\pi, 0 \leq \beta < \pi.$$

Se $\mathbf{y} = (0, 1, 0)$ e $\mathbf{z} = (0, 0, 1)$, tem-se que:

$$\mathbf{v} = R(\alpha, \mathbf{z})R(\beta, \mathbf{y})\mathbf{z}.$$

Para construir a matriz de rotação que efectua a rotação à volta de \mathbf{v} , é necessário transformar \mathbf{v} em \mathbf{z} invertendo a transformação previamente citada, realizar a rotação de

\mathbf{z} por θ , e voltar a inclinar \mathbf{z} na direcção de \mathbf{v} (onde começou):

$$R(\theta, \mathbf{v}) = R(\alpha, \mathbf{z}) \cdot R(\beta, \mathbf{y}) \cdot R(\theta, \mathbf{z}) \cdot R(\beta, \mathbf{y})^T \cdot R(\alpha, \mathbf{z})^T.$$

Calculando todas as componentes deste produto de matrizes, e reescrevendo todas as ocorrências de α e β em termos de

$$\mathbf{v} = (\cos \alpha \sin \beta, \sin \alpha \sin \beta, \cos \beta) = (v_1, v_2, v_3),$$

obtém-se, a matriz de rotação:

$$R(\theta, \mathbf{v}) = \begin{bmatrix} c + (1-c)v_1^2 & (1-c)v_1v_2 - (s)v_3 & (1-c)v_1v_3 + (s)v_2 \\ (1-c)v_2v_1 + (s)v_3 & c + (1-c)v_2^2 & (1-c)v_2v_3 - (s)v_1 \\ (1-c)v_3v_1 - (s)v_2 & (1-c)v_3v_2 + (s)v_1 & c + (1-c)v_3^2 \end{bmatrix}, \quad (3.10)$$

onde $c = \cos \theta$ e $s = \sin \theta$. Em conclusão, a matriz anterior representa a rotação, por um ângulo de rotação θ , em torno de um eixo $\mathbf{v} = (v_1, v_2, v_3)$.

Salientamos que realizar uma rotação em torno de um eixo que não passe pela origem é mais trabalhosa. Assim, dados dois pontos p_1 e p_2 , define-se o eixo de rotação e a seguir efectuam-se as seguintes transformações:

1. Translação para que o eixo de rotação passe pela origem;
2. Rotação para que o eixo de rotação fique alinhado com um dos principais eixos de coordenadas (O_x, O_y e O_z);
3. Rotação do objecto à volta do eixo de coordenadas;
4. Rotação inversa à realizada em (2.);
5. Translação inversa à realizada em (1.).

Representação de matrizes com coordenadas homogêneas

No espaço tridimensional, as rotações estão frequentemente associadas a outros tipos de transformação, como as translações, mudanças de escala, reflexões, etc. Por exemplo, a

posição geral de um corpo rígido pode ser dada pela combinação de uma translação com uma rotação. As matrizes 3×3 não permitem representar todas as transformações referidas, e em particular as translações. Para solucionar este problema são utilizadas matrizes de transformação 4×4 com coordenadas homogêneas, que permitem representar de forma compacta, numa única entidade matemática, a composição de diversas transformações no espaço.

Consideremos uma transformação que leva um referencial R_i noutra referencial R_j , como ilustra a Figura 3.4.

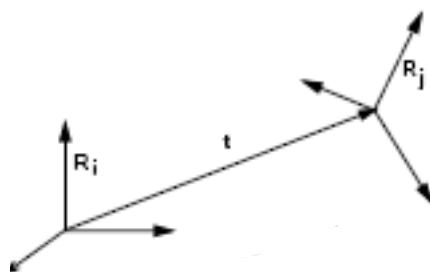


Figura 3.4: Transformação entre referenciais

Esta transformação pode ser representada por uma matriz homogênea 4×4 :

$$A = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^t & 1 \end{bmatrix},$$

em que R representa a matriz de rotação, de dimensão 3×3 e \mathbf{t} representa o vector de translação.

Vantagens da representação matricial

Em termos matemáticos, a representação matricial parece ideal para definir rotações, uma vez que \mathbf{SO}_3 é exactamente o grupo que se pretende representar:

- A representação matricial para uma rotação em \mathbb{R}^3 é única. Ver-se-á que as outras representações não gozam desta propriedade de um para um, incluindo os quaterniões.

- As matrizes facilitam os cálculos, a composição de rotações exprime-se por um simples produto, a rotação inversa pela transposta. Além disso, as matrizes de rotação podem ser compostas a outros tipos de matrizes, como por exemplo, homotetias e inversões.
- As operações matemáticas sobre matrizes são bem conhecidas.
- A principal vantagem da representação matricial é o facto da matriz homogénea representar todas as transformações básicas (*e.g.*, translação, escala, projecção, *shear*).

Desvantagens:

Esta representação, matematicamente conveniente, apresenta vários problemas quando implementada num sistema computacional de precisão finita. Assim,

- Utilizando matrizes 3×3 , são necessários 9 parâmetros para representar uma estrutura com apenas 3 graus de liberdade. Significa que seis restrições não lineares deverão ser impostas para garantir que os vectores colunas são ortonormais e que o determinante é positivo. Quando uma das seis restrições é violada, as matrizes têm propensão para “andar à deriva”, dado que a matriz introduz rotações à volta de um eixo arbitrário. Combater este problema requer conservar a ortogonalidade da matriz, i.e., garantir que obedece às restrições, o que é computacionalmente caro, se for feito com frequência. Uma forma comum que preservar as restrições é utilizar o algoritmo Gram-Schmidt para calcular a matriz corrigida.
- Quando muitas rotações são concatenadas numericamente, podem surgir erros de arredondamentos que também provocam a perda da ortogonalidade especial, pois introduzem efeitos de escalonamento e corte. Para ultrapassar este problema pode recorrer-se de novo ao processo de ortogonalização de Gram-Schmidt mas esta solução pode consumir muitos ciclos de CPU, como já foi referido.
- Uma matriz de rotação pode ser difícil de construir se não for conhecida a base do espaço em que se realiza a rotação.

- Outro problema com as matrizes de rotação é a dificuldade de interpolar rotações entre duas orientações. As interpolações resultantes são visualmente irregulares e bruscas, o que não é actualmente aceitável (por exemplo, em jogos de computador).

Ângulos de Euler

Os ângulos de Euler são uma alternativa para representar rotações. A parametrização de \mathbf{SO}_3 por ângulos de Euler utiliza o facto de que qualquer rotação pode ser descrita por três rotações sucessivas à volta dos eixos O_x , O_y e O_z .

Definição 3.1. Chamamos ângulos de Euler aos ângulos α , β e γ cujas respectivas matrizes são dadas por

$$R(\alpha, \mathbf{x}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \text{sen } \alpha \\ 0 & -\text{sen } \alpha & \cos \alpha \end{bmatrix},$$

$$R(\beta, \mathbf{y}) = \begin{bmatrix} \cos \beta & 0 & -\text{sen } \beta \\ 0 & 1 & 0 \\ \text{sen } \beta & 0 & \cos \beta \end{bmatrix},$$

$$R(\gamma, \mathbf{z}) = \begin{bmatrix} \cos \gamma & \text{sen } \gamma & 0 \\ -\text{sen } \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Estas matrizes foram introduzidas em (3.7), (3.8) e (3.6), por surgirem de forma intuitiva das equações que definem as rotações à volta dos eixos coordenados.

Uma rotação definida pelos ângulos de Euler (α, β, γ) resulta da factorização de três rotações sequenciais à volta dos três eixos de coordenadas. Uma vez que as rotações à volta dos eixos não comutam, a ordem pela qual as rotações são aplicadas é importante. Qualquer combinação de eixos pode ser utilizada para representar rotações pelo que existem doze possibilidades para compor as três rotações elementares:

$$xyz, xyx, yzx, yzy, zxy, zxz, xzy, xzx, yxz, yxy, zyx, zyz.$$

Usualmente os ângulos de Euler referem-se à formulação de eixo móvel. No entanto, tal como já foi referido, a perspectiva *sistema fixo* obtém-se do *sistema móvel*, invertendo

a ordem da multiplicação das matrizes dos três factores. Por exemplo, uma rotação no sistema de coordenadas fixo expressa pela combinação de eixos xyz é a mesma que a realizada no sistema de coordenadas móvel zyx .

Estando os ângulos de Euler definidos na perspectiva *sistema móvel*, uma rotação à volta de um eixo em movimento significa que a rotação é realizada à volta de eixos “locais”: primeiro à volta do eixo local O_z , a seguir à volta do novo eixo correspondente a O_y que se denota por O_{y_1} , por fim à volta do novo eixo correspondente a O_x , o qual sofreu duas rotações e se denota por O_{x_2} .

A matriz correspondente a esta sequência de rotações obtém-se pelo produto:

$$\begin{aligned}
 R_{O_x O_y O_z}(\alpha, \beta, \gamma) &:= R(\gamma, \mathbf{z})R(\beta, \mathbf{y})R(\alpha, \mathbf{x}) \\
 &= \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \\
 &= \begin{bmatrix} \cos \gamma \cos \beta & \cos \gamma \sin \beta \sin \alpha - \sin \gamma \cos \alpha & \cos \gamma \sin \beta \cos \alpha + \sin \gamma \sin \alpha \\ \sin \gamma \cos \beta & \sin \gamma \sin \beta \sin \alpha + \cos \gamma \cos \alpha & \sin \gamma \sin \beta \cos \alpha - \cos \gamma \sin \alpha \\ -\sin \beta & \cos \beta \sin \alpha & \cos \beta \cos \alpha \end{bmatrix}.
 \end{aligned} \tag{3.11}$$

Observe-se que a notação $R_{O_x O_y O_z}$ realça a ordem pela qual as rotações são aplicadas e não a ordem do produto de matrizes.

Salienta-se de novo, a importância da ordem qual as rotações são efectuadas visto que podem ser representadas por diferentes combinações de ângulos de Euler.

Face ao número de decomposições possíveis, várias convenções foram adoptadas, em função do domínio no qual os ângulos são aplicados, das quais se destacam:

- $zyx \rightarrow Roll(O_x), Pitch(O_y)$ e $Yaw(O_z)$

Esta convenção é a mais conhecida, sendo utilizada na aeronáutica. Descreve os

movimentos de um avião (ver Figura 3.5). Os aviões têm três graus de liberdade, i.e., têm três eixos naturais de rotação, perpendiculares dois a dois.

- *Roll* define uma rotação à volta do eixo que corresponde à direcção do avião.
 - *Pitch* induz uma rotação para cima ou para baixo.
 - *Yaw* é uma rotação esquerda-direita ao longo de um eixo vertical.
- *zyz* -> *rotation, precession* e *mutation* - Estes termos são utilizados em Astronomia (*e.g.* na localização da atitude - orientação de um engenho espacial em relação aos três eixos do referencial - de um corpo rígido no espaço) e em Robótica.
 - *zzz* -> Denominada de “*x - convention*”, é muito utilizada na Física.

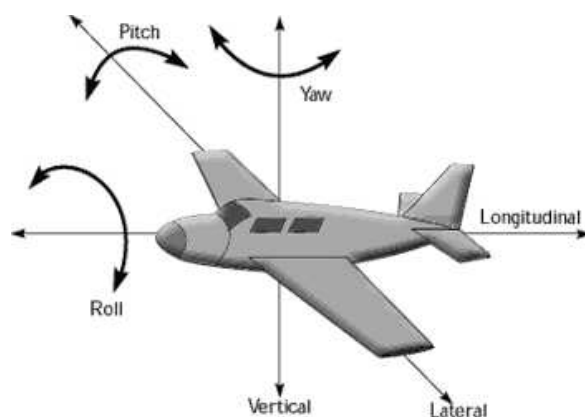


Figura 3.5: Avião com os seus 3 eixos de rotação

Composição de rotações

A composição de rotações não é simples pois não é suficiente adicionar os ângulos uma vez que os eixos também estão submetidos às rotações. A escolha dos eixos e a ordem das rotações têm de ser respeitadas, e existem diversas convenções, como já foi referido. Assim a composição de rotações com ângulos de Euler é problemática porque a ordem da composição determina a matriz final.

Vantagens:

- Os ângulos de Euler são intuitivos e de fácil compreensão pelo que são muito utilizados.
- A representação dos ângulos de Euler é natural e eficiente: utiliza três variáveis (ângulos) para representar três graus de liberdade. Assim, não apresenta restrições a respeitar pelo que não necessita de reajustamentos.
- Forma simples para realizar rotações em torno dos eixos coordenados.

Desvantagens:

Esta representação apresenta vários inconvenientes:

- A maior parte das vezes não existe uma decomposição evidente da rotação geral em três rotações à volta dos eixos. Se, por exemplo, um animador computacional quiser efectuar uma rotação de $\pi/6$ à volta de um eixo de rotação dado pelo vector $(1, 1, 1)$, é difícil determinar os correspondentes ângulos de Euler para os três eixos de base.
- Existem 12 formas diferentes para a composição das rotações elementares. Por exemplo, uma rotação de $(\frac{\pi}{4}, \frac{\pi}{4}, 0)$ corresponde a diferentes orientações em função da convenção $(xyz, yxz, \text{etc.})$ adoptada.
- Nem sempre existe forma de representar uma rotação com ângulos de Euler que corresponda a uma série de rotações concatenadas.
- Uma interpolação suave entre duas orientações envolve integração numérica, o que pode ser computacionalmente caro. A interpolação apresenta ainda dificuldades por existirem várias formas de definir a mesma rotação.
- A principal desvantagem dos ângulos de Euler é a possível perda de um grau de liberdade na realização de rotações à volta dos eixos. Esta singularidade das coordenadas é conhecida por *gimbal lock*.

O Gimbal Lock

Uma vez que uma matriz de rotação final depende da ordem das multiplicações, é possível que por vezes, em certas sequências de rotações, um eixo vá confundir-se com outro.

Esta singularidade é referida como *gimbal lock* por razões históricas.



Figura 3.6: Gimbal

Um *gimbal* é um dispositivo físico que consiste em arcos concêntricos, com pivots que conectam os arcos adjacentes, permitindo-lhes fazer rotações uns dentro dos outros, conforme mostra a Figura 3.6.

Um gimbal com três anéis ligados ortogonalmente corresponde a uma implementação física dos ângulos de Euler, com um eixo em movimento. Os *gimbals* são frequentemente usados para manter giroscópios, em sensores de atitude, na indústria aeronáutica. Uma vez que um giroscópio tem de ficar fixo no espaço, um *gimbal* vai rodar para manter o giroscópio na orientação pretendida.

Vejamos um exemplo. Consideremos o sistema de coordenadas na primeira imagem da Figura 3.7. Após realizar uma rotação de $\pi/6$ em torno do eixo O_x , obtém-se a segunda imagem da Figura. Se agora realizar uma rotação de $\pi/2$ em torno do eixo O_y , resulta na terceira imagem da Figura. Como podemos verificar o eixo O_z actual está alinhado com o eixo O_{x_0} . Ocorreu o *gimbal lock*!. Qualquer rotação em torno do eixo O_z afecta o mesmo grau de liberdade que em torno do eixo O_x . Perdeu-se a capacidade de efectuar rotações em torno de um terceiro grau de liberdade.

O *gimbal lock* é a razão pela qual a indústria aeroespacial foi uma das primeiras a utilizar os quaterniões para representar orientações - satélites, foguetes e aviões -, e não estava

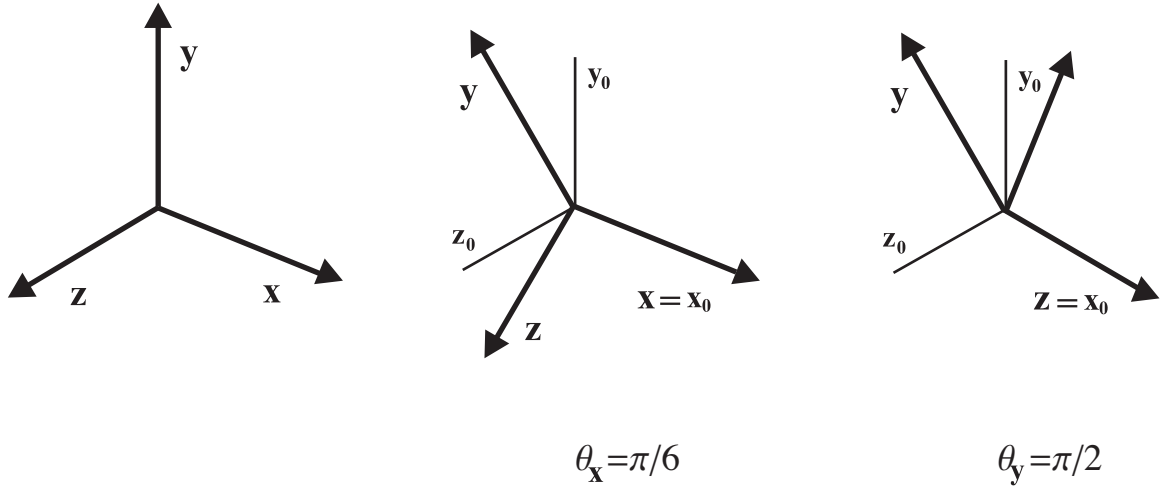


Figura 3.7: Gimbal Lock

nada satisfeita com a possibilidade dos giroscópios bloquearem e provocarem acidentes. Em [Hoatm] explica-se como o Módulo de Excursão à Lua *Apollo* sofreu do *gimbal lock*. Esta situação foi ainda dramatizada no filme *Apollo 13*.

Este problema pode ser exemplificado no modo matricial. Imagine-se, por exemplo, uma sequência de rotações a serem realizadas por um simulador de voo. Especifica-se que a primeira rotação é de ângulo α à volta do eixo O_x , a segunda rotação é de $\frac{\pi}{2}$ em torno do eixo O_y e a terceira de γ em torno do eixo O_z . A rotação em torno do eixo O_y causou um alinhamento entre os eixos O_x e O_z , provocando a perda de um grau de liberdade na rotação, o que matematicamente corresponde à perda de um grau de liberdade na matriz genérica de rotação, dada por

$$\begin{aligned}
 R_{O_x O_y O_z}(\alpha, \pi/2, \gamma) &= R_{O_z}(\gamma)R_{O_y}(\pi/2)R_{O_x}(\alpha) \\
 &= \begin{bmatrix} 0 & \cos \gamma \sin \alpha - \sin \gamma \cos \alpha & \cos \gamma \cos \alpha + \sin \gamma \sin \alpha \\ 0 & \sin \gamma \sin \alpha + \cos \gamma \cos \alpha & \sin \gamma \cos \alpha - \cos \gamma \sin \alpha \\ -1 & \cos \beta \sin \alpha & \cos \beta \cos \alpha \end{bmatrix} \\
 &= \begin{bmatrix} 0 & \sin(\alpha - \gamma) & \cos(\alpha - \gamma) \\ 0 & \cos(\alpha - \gamma) & -\sin(\alpha - \gamma) \\ -1 & 0 & 0 \end{bmatrix}.
 \end{aligned}$$

Esta expressão mostra que a rotação $R(\alpha, \frac{\pi}{2}, \gamma)$ depende somente da diferença $\alpha - \gamma$ e desta forma, ela tem somente um grau de liberdade em vez de dois. Para $\theta = \frac{\pi}{2}$, os ângulos α e γ resultam em rotações à volta do mesmo eixo.

Eixo-Ângulo

Uma rotação pode ser definida de outra forma à custa de um par (\mathbf{n}, θ) , onde \mathbf{n} é um vector unitário que define o eixo de rotação e θ é, como habitualmente, o ângulo de rotação. É a chamada representação Eixo-Ângulo.

Um vector \mathbf{r} que sofre uma rotação de θ graus em torno de um eixo, dado por um vector unitário \mathbf{n} , transforma-se num vector $R\mathbf{r}$, pelo processo que passamos a apresentar. Começamos por escrever o vector \mathbf{r} como a soma de duas componentes, uma componente

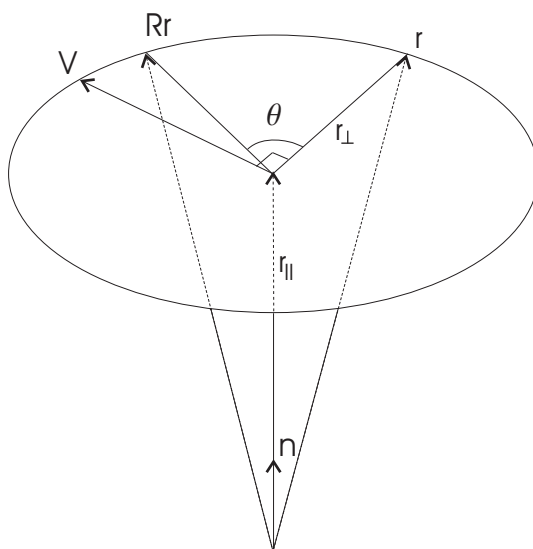


Figura 3.8: Rotação do vector \mathbf{r}

paralela a \mathbf{n} e outra ortogonal a \mathbf{n} , \mathbf{r}_{\parallel} e \mathbf{r}_{\perp} respectivamente, em que:

- $\mathbf{r}_{\parallel} = (\mathbf{r} \cdot \mathbf{n})\mathbf{n}$ - projecção de \mathbf{r} em \mathbf{n} e,
- $\mathbf{r}_{\perp} = \mathbf{r} - \mathbf{r}_{\parallel} = \mathbf{r} - (\mathbf{r} \cdot \mathbf{n})\mathbf{n}$ - componente ortogonal a \mathbf{n}

Tem-se então que:

$$\mathbf{r} = \mathbf{r}_{\parallel} + \mathbf{r}_{\perp} = (\mathbf{r} \cdot \mathbf{n})\mathbf{n} + \mathbf{r} - (\mathbf{r} \cdot \mathbf{n})\mathbf{n}.$$

O vector $R\mathbf{r}$ será obtido adicionando $(R\mathbf{r})_{\parallel}$ e $(R\mathbf{r})_{\perp}$. De notar que $(R\mathbf{r})_{\parallel} = \mathbf{r}_{\parallel}$ (dado que \mathbf{r}_{\parallel} tem a mesma direcção que \mathbf{n}), pelo que:

$$R\mathbf{r} = \mathbf{r}_{\parallel} + (R\mathbf{r})_{\perp}.$$

Para observar a rotação de \mathbf{r} , coloca-se um sistema de coordenadas de duas dimensões no plano ortogonal a \mathbf{n} e que contém os pontos designados por \mathbf{r} e $R\mathbf{r}$.

Para isso, é necessário um vector \mathbf{v} , ortogonal a \mathbf{r}_{\perp} e a \mathbf{n} (ver Figura 3.8),

$$\mathbf{v} = \mathbf{n} \times \mathbf{r}_{\perp} = \mathbf{n} \times (\mathbf{r} - (\mathbf{r} \cdot \mathbf{n})\mathbf{n}) = \mathbf{n} \times \mathbf{r} - \mathbf{n} \times (\mathbf{r} \cdot \mathbf{n})\mathbf{n} = \mathbf{n} \times \mathbf{r} - \vec{0} = \mathbf{n} \times \mathbf{r}.$$

Verifica-se que o componente ortogonal de $R\mathbf{r}$ sobre \mathbf{n} , $(R\mathbf{r})_{\perp}$ pode ser escrito, na base \mathbf{n} , \mathbf{r}_{\perp} , $\mathbf{n} \times \mathbf{r}_{\perp}$ do seguinte modo:

$$(R\mathbf{r})_{\perp} = \mathbf{r}_{\perp} \cos \theta + \mathbf{v} \sin \theta.$$

Finalmente, obtém-se para $R\mathbf{r}$:

$$\begin{aligned} R\mathbf{r} &= (R\mathbf{r})_{\parallel} + (R\mathbf{r})_{\perp} \\ &= \mathbf{r}_{\parallel} + \mathbf{r}_{\perp} \cos \theta + \mathbf{v} \sin \theta \\ &= (\mathbf{r} \cdot \mathbf{n})\mathbf{n} - (\mathbf{r} \cdot \mathbf{n})\mathbf{n} \cos \theta + \mathbf{r} \cos \theta + (\mathbf{n} \times \mathbf{r}) \sin \theta \\ &= (1 - \cos \theta)(\mathbf{r} \cdot \mathbf{n})\mathbf{n} + \mathbf{r} \cos \theta + (\mathbf{n} \times \mathbf{r}) \sin \theta. \end{aligned} \tag{3.12}$$

Além disso, como:

$$\mathbf{n} \times (\mathbf{n} \times \mathbf{r}) = (\mathbf{r} \cdot \mathbf{n})\mathbf{n} - (\mathbf{n} \cdot \mathbf{n})\mathbf{r}$$

e \mathbf{n} é unitário, obtém-se:

$$(\mathbf{r} \cdot \mathbf{n})\mathbf{n} = \mathbf{r} + \mathbf{n} \times (\mathbf{n} \times \mathbf{r}),$$

que ao substituir na equação (3.12), resulta na conhecida fórmula de Rodrigues:

$$R\mathbf{r} = \mathbf{r} + (\mathbf{n} \times \mathbf{r}) \sin \theta + (1 - \cos \theta)\mathbf{n} \times (\mathbf{n} \times \mathbf{r}). \tag{3.13}$$

Composição de rotações

Na representação eixo-ângulo, a composição de rotações é complexa pelo que é necessário realizar uma conversão intermédia para outra representação, sendo computacionalmente caro.

Vantagens

Esta representação apresenta uma interpretação geométrica imediata, o que a torna interessante como forma de visualização intuitiva.

Desvantagens

- A perda da norma unitária do eixo embora seja possível normalizar de novo.
- Erros numéricos podem afectar o valor do ângulo.
- Dificuldade computacional na composição de duas rotações.
- Existe um número infinito de escolhas para o ângulo (múltiplo de 2π) que representam a mesma rotação. Frequentemente, convencionou-se que o eixo tem norma unitária e que o ângulo pertence ao intervalo $[-\pi, \pi]$. Mesmo assim dois pares eixo-ângulo identificam a mesma rotação: $(-\mathbf{n}, -\theta)$ refere-se à mesma rotação que (\mathbf{n}, θ) , o que gera ambiguidade na escolha de orientação do eixo.
- Multiplicidade de representações da identidade. Uma rotação zero à volta de qualquer eixo corresponde sempre à mesma rotação, i.e., à rotação identidade. Por outras palavras, a representação da rotação identidade não é única - de facto, existe um número infinito - o que pode causar graves problemas nos algoritmos quando as rotações aproximam o elemento identidade.

3.3 Rotações com quaterniões

Hamilton procurou descrever rotações no espaço, do mesmo modo que os números complexos descrevem rotações no plano.

Iniciamos esta secção com algumas propriedades que irão permitir demonstrar que, de facto, os quaterniões implementam rotações no espaço.

Proposição 3.1. *Sejam $q \in \mathbb{H}_1$ e $p = p_0 + \mathbf{p} \in \mathbb{H}$. Então:*

$$qpq^{-1} = p',$$

onde $p' \in \mathbb{H}$ é tal que:

$$p' = p_0 + \mathbf{p}' \text{ e } \|\mathbf{p}\| = \|\mathbf{p}'\|.$$

Demonstração. Começamos por provar que:

$$\operatorname{Re}(p') = p_0 = \operatorname{Re}(p).$$

Como $q^{-1} = \bar{q}$ para $q \in \mathbb{H}_1$ e $\operatorname{Re}(p') = \frac{p' + \bar{p}'}{2}$, obtém-se

$$\begin{aligned} 2\operatorname{Re}(p') &= qp\bar{q} + \overline{qp\bar{q}} \\ &= q(p_0 + \mathbf{p})\bar{q} + q(p_0 - \mathbf{p})\bar{q} \\ &= 2p_0. \end{aligned}$$

Acabamos de provar que $p' = p_0 + \mathbf{p}'$. Resta provar que $\|\mathbf{p}\| = \|\mathbf{p}'\|$, ou ainda que $\|p\| = \|p'\|$. Mas,

$$\|p'\| = \|qpq^{-1}\| = \|q\|\|p\|\|q^{-1}\| = \|p\|,$$

o que conclui a demonstração. □

Nota 3.1. *Todo o quaterniões não nulo pode ser escrito na forma αq , com $\alpha \in \mathbb{R} \setminus \{0\}$ e $q \in \mathbb{H}_1$. Como*

$$(\alpha q)p(\alpha q)^{-1} = qpq^{-1} = qp\bar{q},$$

o resultado anterior e todos os que de seguida vamos apresentar podem ser generalizados a \mathbb{H} .

Iremos agora demonstrar o principal resultado desta secção. Iremos estabelecer a correspondência entre o operador qpq^{-1} (ou, quando $q \in \mathbb{H}_1$, $qp\bar{q}$) e a representação de rotações eixo-ângulo, apresentada na Secção 3.2, através do seguinte teorema:

Teorema 3.1. *Seja $q \in \mathbb{H}_1$, $q = \cos \theta + \mathbf{n} \sin \theta$ e seja $p = 0 + \mathbf{p} \in \mathbb{H}_{\mathbb{R}^3}$, $\mathbf{p} \in \mathbb{R}^3$. Então $p' = qpq^{-1}$ corresponde ao vector que resulta da rotação de p em torno do eixo \mathbf{n} por um ângulo 2θ .*

Demonstração. Consideremos $R_q(p) = qpq^{-1} = qp\bar{q}$ em que $p = 0 + \mathbf{p}$ e $q = q_0 + \mathbf{q}$ é um quaternião unitário. Tem-se que:

$$\begin{aligned} R_q(p) &= (q_0 + \mathbf{q})(0 + \mathbf{p})(q_0 - \mathbf{q}) \\ &= (q_0 + \mathbf{q})(\mathbf{p} \cdot \mathbf{q} + q_0\mathbf{p} - \mathbf{p} \times \mathbf{q}) \\ &= (q_0 + \mathbf{q})[\mathbf{q} \cdot \mathbf{p} + (q_0\mathbf{p} - \mathbf{p} \times \mathbf{q})] \\ &= q_0(\mathbf{q} \cdot \mathbf{p}) + q_0(q_0\mathbf{p} - \mathbf{p} \times \mathbf{q}) + \mathbf{q}(\mathbf{q} \cdot \mathbf{p}) + \mathbf{q}(q_0\mathbf{p} - \mathbf{p} \times \mathbf{q}). \end{aligned}$$

De (2.11) resulta,

$$\begin{aligned} R_q(p) &= q_0(\mathbf{q} \cdot \mathbf{p}) + q_0^2\mathbf{p} - q_0(\mathbf{p} \times \mathbf{q}) + \mathbf{q}(\mathbf{q} \cdot \mathbf{p}) - \mathbf{q} \cdot (q_0\mathbf{p} - \mathbf{p} \times \mathbf{q}) + \mathbf{q} \times (q_0\mathbf{p} - \mathbf{p} \times \mathbf{q}) \\ &= q_0(\mathbf{q} \cdot \mathbf{p}) + q_0^2\mathbf{p} - q_0(\mathbf{p} \times \mathbf{q}) + \mathbf{q}(\mathbf{q} \cdot \mathbf{p}) - (\mathbf{q} \cdot \mathbf{p})q_0 + \mathbf{q} \cdot (\mathbf{p} \times \mathbf{q}) + \mathbf{q} \times (q_0\mathbf{p}) - \mathbf{q} \times (\mathbf{p} \times \mathbf{q}) \\ &= q_0^2\mathbf{p} - q_0(\mathbf{p} \times \mathbf{q}) + (\mathbf{q} \cdot \mathbf{p})\mathbf{q} + \mathbf{q} \times (q_0\mathbf{p}) - \mathbf{q} \times (\mathbf{p} \times \mathbf{q}) \\ &= q_0^2\mathbf{p} + (\mathbf{q} \cdot \mathbf{p})\mathbf{q} - \mathbf{q} \times (\mathbf{p} \times \mathbf{q}) - 2q_0(\mathbf{p} \times \mathbf{q}). \end{aligned}$$

Como, para $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \in \mathbb{R}^3$, $\mathbf{v}_1 \times (\mathbf{v}_2 \times \mathbf{v}_3) = (\mathbf{v}_1 \cdot \mathbf{v}_3)\mathbf{v}_2 - (\mathbf{v}_1 \cdot \mathbf{v}_2)\mathbf{v}_3$, obtém-se

$$\begin{aligned} R_q(p) &= q_0^2\mathbf{p} + (\mathbf{q} \cdot \mathbf{p})\mathbf{q} - (\mathbf{q} \cdot \mathbf{q})\mathbf{p} + (\mathbf{q} \cdot \mathbf{p})\mathbf{q} + 2q_0(\mathbf{q} \times \mathbf{p}) \\ &= (q_0^2 - \mathbf{q} \cdot \mathbf{q})\mathbf{p} + 2(\mathbf{q} \cdot \mathbf{p})\mathbf{q} + 2q_0(\mathbf{q} \times \mathbf{p}). \end{aligned}$$

Como já foi visto, o quaternião unitário $q = q_0 + \mathbf{q}$, pode ser reescrito como $q = \cos \theta +$

$\mathbf{n} \sen \theta$, em que $\|\mathbf{n}\| = 1$. Substituindo $q_0 = \cos \theta$ e $\mathbf{q} = \mathbf{n} \sen \theta$ em $R_q(p)$, tem-se:

$$\begin{aligned}
R_q(p) &= (q_0^2 - \mathbf{q} \cdot \mathbf{q})\mathbf{p} + 2(\mathbf{q} \cdot \mathbf{p})\mathbf{q} + 2q_0(\mathbf{q} \times \mathbf{p}) \\
&= (\cos^2 \theta - \mathbf{n} \sen \theta \cdot \mathbf{n} \sen \theta)\mathbf{p} + 2(\mathbf{n} \sen \theta \cdot \mathbf{p})\mathbf{n} \sen \theta + 2 \cos \theta (\mathbf{n} \sen \theta \times \mathbf{p}) \\
&= (\cos^2 \theta - \sen^2 \theta (\mathbf{n} \cdot \mathbf{n}))\mathbf{p} + 2 \sen^2 \theta (\mathbf{n} \cdot \mathbf{p})\mathbf{n} + 2 \cos \theta \sen \theta (\mathbf{n} \times \mathbf{p}) \\
&= (\cos^2 \theta - \sen^2 \theta)\mathbf{p} + 2 \sen^2 \theta (\mathbf{n} \cdot \mathbf{p})\mathbf{n} + 2 \cos \theta \sen \theta (\mathbf{n} \times \mathbf{p}) \\
&= \mathbf{p} \cos(2\theta) + 2 \cdot \frac{1}{2} \cdot (1 - \cos 2\theta)(\mathbf{n} \cdot \mathbf{p})\mathbf{n} + \sen(2\theta)(\mathbf{n} \times \mathbf{p}) \\
&= (1 - \cos 2\theta)(\mathbf{n} \cdot \mathbf{p})\mathbf{n} + \mathbf{p} \cos(2\theta) + (\mathbf{n} \times \mathbf{p}) \sen(2\theta). \tag{3.14}
\end{aligned}$$

Desta derivação, observamos que o resultado obtido é o mesmo vector expresso na representação eixo-ângulo

$$(1 - \cos \theta)(\mathbf{r} \cdot \mathbf{n})\mathbf{n} + \mathbf{r} \cos \theta + (\mathbf{n} \times \mathbf{r}) \sen \theta.$$

pela fórmula de Rodrigues apresentada na Equação (3.12), com excepção do ângulo, pois obteve-se 2θ em vez de θ . \square

Assim, dado um vector unitário \mathbf{n} e um ângulo de rotação θ , o quaternião unitário $q = \cos \theta + \mathbf{n} \sen \theta$ efectua uma rotação de p em torno do eixo \mathbf{n} de ângulo 2θ .

Como consequência importante deste teorema obtemos o seguinte corolário:

Corolário 3.1. *Qualquer rotação de ângulo θ em torno de um eixo \mathbf{n} , com $\|\mathbf{n}\| = 1$, pode ser obtida através de um quaternião unitário.*

Demonstração. No teorema anterior escolhe-se q de modo que $q = \cos \frac{\theta}{2} + \mathbf{n} \sen \frac{\theta}{2}$. Assim, é obtida a rotação desejada. \square

Exemplo 3.3. Consideremos uma rotação de ângulo $\frac{\pi}{2}$, em torno do vector $(0, 1, 0) \in \mathbb{R}^3$. O quaternião que representa esta rotação é:

$$q = \cos \frac{\pi}{4} + \sen \frac{\pi}{4} j = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} j.$$

Equivalência com a representação com matrizes de rotação

Dado o relacionamento entre quaterniões e rotações em \mathbb{R}^3 espera-se que haja uma forma de escrever a representação de rotações com quaterniões em termos matriciais. Como já foi visto na Secção 2.4, a multiplicação de quaterniões tem uma representação matricial. Relembremos que os produtos pq e qp podem ser expressos como $pR(q)$ e $pL(q)$ respectivamente (c.f. com (2.17) e (2.18)), em que,

$$R(q) = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \quad \text{e} \quad L(q) = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix}.$$

Além disso, $R(\bar{q}) = R(q)^T$. Dispondo da forma matricial da multiplicação à esquerda por q e à direita por \bar{q} , podemos, a partir dos quaterniões, gerar uma matriz de rotação associada a uma dada rotação.

Assim, a matriz de rotação corresponde ao produto das duas matrizes, uma vez que, considerando um quaternião puro r ,

$$r' = qr\bar{q} = q(r\bar{q}) = q(rR(q)^T) = (rR(q)^T)L(q) = r(R(q)^T L(q)),$$

em que,

$$R(q)^T L(q) = \begin{bmatrix} q_0^2 + q_1^2 + q_2^2 + q_3^2 & 0 & 0 & 0 \\ 0 & q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 0 & 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 0 & 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

Atendendo ao facto do quaternião q ser unitário, tem-se:

$$R(q)^T L(q) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 0 & 2(q_1q_2 + q_0q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 - q_0q_1) \\ 0 & 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix}.$$

A matriz $R(q)^T L(q)$ é ortogonal uma vez que as duas matrizes $R(q)$ e $L(q)$ são ortogonais quando q é unitário.

Podemos agora estabelecer a expressão da matriz de rotação \mathbf{R} em função de um quaternião unitário q :

$$\mathbf{R} = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix}.$$

Verifica-se que a matriz \mathbf{R} é ortogonal e o seu determinante é 1 pelo que podemos concluir que \mathbf{R} é uma matriz de rotação, estabelecendo deste modo a equivalência. Podemos ainda escrever, na forma de matriz homogénea:

$$R(q)^T L(q) = \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{R} \end{bmatrix}.$$

Nota 3.2. Geralmente o produto de matrizes não comuta, mas estas matrizes comutam uma vez que representam a multiplicação à esquerda e à direita, e a multiplicação de quaterniões é associativa.

Composição de rotações

De modo análogo às matrizes de rotação, multiplicam-se dois quaterniões para somar as rotações que eles representam. Assim, a composição de rotações faz-se naturalmente pela multiplicação dos quaterniões correspondentes. De modo formal tem-se:

Proposição 3.2. *Sejam $p, q \in \mathbb{H}_1$. Uma rotação por p seguida por uma rotação por q é equivalente a uma rotação por qp .*

Demonstração. Seja $r \in \mathbb{H}_{\mathbb{R}^3}$, então

$$\begin{aligned}
 (R_q \circ R_p)(r) &= R_q(R_p(r)) \\
 &= R_q(pr\bar{p}) \\
 &= q(pr\bar{p})\bar{q} \\
 &= (qp)r(\overline{qp}) \\
 &= R_{qp}(r).
 \end{aligned}$$

□

A proposição anterior mostra que a composição de duas rotações é uma rotação, i.e., a rotação correspondente ao quaternião unitário qp . Esta propriedade pode ser generalizada à composição de qualquer número de rotações.

Exemplo 3.4. Uma rotação de ângulo π à volta do eixo O_y , seguida de uma rotação de ângulo π à volta do eixo O_z será representada pelo quaternião $jk = i$.

Podemos concatenar quantos quaterniões desejarmos para produzir um simples quaternião que representa uma série de rotações.

Note-se que o operador de rotação qpq^{-1} corresponde à perspectiva de rotação num sistema de coordenadas móvel enquanto que o produto $q^{-1}pq$ efectua rotações num sistema de coordenadas fixo.

De seguida iremos apresentar a relação entre um quaternião, o seu inverso e o seu simétrico. Consideremos os quaterniões $q \in \mathbb{H}_1$ e $p \in \mathbb{H}_{\mathbb{R}^3}$.

- q e q^{-1}

Geometricamente o inverso q^{-1} de um quaternião q efectua uma rotação com o mesmo ângulo que q , mas o eixo tem sentido contrário (c.f. com a Figura 3.9). Assim, se invertermos o eixo de q^{-1} de modo a fazê-lo coincidir com o eixo de q , veremos que q^{-1} realiza uma rotação contrária à de q , ou seja, o que q faz, q^{-1} desfaz.

Formalizando, tem-se:

$$R_q(p) = qpq^{-1} \quad \text{e} \quad R_{q^{-1}}(p) = q^{-1}p(q^{-1})^{-1} = q^{-1}pq$$

e observe-se que

$$R_{q^{-1}}(R_q(p)) = q^{-1}(qpq^{-1})q = q^{-1}qpq^{-1}q = p.$$

- q e $-q$

Verifica-se que,

$$R_{-q}(p) = (-q)p(-q)^{-1} = qpq^{-1} = R_q(p).$$

Assim, os quaterniões q e $-q$ representam exactamente a mesma rotação, como se pode observar na Figura 3.9. Isto pode ser surpreendente, mas era de se esperar: uma rotação de um ângulo θ em torno de um eixo \mathbf{n} (o quaternião q) pode também ser expressa como uma rotação de um ângulo $-\theta$ em torno do eixo $-\mathbf{n}$ (o quaternião $-q$).

Esta propriedade, chamada *valor dual* ou *dupla cobertura*, representa a correspondência 2:1 dos quaterniões para as rotações.

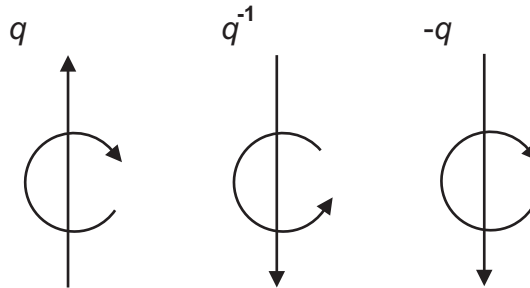


Figura 3.9: Relação entre um quaternião, o seu inverso e o seu simétrico

Vantagens e desvantagens

Vantagens

- É mais simples garantir que um quaternião seja unitário do que preservar a ortogonalidade de uma matriz.

- Os quaterniões unitários estão fortemente relacionados com a notação geometricamente intuitiva da representação eixo-ângulo. Representam uma rotação como um ângulo de rotação e o eixo sobre o qual deve ser feita a rotação, o que é mais natural do que tentar decompor uma rotação em ângulos de Euler, por exemplo.
- A aplicação entre quaterniões e rotações não é ambígua, com excepção de toda a rotação ser representada por dois quaterniões. O facto de os quaterniões q e $-q$ representarem a mesma rotação é um aspecto interessante matematicamente. Isto significa que as rotações vêm aos pares. Dada uma rotação em torno de um eixo e numa direcção qualquer, a mesma rotação pode ser obtida aplicando a rotação em torno do eixo oposto, na direcção oposta (ver Secção 3.3).
- A representação de rotações com quaterniões é compacta, no sentido de que têm quatro graus de liberdade, uma grande vantagem quando se compara às matrizes de rotação que têm nove graus de liberdade.
- Teoricamente todos os quaterniões não nulos podem ser usados para as rotações. Na prática apenas os quaterniões unitários são utilizados. Assim, apenas uma restrição deve ser respeitada em comparação com as seis restrições das matrizes de rotação.
- Quando são necessárias composições de rotações, o uso de quaterniões é muito simples, pois corresponde apenas a multiplicar os dois quaterniões envolvidos.
- Os quaterniões não apresentam o problema do gimbal lock que surge quando utilizamos ângulos de Euler, pois este é um problema inerente à representação matricial usada.
- As rotações realizadas com quaterniões não são influenciadas pela escolha do sistema de coordenadas e o utilizador de um sistema de animação não tem de se preocupar quanto à convenção da ordem das rotações, algo que não é possível quando se usa ângulos de Euler.

- Por fim, e não menos importante, os quaterniões não apresentam problemas na interpolação. Permitem rotações mais suaves e realistas. Os quaterniões permitem realizar a chamada interpolação linear esférica - SLERP (*Spherical Linear intERPolation*).

Desvantagens

- Os quaterniões representam apenas rotações. A parametrização de outras transformações não é adequada ou até mesmo impossível. Pode-se implementar uma translação, considerando o vector translação como um quaternião com parte escalar nula. Em [Mai90], um tipo de quaternião homogéneo é definido com uma multiplicação específica que implementa translações e rotações. Embora seja possível definir quaterniões homogéneos, esta extensão não é tão elegante como as matrizes homogéneas. A extensão homogénea parece ignorada na literatura, sendo os quaterniões usados exclusivamente para representar rotações e para as outras transformações são utilizadas matrizes.
- Os quaterniões são difíceis de visualizar e pouco intuitivos.
- Indeterminação na orientação dos eixos (também ocorre na representação eixo-ângulo).
- Uma outra desvantagem é que a teoria sobre os quaterniões não está incluída na maioria dos currículos actuais de Matemática, criando desta forma a aparência de ser complicada. Embora se possa estudar o grupo dos quaterniões em Álgebra, o conhecimento sobre quaterniões é, em geral, pouco divulgado.

3.4 Análise comparativa das representações

Nesta secção é apresentado um balanço das vantagens e desvantagens das diversas representações de rotações apresentadas assim como uma análise do esforço computacional despendido nas rotações com matrizes e com quaterniões. Confrontando os diversos métodos apresentados e avaliando as vantagens e desvantagens em termos de eficiência, robustez

Métodos	Vantagens	Desvantagens
Matrizes de rotação	<ul style="list-style-type: none"> - A representação matricial para uma rotação é única; - As operações com matrizes são bem conhecidas; - Facilitam os cálculos; - As matrizes homogêneas representam todas as transformações básicas. 	<ul style="list-style-type: none"> - 9 graus de liberdade; - Seis restrições de ortogonalidade; - Risco de erros de arredondamentos na concatenação sucessiva de matrizes; - A matriz de rotação é de difícil construção quando não se conhece a base do espaço em que se realiza a rotação; - A interpolação é problemática.
Ângulos de Euler	<ul style="list-style-type: none"> - 3 graus de liberdade; - Método bastante intuitivo; - Representação natural e eficiente; - Forma simples para rotações em torno dos eixos coordenados. 	<ul style="list-style-type: none"> - Nem sempre existe uma decomposição evidente da rotação em três rotações à volta dos eixos coordenados; - Existem 12 formas diferentes para a composição de rotações elementares; - A representação de rotações concatenadas é bastante complexa; - Possível perda de um grau de liberdade - Gimbal Lock; - Interpolação problemática.
Eixo-Ângulo	<ul style="list-style-type: none"> - 4 graus de liberdade; - Boa visualização. 	<ul style="list-style-type: none"> - Possível perda da norma unitária; - Erros numéricos podem afectar o valor do ângulo; - Dificuldade computacional na composição de rotações; - Ambiguidade na escolha da orientação do eixo; - Multiplicidade de representação de identidade; - Interpolação problemática.
Quaterniões	<ul style="list-style-type: none"> - 4 graus de liberdade; - Simplicidade e economia; - Facilidade na combinação de rotações; - Não são influenciados pela escolha do sistema de coordenadas; - Interpolação perfeita. 	<ul style="list-style-type: none"> - Indeterminação na orientação dos eixos: q e $-q$ representam a mesma rotação; - Representam apenas rotações; - Pouco intuitivos e de difícil visualização.

Tabela 3.1: Quadro resumo das vantagens e desvantagens das representações de rotação

e facilidade de utilização e visualização (ver Tabela 3.1), conclui-se que a implementação dos quaterniões quando comparada com outras alternativas, é simples, barata e bem comportada.

No entanto, a necessidade de realizar conversões entre as diversas representações surge nas implementações computacionais, de modo a tirar partido das vantagens de cada parametrização, sempre com o objectivo de otimizar as transformações a realizar.

Conversão entre representações

Os quaterniões são um método muito eficiente e útil para armazenar e realizar rotações, oferecendo várias vantagens sobre outros métodos. No entanto, algumas API ainda não

têm funções para tratar os quaterniões, devido à sua pouca divulgação. Por exemplo, a API OpenGL, da *Silicon Graphics* não suporta directamente quaterniões, uma rotação é dada por um eixo à volta do qual é efectuada a rotação e por um ângulo, pelo que é necessário converter as orientações definidas com quaterniões para a convenção OpenGL (Eixo-ângulo) e vice-versa. Por outro lado, a API Direct3D especifica rotações com matrizes, pelo que uma conversão quaternião-para-matriz é necessária. Os quaterniões são difíceis de visualizar e nada intuitivos se não estiverem associados à notação eixo-ângulo. Este problema é ultrapassado se os quaterniões forem utilizados para representar rotações internamente, e for usado outro método (por exemplo, eixo-ângulo, ângulos de Euler) como representação externa.

Nesta secção iremos apresentar algoritmos de conversão entre as representações apresentadas e estimar os custos envolvidos, tendo por base o artigo de David Eberly [Ebe04]. As operações contempladas na avaliação dos custos são:

- +: Adição
- \times : Multiplicação
- /: Divisão
- F: Avaliação de uma função existente tal como funções trigonométricas, que por sua vez são frequentemente compostas por adições, multiplicações e divisões
- C: Comparações, que também podem ter um custo computacional significativo.

As conversões e respectivos custos apresentadas nesta secção são:

1. Quaternião para Matriz - $q \mapsto R(\theta, \mathbf{v})$
2. Matriz para Quaternião - $R(\theta, \mathbf{v}) \mapsto q$
3. Quaternião para Ângulos de Euler - $q \mapsto R_{\alpha\beta\gamma}$
4. Ângulos de Euler para Quaternião - $R_{\alpha\beta\gamma} \mapsto q$

5. Quaterniões para Eixo-ângulo - $q \mapsto (\mathbf{n}, \theta)$

6. Eixo-ângulo para Quaterniões - $(\mathbf{n}, \theta) \mapsto q$

1. $\mathbf{q} \mapsto R(\theta, \mathbf{v})$

Consideremos o quaterniões unitário $q = q_0 + q_1i + q_2j + q_3k$ e a matriz de rotação $R(\theta, \mathbf{v})$ definida por (3.10). Usando as identidades $2\sin^2(\theta/2) = 1 - \cos\theta$ e $\sin\theta = 2\sin(\theta/2)\cos(\theta/2)$, é fácil mostrar que:

$$\begin{aligned} 2q_0q_1 &= \sin\theta v_1; & 2q_0q_2 &= \sin\theta v_2; & 2q_0q_3 &= \sin\theta v_3; \\ 2q_1^2 &= (1 - \cos\theta)v_1^2; & 2q_1q_2 &= (1 - \cos\theta)v_1v_2; & 2q_1q_3 &= (1 - \cos\theta)v_1v_3; \\ 2q_2^2 &= (1 - \cos\theta)v_2^2; & 2q_2q_3 &= (1 - \cos\theta)v_2v_3; & 2q_3^2 &= (1 - \cos\theta)v_3^2. \end{aligned}$$

O lado direito destas equações são termos da representação matricial da Fórmula de Rodrigues:

$$I + \sin\theta K(\mathbf{v}) + (1 - \cos\theta)K^2(\mathbf{v}),$$

onde $K(\mathbf{v})$ é a matriz (2.19). Realizando as substituições e simplificando, obtém-se a matriz de rotação:

$$R = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix}.$$

Custo de conversão:

A conversão requer 12 multiplicações para o cálculo dos termos:

$$\begin{aligned} t_{q_1} &= 2q_1; & t_{q_2} &= 2q_2; & t_{q_3} &= 2q_3; \\ t_{q_0q_1} &= q_0t_{q_1}; & t_{q_0q_2} &= q_0t_{q_2}; & t_{q_0q_3} &= q_0t_{q_3}; \\ t_{q_1q_1} &= q_1t_{q_1}; & t_{q_1q_2} &= q_1t_{q_2}; & t_{q_1q_3} &= q_1t_{q_3}; \\ t_{q_2q_2} &= q_2t_{q_2}; & t_{q_2q_3} &= q_2t_{q_3}; & t_{q_3q_3} &= q_3t_{q_3}. \end{aligned}$$

As entradas da matriz de rotação requerem 12 adições. O custo total de operações é $12A + 12M$.

2. $R(\theta, \mathbf{v}) \mapsto \mathbf{q}$

Pretende-se realizar a conversão de uma matriz $R := R(\theta, \mathbf{v})$ para um quaternião $q = q_0 + q_1i + q_2j + q_3k$. Como já foi referido $\cos \theta = (tr(R) - 1)/2$ (cf. com a Equação (3.9)). Utilizando a identidade $2 \cos^2(\frac{\theta}{2}) = 1 + \cos \theta$ verifica-se que $q_0^2 = \cos^2(\frac{\theta}{2}) = (tr(R) + 1)/4$ ou

$$|q_0| = \frac{1}{2} \sqrt{tr(R) + 1}.$$

- Se $tr(R) > 0$

Então $|q_0| > \frac{1}{2}$, e escolhemos, sem perda de generalidade,

$$q_0 = \frac{1}{2} \sqrt{tr(R) + 1}.$$

Observando os elementos r_{ij} da matriz

$$R = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix},$$

conclui-se que

$$r_{23} - r_{32} = 4q_0q_1,$$

$$r_{13} - r_{31} = 4q_0q_2,$$

$$r_{21} - r_{12} = 4q_0q_3$$

e assim,

$$q_1 = \frac{r_{23} - r_{32}}{4q_0}, \quad q_2 = \frac{r_{31} - r_{13}}{4q_0}, \quad q_3 = \frac{r_{12} - r_{21}}{4q_0}.$$

- Se $tr(R) \leq 0$

Então, $|q_0| \leq \frac{1}{2}$, e a expressão anterior pode originar *overflow*, caso q_0 seja muito pequeno. A alternativa é identificar o maior termo q_1, q_2 e q_3 da diagonal principal da matriz de rotação (3.4). Assim, em função do maior termo da diagonal ser:

– r_{11}

Então q_1 é o maior componente em módulo e $4q_1^2 = r_{11} - r_{22} - r_{33} + 1$ i.e.,

$$q_1 = \frac{1}{2}\sqrt{r_{11} - r_{22} - r_{33} + 1}$$

e

$$q_0 = \frac{r_{23} - r_{32}}{4q_1}, \quad q_2 = \frac{r_{12} + r_{21}}{4q_1} \quad e \quad q_3 = \frac{r_{13} + r_{31}}{4q_1};$$

– r_{22}

Tem-se $4q_2^2 = r_{22} - r_{11} - r_{33} + 1$ i.e.,

$$q_2 = \frac{1}{2}\sqrt{r_{22} - r_{11} - r_{33} + 1}$$

e

$$q_0 = \frac{r_{31} - r_{13}}{4q_2}, \quad q_1 = \frac{r_{12} + r_{21}}{4q_2} \quad e \quad q_3 = \frac{r_{23} + r_{32}}{4q_2};$$

– r_{33}

Tem-se $4q_3^2 = r_{33} - r_{11} - r_{22} + 1$ i.e.,

$$q_3 = \frac{1}{2}\sqrt{r_{33} - r_{11} - r_{22} + 1}$$

e

$$q_0 = \frac{r_{12} - r_{21}}{4q_3}, \quad q_1 = \frac{r_{13} + r_{31}}{4q_3} \quad e \quad q_2 = \frac{r_{23} + r_{32}}{4q_3}.$$

Custo de conversão:

A conversão depende do sinal do traço de M . Calcular o traço da matriz requer 2 adições.

Se $tr(R) > 0$ (esta comparação é considerada no custo - 1C), o cálculo $(tr(R) - 1)/2$ envolve 1 adição, 1 divisão e 1 chamada de função. Por sua vez, a expressão $\lambda = 1/4q_0$ envolve 1 multiplicação e 1 divisão. São ainda necessárias 3 adições e 3 multiplicações para calcular os termos $q_1 = \lambda(r_{23} - r_{32})$, $q_2 = \lambda(r_{31} - r_{13})$ e $q_3 = \lambda(r_{12} - r_{21})$. O custo total é de $6A + 4M + 2D + 1F + 1C$.

Se $tr(R) < 0$, o valor máximo dos termos da diagonal da matriz de rotação tem de ser encontrado, o que requer 2 comparações. Suponhamos que r_{11} é o maior valor. Então o

cálculo $q_1 = \sqrt{r_{11} - r_{22} - r_{33} + 1}/2$ envolve 3 adições, 1 divisão e 1 chamada de função. A expressão $\lambda = 1/4q_1$ necessita de 1 multiplicação e 1 divisão. Os termos $q_0 = \lambda(r_{23} - r_{32})$, $q_2 = \lambda(r_{12} - r_{21})$ e $q_3 = \lambda(r_{13} - r_{31})$ requerem 3 adições e 3 multiplicações. O custo total é de $6A + 4M + 2D + 1F + 3C$.

3. $\mathbf{q} \mapsto \mathbf{R}_{\alpha\beta\gamma}$

Esta conversão realiza-se à custa das conversões intermédias, quaternião para matriz e matriz para ângulos de Euler (conversão apresentada por Shoemake em [Sho85], cujo esforço computacional é $1A + 5M + 7F$).

Custo de conversão:

O custo total corresponde à soma dos custos das conversões intermédias, $(12A + 12M) + (1A + 5M + 7F) = 13A + 17M + 7F$.

4. $\mathbf{R}_{\alpha\beta\gamma} \mapsto \mathbf{q}$

A conversão de ângulos de Euler para quaterniões faz-se através da multiplicação de quaterniões. Cada ângulo de Euler é convertido num par eixo-ângulo com o respectivo eixo. Estes três pares são convertidos para quaterniões e multiplicados. O quaternião final é o resultado desejado. Das doze combinações possíveis de ângulos de Euler já referidas, usamos nesta conversão, a convenção *zyx* (*yaw*, *pitch*, *roll*). Lembra-se que uma rotação geral é obtida por uma rotação em torno do eixo O_z por um ângulo γ , seguida de uma rotação em torno do eixo O_y por um ângulo β e por fim de uma rotação em torno do eixo O_x por um ângulo α . Para usar os componentes expressos em termos de quaterniões, obtém-se inicialmente um quaternião para cada rotação simples:

$$q_x = \cos \frac{\alpha}{2} + \text{sen} \frac{\alpha}{2}i; \quad q_y = \cos \frac{\beta}{2} + \text{sen} \frac{\beta}{2}j; \quad q_z = \cos \frac{\gamma}{2} + \text{sen} \frac{\gamma}{2}k.$$

Ao multiplicar estes quaterniões obtém-se o quaternião pretendido, $q = q_z q_y q_x$ com os componentes:

$$\begin{aligned} q_0 &= \cos \frac{\alpha}{2} \cos \frac{\beta}{2} \cos \frac{\gamma}{2} + \operatorname{sen} \frac{\alpha}{2} \operatorname{sen} \frac{\beta}{2} \operatorname{sen} \frac{\gamma}{2}, \\ q_1 &= \operatorname{sen} \frac{\alpha}{2} \cos \frac{\beta}{2} \cos \frac{\gamma}{2} - \cos \frac{\alpha}{2} \operatorname{sen} \frac{\beta}{2} \operatorname{sen} \frac{\gamma}{2}, \\ q_2 &= \cos \frac{\alpha}{2} \operatorname{sen} \frac{\beta}{2} \cos \frac{\gamma}{2} + \operatorname{sen} \frac{\alpha}{2} \cos \frac{\beta}{2} \operatorname{sen} \frac{\gamma}{2}, \\ q_3 &= \cos \frac{\alpha}{2} \cos \frac{\beta}{2} \operatorname{sen} \frac{\gamma}{2} - \operatorname{sen} \frac{\alpha}{2} \operatorname{sen} \frac{\beta}{2} \cos \frac{\gamma}{2}. \end{aligned}$$

Custo de conversão:

A conversão de três pares eixo-ângulo nos respectivos quaterniões é de $3 \times (1M + 1D + 2F)$. Dois produtos de quaterniões requerem $2 \times (16M + 12A)$. O custo total da conversão é de $24A + 35M + 3D + 6F$.

5. $\mathbf{q} \mapsto (\mathbf{n}, \theta)$

Seja $q = q_0 + q_1i + q_2j + q_3k$ um quaternião unitário. Se $|q_0| = 1$, então o ângulo é $\theta = 0$ e qualquer vector director de norma unitária corresponde ao eixo uma vez que não há rotação. Se $|q_0| < 1$, o ângulo obtém-se por $\theta = 2 \operatorname{acos}(q_0)$ e o eixo é calculado por $\mathbf{n} = (q_1, q_2, q_3) / \sqrt{1 - q_0^2}$.

Custo de conversão:

O cálculo de $\theta = 2 \operatorname{acos}(q_0)$ requer 1 chamada de função e 1 multiplicação. Construir $\mathbf{n} = (q_1, q_2, q_3) / \sqrt{1 - q_0^2}$ requer 1 adição, 1 multiplicação, 3 divisões e 1 chamada de função. O custo total de operações é $1A + 2M + 3D + 2F$.

6. $(\mathbf{n}, \theta) \mapsto \mathbf{q}$

Consideremos $\mathbf{n} = (n_1, n_2, n_3)$ um eixo de rotação unitário e θ o ângulo de rotação. O quaternião $q = q_0 + q_1i + q_2j + q_3k$ que representa a rotação satisfaz:

$$q_0 = \cos(\theta/2), \quad q_1 = n_1 \operatorname{sen}(\theta/2), \quad q_2 = n_2 \operatorname{sen}(\theta/2), \quad q_3 = n_3 \operatorname{sen}(\theta/2).$$

Custo de conversão:

Determinar $\theta/2$ usa 1 divisão. Calcular $\operatorname{sen}(\theta/2)$ e $\cos(\theta/2)$ requer 2 chamadas de função.

Por fim o produto $\mathbf{n} \sin(\theta/2)$ utiliza 3 multiplicações. O custo total é $4M + 1D + 2F$.

A tabela seguinte resume os custos associados às conversões entre representações:

Conversões	+	×	/	F	C
$q \mapsto R(\theta, \mathbf{v})$	12	12	0	0	0
$R(\theta, \mathbf{v}) \mapsto q \quad (tr > 0)$	6	4	2	1	1
$R(\theta, \mathbf{v}) \mapsto q \quad (tr \leq 0)$	6	4	2	1	3
$q \mapsto R_{\alpha\beta\gamma}$	13	16	1	7	0
$R_{\alpha\beta\gamma} \mapsto q$	24	35	3	6	0
$q \mapsto (\mathbf{n}, \theta)$	1	2	3	2	0
$(\mathbf{n}, \theta) \mapsto q$	0	3	1	2	0

Tabela 3.2: Número de operações nas conversões entre representações de rotações

Esforço computacional na rotação de vectores

Iremos agora analisar o número de operações envolvidas na rotação de vectores, para as representações Matriz de Rotação e Quaterniões.

A transformação de um vector \mathbf{v} por uma matriz de rotação R corresponde ao produto $\mathbf{u} = R\mathbf{v}$ e requer 6 adições e 9 multiplicações, num total de 15 operações.

Observemos agora uma rotação realizada com quaterniões. Sejam \mathbf{u} e \mathbf{v} quaterniões puros expressos na notação vectorial. Então o quaternião puro $\mathbf{u} = (u_0, u_1, u_2)$ obtido após a rotação é calculado como $\mathbf{u} = q\mathbf{v}\bar{q}$. Aplicando a expressão geral para a multiplicação de quaterniões, o produto $p = q\mathbf{v}$ requer 16 multiplicações e 12 adições (c.f. com (2.6)). O produto $p\bar{q}$ também utiliza o mesmo número de operações. A contagem total de operações é $24A + 32M$ para o *produto geral*.

No entanto, uma vez que a componente real de \mathbf{v} é nula, p necessita apenas de 12 multiplicações e 8 adições. Também sabemos que \mathbf{u} não tem o termo u_0 , pelo que o produto $p\bar{q}$ utiliza 12 multiplicações e 9 adições. Com estas adaptações, contabilizamos um total de $17A + 24M$, para o que designamos de *produto geral adaptado*.

A multiplicação de quaterniões, definida em (2.6), é uma operação “dispendiosa” pois envolve 16 multiplicações internas. Face à importância do produto de quaterniões na avaliação dos custos computacionais, esta operação tem sido objecto de análise no sentido de otimizar o algoritmo, procurando minimizar o número de multiplicações internas envolvidas.

Enuncia-se de seguida um algoritmo mais eficiente, a que chamamos *produto optimizado*, apresentado por Burbanks [Bur96] que envolve apenas 8 multiplicações (à custa de mais adições).

Sejam $p = p_0 + p_1i + p_2j + p_3k$ e $q = q_0 + q_1i + q_2j + q_3k$. Então, definindo

$$t_0 = (p_3 - p_2) * (q_2 - q_3);$$

$$t_1 = (p_0 + p_1) * (q_0 + q_1);$$

$$t_2 = (p_0 - p_1) * (q_2 + q_3);$$

$$t_3 = (p_3 + p_2) * (q_0 - q_1);$$

$$t_4 = (p_3 - p_1) * (q_1 - q_2);$$

$$t_5 = (p_3 + p_1) * (q_1 + q_2);$$

$$t_6 = (p_0 + p_2) * (q_0 - q_3);$$

$$t_7 = (p_0 - p_2) * (q_0 + q_3);$$

$$t_8 = t_5 + t_6 + t_7;$$

$$t_9 = (t_4 + t_8)/2;$$

o produto pq pode ser escrito como:

$$\begin{aligned} pq &= t_0 - t_5 + t_9 \\ &+ (t_1 - t_8 + t_9)i \\ &+ (t_2 - t_7 + t_9)j \\ &+ (t_3 - t_6 + t_9)k. \end{aligned}$$

Facilmente se verifica que o cálculo de cada um dos valores de $t_i; i = 0, \dots, 7$, envolve uma multiplicação e duas adições/subtracções, t_8 envolve duas adições e, finalmente, t_9 requer uma adição e uma divisão. No total, são então necessárias, 8 multiplicações, 19 adições/subtracções e 1 divisão para definir t_0, t_1, \dots, t_9 . O produto pq necessita ainda de mais 8 adições/subtracções, resultando num esforço computacional total de 8 multiplicações, 1 divisão e 27 adições/subtracções.

Com este algoritmo o produto $q\mathbf{v}\bar{q}$ tem um custo de $54A + 18M$. À semelhança do que acontece com a fórmula geral da multiplicação, este algoritmo pode ainda ser adaptado para a rotação de vectores com quaterniões uma vez as operações que envolvem a componente real dos quaterniões não necessitam de ser realizadas e contabilizadas. O primeiro produto $p = q\mathbf{v}$ requer 9 multiplicações e 23 adições, e o produto $p\bar{q}$ necessita de 8 multiplicações e 23 adições, num custo total de $46A + 17M$. Referimo-nos a este produto como *produto optimizado adaptado*.

Por outro lado, observe-se ainda que a conversão de um quaternião q para uma matriz de rotação \mathbf{R} requer 12 multiplicações e 12 adições. Transformar \mathbf{v} por \mathbf{R} utiliza 15 operações ($6A + 9M$). Assim, o processo de converter um quaternião numa matriz e realizar a rotação com a matriz utiliza 39 operações ($18A + 21M$), duas menos do que calcular $q\mathbf{v}\bar{q}$, pela fórmula genérica adaptada. Esta análise resume-se na Tabela 3.3.

Em relação à composição de rotações, observa-se que o produto de duas matrizes de rotação requer 27 multiplicações e 18 adições num custo total de $18A + 27M$.

O produto de dois quaterniões requer 16 multiplicações e 12 adições num custo total de $12A + 16M$, sendo claramente mais eficiente. Desta maneira, ganha-se em eficiência com-

Rotação de n vectores	+	\times	Observações
Matrizes de rotação	$6n$	$9n$	
Quaterniões	$24n$	$32n$	Produto geral
Quaterniões	$17n$	$24n$	Produto geral adaptado
Quaterniões	$54n$	$18n$	Produto otimizado
Quaterniões	$46n$	$17n$	Produto otimizado adaptado
Quaterniões e matrizes	$12 + 6n$	$12 + 9n$	Convertendo primeiro para uma matrix 3×3

Tabela 3.3: Comparação do número de operações envolvidas na rotação de n vectores

putacional usando quaterniões em situações em que muitas rotações devem ser aplicadas a um objecto.

Além disso, a normalização é outro processo que deve ser considerado na avaliação de custos: normalizar um quaternião para ajustar os erros de vírgula flutuante é mais barato do que ortonormalizar uma matriz de rotação com o processo de ortonormalização de Gram-Schmidt.

Composição de rotações	+	\times
Matriz de rotação	18	27
Quaternião	12	16

Tabela 3.4: Número de operações envolvidas na composição de duas rotações

Verifica-se neste conjunto de operações standard que os quaterniões apresentam uma reduzida complexidade na composição de um grande número de rotações. Para as restantes operações, continua a ser mais rápido converter o quaternião resultante para uma matriz 3×3 , antes de realizar outras operações adicionais.

3.5 Considerações adicionais

O estudo que aqui apresentamos das rotações e do uso de quaterniões para representar rotações é, necessariamente, incompleto. O objectivo deste capítulo era apresentar uma área de aplicação importante dos quaterniões que fosse reconhecida tanto por matemáticos como por engenheiros e outros cientistas. A nível da computação gráfica, a utilização de quaterniões para representar orientações é, actualmente, uma realidade. Muitos detalhes foram omitidos, em particular, a interpolação de orientações que assume uma importância vital na animação gráfica. Muitas referências podem ser apontadas como importantes nestas área. Pelo seu carácter inovador e pedagógico, salientamos o livro recente de Hanson, *Visualizing Quaternions* [Han06], onde é feito um tratamento interessante e bastante completo deste tema, tanto do ponto de vista matemático como do ponto de vista computacional.

Capítulo 4

Cálculo numérico e simbólico com quaterniões

Um dos objectivos principais deste projecto consistiu no desenvolvimento de ferramentas simples e amigáveis para operar com quaterniões. Neste sentido, foi codificado, no sistema MATLAB, um conjunto de funções de manipulação de quaterniões e implementação de rotações.

Neste capítulo são apresentados os contributos computacionais do trabalho realizado: a calculadora - QuatCalc - e um pacote de rotinas para rotações.

4.1 Software para Quaterniões

O interesse pelo cálculo numérico e/ou simbólico com quaterniões tem vindo a aumentar desde os anos sessenta devido às aplicações práticas em diversos domínios, já identificadas. Nesta secção fazemos referência ao estado da arte no que, à implementação de quaterniões, diz respeito.

Começamos por apresentar alguns exemplos (considerados mais relevantes e completos para este trabalho) de pacotes disponíveis em três sistemas computacionais.

MATLAB

- *Aerospace Blockset*

Descrição: Conjunto de rotinas para o *Aerospace Blockset*.

Sítio: Simulink

Autor: The MathWorks, Inc

Ano: 1994-2006

- *Quaternion data class*

Descrição: Implementa um conjunto de funções para quaterniões.

Sítio: <http://www.mathworks.com>

Autor: Bill Davidson

Ano: 2006

- *Quaternion Toolbox*

Descrição: Coleção de funções que operam sobre quaterniões.

Sítio: <https://sourceforge.net/projects/qtfm/>

Autor: Steve Sangwine e Nicolas Le Bihan

Ano: 2005

Maple

- *Quat Package*

Descrição: Package de Quaterniões

Sítio: Disponível em CD-ROM, incluído no livro recente *Funktionentheorie in der Ebene und im Raum Reihe: Grundstudium Mathematik* dos autores K. Gürlebeck, K. Habetha e W. Sprössig, editado por *Ein Birkhauser Buch* [GHS06].

Autor: Sebastian Bock

Ano: 2006

- *Quaterman*

Descrição: Biblioteca de cálculo e de manipulação sobre a álgebra dos Quaterniões

Sítio: <http://www.inria.fr/rrrt/rt-0148.html>

Autores: Patrick Capolsini, Stephane Dalmas e Yves Papegay

Ano: 1993

- *QuaternionAlgebra*

Descrição: Pacote que torna possível a manipulação simbólica de quaterniões.

Sítio: <http://privatewww.essex.ac.uk/~sjs/>

[quaternionalgebra/quaternionalgebra.html](http://privatewww.essex.ac.uk/~sjs/quaternionalgebra/quaternionalgebra.html)

Autores: Steve Sangwine

Ano: 2005

Mathematica

- *Algebra‘Quaternions‘*

Sítio: Mathematica

Autor: Wolfram Research

- *Quaternions*

Descrição: Estende o pacote Algebra‘Quaternions‘ às Rotações

Sítio: <http://computing.ee.ethz.ch/sepp/mathematica-5.0.1-rs.SEPP/>

[scratch/AddOns/StandardPackages/Algebra/Quaternions.m](http://computing.ee.ethz.ch/sepp/mathematica-5.0.1-rs.SEPP/scratch/AddOns/StandardPackages/Algebra/Quaternions.m)

Contexto: Algebra‘Quaternions‘

Autor: Jason Kastner

Ano: 2000

- *Quaternica*

Descrição: Pacote para a manipulação de expressões que envolvem Quaterniões

Sítio: <http://library.wolfram.com/infocenter/Articles/1405/>

Autor: Stephane Dalmas e Yves Papegay

Ano: 1995

Em seguida, apresentamos algumas das rotinas identificadas através de pesquisas via Internet:

- *Qcalc*

Descrição: Calculadora de quaterniões desenvolvida em Java

Sítio: www.theworld.com/~sweetser/java/qcalc/qcalc.html

Autor: Doug Sweetser

Ano: 1998

- *Quaternion Arithmetic, Functions and Calculus*

Descrição: Conjunto de rotinas JAVA implementando operações sobre quaterniões e que permite realizar conversões entre representações de rotações.

Sítio: www.euclideanspace.com/math/

Autor: Martin John Baker

Ano: 1998-2006

- *Quaternion Class*

Descrição: Conjunto de rotinas C++ implementando operações sobre quaterniões

Sítio: http://celestia.teyssier.org/source-documentation/quaternion_8h.html

Autor: Chris Laurel

- *Visualizing Quaternions*

Descrição: “Quaternion Survival Kit” - Package de Quaterniões incluído no livro recente *Visualizing Quaternions* do autor Andrew J. Hanson editado por *Morgan-Kaufmann/Elsevier*, 2006.

Sítio: <http://books.elsevier.com/companions/0120884003/vq/index.html>

Autor: Andrew Hanson

Ano: 2006

- *Calc 3D*

Descrição: Pacote de software com operações algébricas

Sítio: <http://www.greuer.de/pcalc3d.html>

Autor: Andreas Greuer

- *Panda3D API: LQuaternionf*

Descrição: Classe quaternião, desenvolvida em C++, para a biblioteca Panda3D.

Sítio: <http://panda3d.org/apiref.php?page=LQuaternionf>

Autor: Panda 3D Software Copyright (c) 2001 - 2004, Disney Enterprises, Inc.

4.2 O MATLAB

Um dos objectivos deste trabalho foi desenvolver software para tratamento numérico e/ou simbólico de quaterniões. Embora existam, tal como já foi referido, vários pacotes disponíveis, pretendíamos desenvolver uma calculadora de manipulação simples à qual está evidentemente associado um conjunto de rotinas que podem ser utilizadas independentemente. Para este trabalho foi utilizada uma linguagem de programação que disponibiliza um ambiente de desenvolvimento baseado sobre um interpretador especializado no cálculo numérico matricial - o MATLAB. A nossa opção pelo sistema MATLAB teve várias motivações:

- É muito conhecido e amplamente divulgado;
- Permite o desenvolvimento de interfaces gráficas através da *tool* GUIDE e dá acesso a ferramentas de visualização fáceis de utilizar;
- É de fácil aprendizagem, em comparação outras linguagens de programação (e.g., C++, JAVA);
- Disponibiliza muitas funções predefinidas;
- É uma linguagem Orientada aos Objectos, de fiabilidade reconhecida;
- Possui um compilador que permite a geração de programas *stand-alone* executáveis em computadores sem o MATLAB instalado;
- Disponibiliza um Web Server que possibilita aceder na Internet a aplicações criadas no MATLAB.

4.3 A classe @quaternion

Em termos de implementação optou-se por uma abordagem *Object Oriented* criando uma classe @quaternion para a definição de um quaternião.

Na programação orientada aos objectos, uma classe pode ser vista como o encapsulamento de um determinado tipo de objectos (neste caso, os objectos são os quaterniões).

Uma classe é constituída por:

- um construtor;
- um conjunto de métodos (ou operações) sobre os objectos da classe.

O construtor implementa as regras internas que definem o quaternião. Pode ser visto como uma função à qual é passado um conjunto de parâmetros devolvendo um objecto da classe em causa.

Cada objecto da classe herda todos os métodos e características da classe a que pertence.

Para a classe @quaternion, o construtor tem dois parâmetros:

- Quaternião - Expressão que representa o quaternião;
- Modo - Indica a forma (\mathbb{H} ou \mathbb{R}^4) do quaternião passado ao construtor.

Esta estrutura permite a utilização da classe @quaternion a partir da linha de comandos do MATLAB.

De seguida serão intercalados como exemplos, algumas operações realizadas na *prompt* do MATLAB. Começamos pelo exemplo do construtor de quaterniões no modo \mathbb{H} e no modo \mathbb{R}^4 :

Exemplo 4.1. O quaternião $q = 1 + \sqrt{2}i - 3/5j - 8k$ pode ser definido por:

```
>> q = quaternion('1 + sqrt(2)i -3/5j - 8k', 'H')
```

```
q = 1.0000    1.4142   -0.6000   -8.0000
```

```
>> q = quaternion([1, sqrt(2), -3/5, -8])
```

```
q = 1.0000    1.4142   -0.6000   -8.0000
```

Os métodos agrupam as funções quaterniônicas elementares, trigonométricas e hiperbólicas.

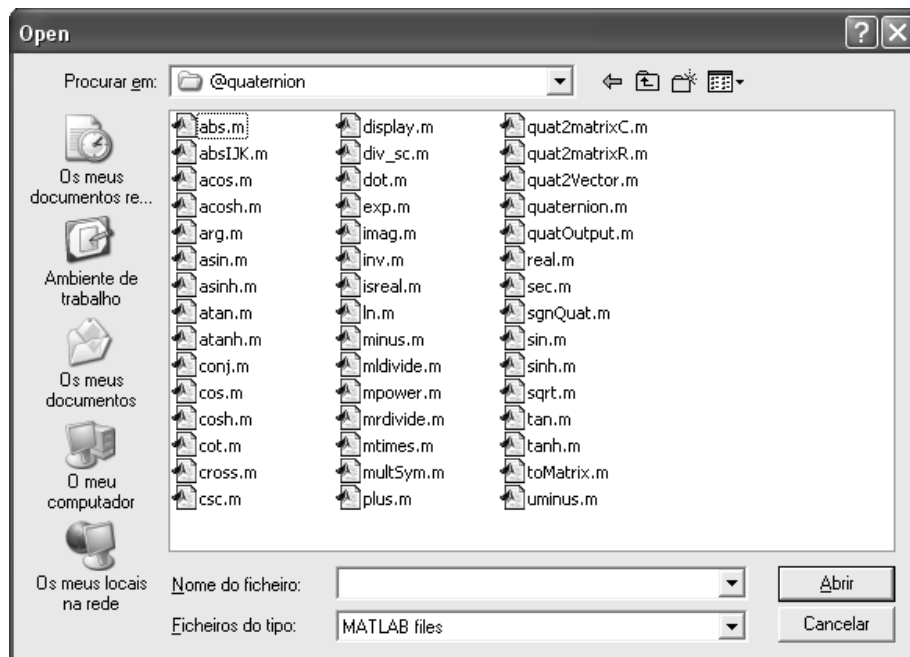


Figura 4.1: Construtor e métodos da classe @quaternion

Exemplo 4.2.

```
>> q = quaternion([3, 1/5, sqrt(7), -3])
```

```
q = 3.0000    0.2000    2.6458   -3.0000
```

```
>> p = conj(q)
```

```
p = 3.0000   -0.2000   -2.6458    3.0000
```

Qualquer classe definida herda um conjunto de funções básicas do próprio MATLAB (e.g. adição, multiplicação, etc). No entanto, para que essas operações possam ser usadas, é necessário defini-las, ou redefini-las, no universo da classe - a adição base do MATLAB funciona para números (objectos) inteiros, reais e ainda para matrizes, mas não funcionará para os objectos da classe visto que não está definido o que significa adicionar quaterniões.

Por outras palavras, é necessário definir ou implementar o comportamento das várias operações como a adição e multiplicação para cada classe. A esta redefinição de operações (ou métodos) dá-se o nome de *overloading*.

Na classe @quaternion foram redefinidas funções existentes no MATLAB, de modo a operar correctamente com quaterniões. Por outro lado também foram criados novos operadores. Alguns métodos podem ainda ser substituídos pelos símbolos, como: plus $\mapsto +$; minus $\mapsto -$; uminus $\mapsto -$; mtimes $\mapsto *$; mldivide $\mapsto \backslash$; mrdivide $\mapsto /$ e mpower $\mapsto \wedge$.

Exemplo 4.3.

```
>> q = quaternion('1 + 2k', 'H')
q = 1      0      0      2

>> p = quaternion('5 -2i + 3k', 'H')
p = 5      -2      0      3

>> p*q
ans = -1      -2      4      13

>> q*p
ans = -1      -2      -4      13

>> q/p
ans = 0.2895      0.0526      0.1053      0.1842

>> q\p
ans = 0.2895      0.0526      -0.1053      0.1842

>> q^p
ans = -1.0617      -0.3285      -0.4520      1.6227
```


Na classe @quaternion foram também incluídos métodos para representação de quaterniões na forma matricial real e complexa:

Exemplo 4.4.

- `quat2matrixC` - Representação com matrizes complexas 2×2 ;

```
>> q = quaternion([1, 2, 3, 4]);
>> MC = quat2matrixC(q)
MC = 1.0000 + 2.0000i    3.0000 + 4.0000i
      -3.0000 + 4.0000i    1.0000 - 2.0000i
```

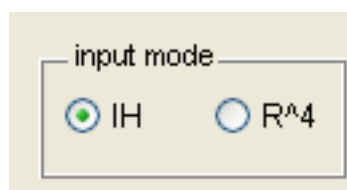
- `quat2matrixR` - Representação com matrizes reais 4×4 ;

```
>> MR = quat2matrixR(q)
MR = 1    2    3    4
      -2   1   -4   3
      -3   4    1  -2
      -4  -3    2    1
```

4.4 A calculadora QuatCalc

QuatCalc é uma calculadora que permite realizar operações com quaterniões. Possibilita a introdução de quaterniões de dois modos diferentes:

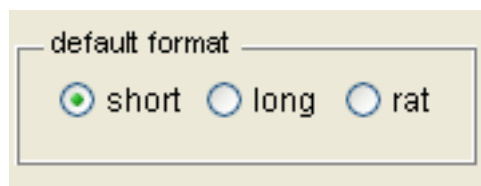
- Modo \mathbb{H} : $q = q_0 + q_1i + q_2j + q_3k$
- Modo \mathbb{R}^4 : $q = [q_0, q_1, q_2, q_3]$



Após escolher o modo de entrada e o formato pretendido, o utilizador introduz o quaterniões no campo de entrada e selecciona, com o rato, a operação a realizar. Caso a expressão introduzida não seja válida, a calculadora assume o quaterniões nulo. Por defeito, a calculadora assume \mathbb{H} como modo de entrada e o formato *short*.

Os operandos e seus resultados são apresentados numa *listbox*, separados respectivamente por uma linha tracejada ou uma linha contínua. Os dados da *listbox* podem ser reutilizados através de um duplo *click*, numa linha da *listbox*.

A calculadora QuatCalc suporta operações nos formatos *long*, *short* e *rational*.



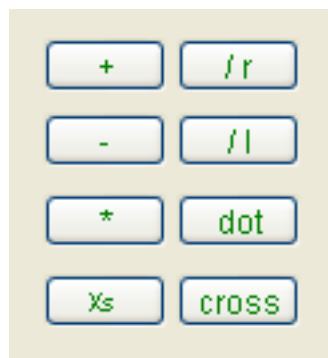
O último resultado obtido após uma operação pode ainda ser convertido num formato diferente do seleccionado previamente, primindo no *Output Format*.

Operações Suportadas

A QuatCalc está organizada em três conjuntos de operações:

- Operações Elementares com dois Operandos

- “ + ” - Adição
- “ - ” - Subtracção
- “ * ” - Multiplicação
- “ \times_s ” - Produto simétrico
- “ /r ” - Divisão à direita
- “ \l ” - Divisão à esquerda
- “ dot ” - Produto interno
- “ cross ” - Produto externo



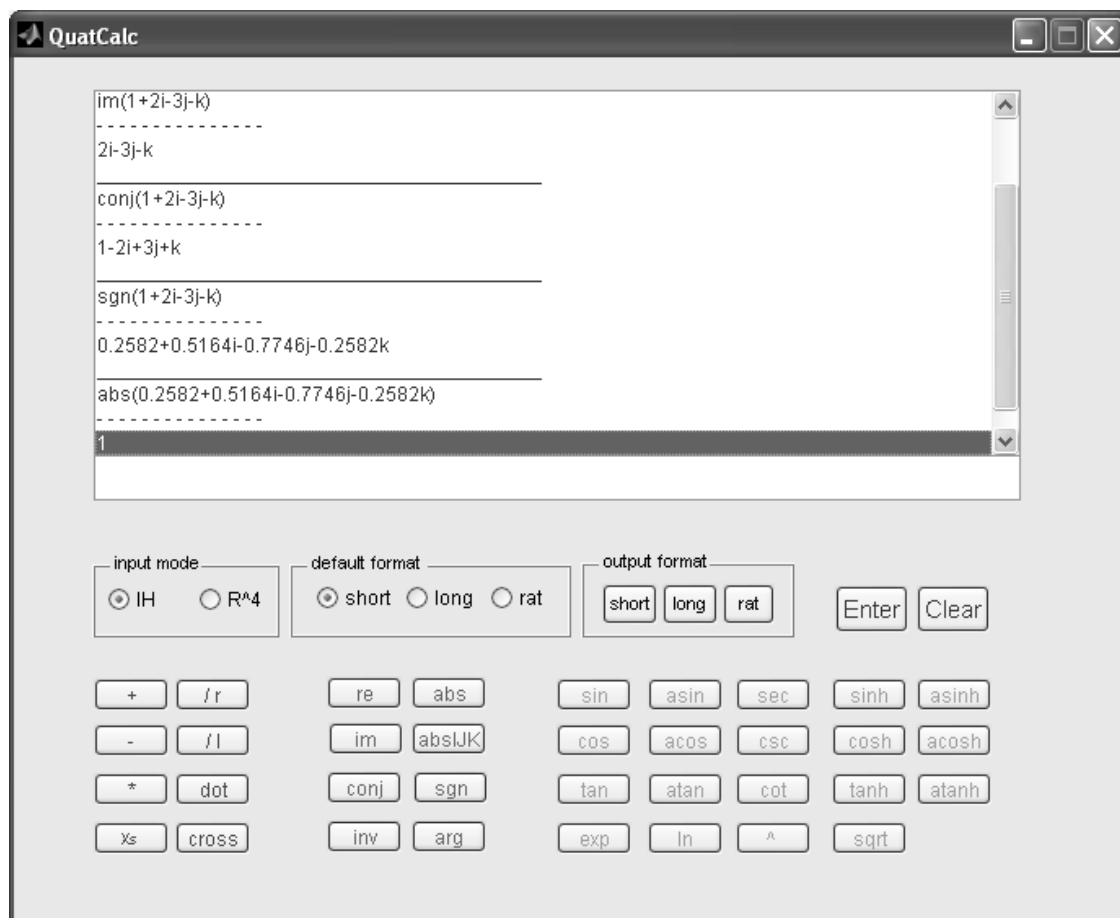


Figura 4.2: A calculadora QuatCalc

- Operações Elementares com um Operando

- “re” - Parte Real
- “im” - Parte Imaginária
- “conj” - Conjugado
- “inv” - Inverso
- “abs” - Norma (ou Valor Absoluto)



- “absIJK” - Norma da parte imaginária
- “sgn” - Sinal
- “arg” - Argumento
- Funções Trigonométricas, Hiperbólicas e Exponenciais
 - “sin” - Seno
 - “cos” - Coseno
 - “tan” - Tangente
 - “asin” - Arco Seno
 - “acos” - Arco Coseno
 - “atan” - Arco Tangente
 - “sec” - Secante
 - “csc” - Co-secante
 - “cot” - Co-tangente
 - “sinh” - Seno Hiperbólico
 - “cosh” - Coseno Hiperbólico
 - “tanh” - Tangente Hiperbólica
 - “asinh” - Arco Seno Hiperbólico
 - “acosh” - Arco Coseno Hiperbólico
 - “atanh” - Arco Tangente Hiperbólico
 - “exp” - Exponencial
 - “ln” - Logaritmo
 - “^” - Potência
 - “sqrt” - Raiz Quadrada

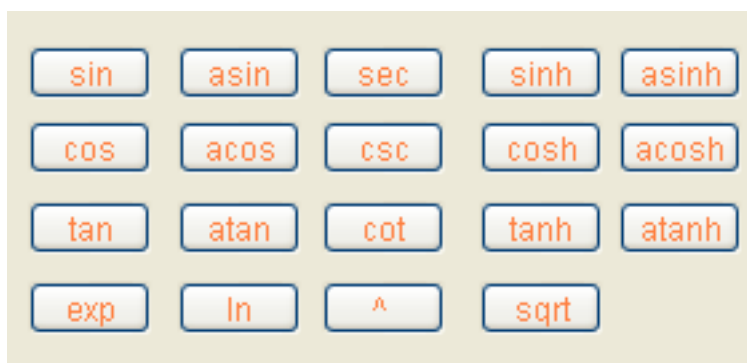


Figura 4.3: Funções trigonométricas, hiperbólicas e exponenciais

4.5 Rotinas para rotações 3D

Adicionalmente à classe `@quaternion`, foi também implementado um conjunto de rotinas que implementam a rotação de um vector \mathbf{v} por um ângulo θ , em torno de um eixo \mathbf{n} . O modo de utilização destas funções está acessível invocando o `help` na linha de comandos do MATLAB. Salientamos que estas funções não estão na classe `@quaternion`, uma vez que nem todas operam sobre quaterniões. Assim, um quaternião deve ser definido como um vector de 4 componentes, sempre que é parâmetro de entrada de alguma destas funções.

De seguida ilustramos, através de um exemplo, cada uma das rotinas implementadas:

Exemplo 4.5.

- `rotByMatrix` - Rotação usando Matriz de Rotação

```
>> [v, R] = rotByMatrix([1,0,0], pi, [0,1,0])
v = 0 -1.0000 0.0000
R = 1.0000 0 0
    0 -1.0000 -0.0000
    0 0.0000 -1.0000
```

- `rotByAxisAngle` - Rotação usando Eixo-Ângulo

```
>> v = rotByAxisAngle([1,0,0], pi, [0,1,0])
v = 0 -1.0000 0.0000
```

- rotByQuat - Rotação usando Quaternião

```
>> [v, q] = rotByQuat([1,0,0], pi, [0,1,0])
v = 0 -1 0
q = 0.0000 1.0000 0 0
```

Conversão entre representações de rotações

Tal como já foi referido, a conversão entre representações é essencial para tirar partido das diferentes vantagens inerentes a cada uma das representações. Nesse sentido, foi também desenvolvido um conjunto de rotinas que efectua a conversão de rotações entre quaterniões e matriz de rotação, eixo-ângulo e ângulos de Euler. Mais uma vez, apresentamos cada uma das rotinas através de um exemplo.

Exemplo 4.6.

- quat2matrix - Conversão de quaternião para matriz de rotação

```
>> q=[1,1,1,1];
>> R=quat2matrix(q)
R = 0.0000 -0.0000 1.0000
     1.0000 0.0000 -0.0000
    -0.0000 1.0000 0.0000
```

Note-se que esta função, embora aceite como argumento, um quaternião não unitário, devolve a matriz associada ao quaternião introduzido normalizado.

- matrix2quat - Conversão de matriz de rotação para quaternião

```
>> p=matrix2quat(R)
p =  0.5000    0.5000    0.5000    0.5000
```

Como se pode comprovar, o quaternião p obtido corresponde ao quaternião inicial q normalizado.

- `axisAngle2quat` - Conversão de eixo-ângulo para quaternião

```
>> q = axisAngle2quat([1 0 0], pi/4)
q =  0.9239    0.3827    0    0
```

- `quat2axisAngle` - Conversão de quaternião para eixo-ângulo

```
>> [n, theta] = quat2axisAngle(q)
n =  1.0000    0    0
theta =  0.7854
```

- `euler2quat` - Conversão de ângulos de Euler para quaternião

```
>> q = euler2quat(pi/2,pi/2,pi/2)
q =  0.7071    0.0000    0.7071    0.0000
```

4.6 QuatRotation - Visualização de rotações 3D

O `QuatRotation` foi o programa desenvolvido no âmbito deste estudo e que, recorrendo a alguns dos métodos apresentados acima, permite a visualização da rotação de objectos no espaço tridimensional.

O utilizador pode seleccionar os objectos disponíveis através de um *popup-menu*:

- Ponto
- Tetraedro

- Hexaedro
- Octaedro
- Icosaedro
- Dodecaedro

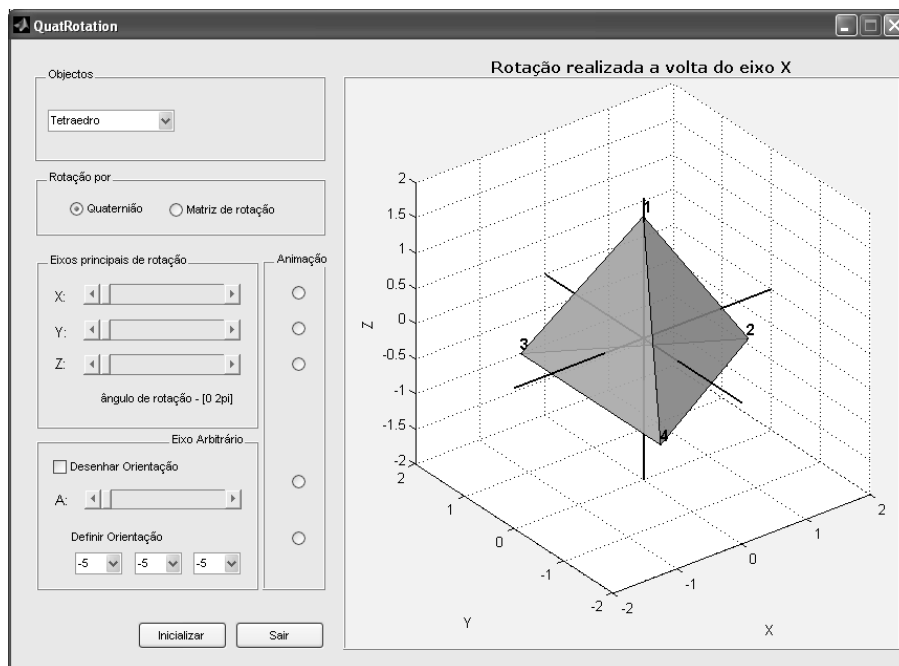


Figura 4.4: Rotação de Poliedros em \mathbb{R}^3

Sempre que o utilizador selecciona um dos objectos, o programa actualiza a figura no referencial.

O programa disponibiliza duas formas de rotação dos objectos:

- Rotação Manual - neste modo o utilizador usa um *slider* que lhe permite especificar o ângulo de rotação no intervalo de $[0, 2\pi]$;
- Rotação Dinâmica - neste modo o objecto roda ininterruptamente à volta de um eixo. A rotação dinâmica é implementada através do incremento sucessivo de um ângulo fixo.

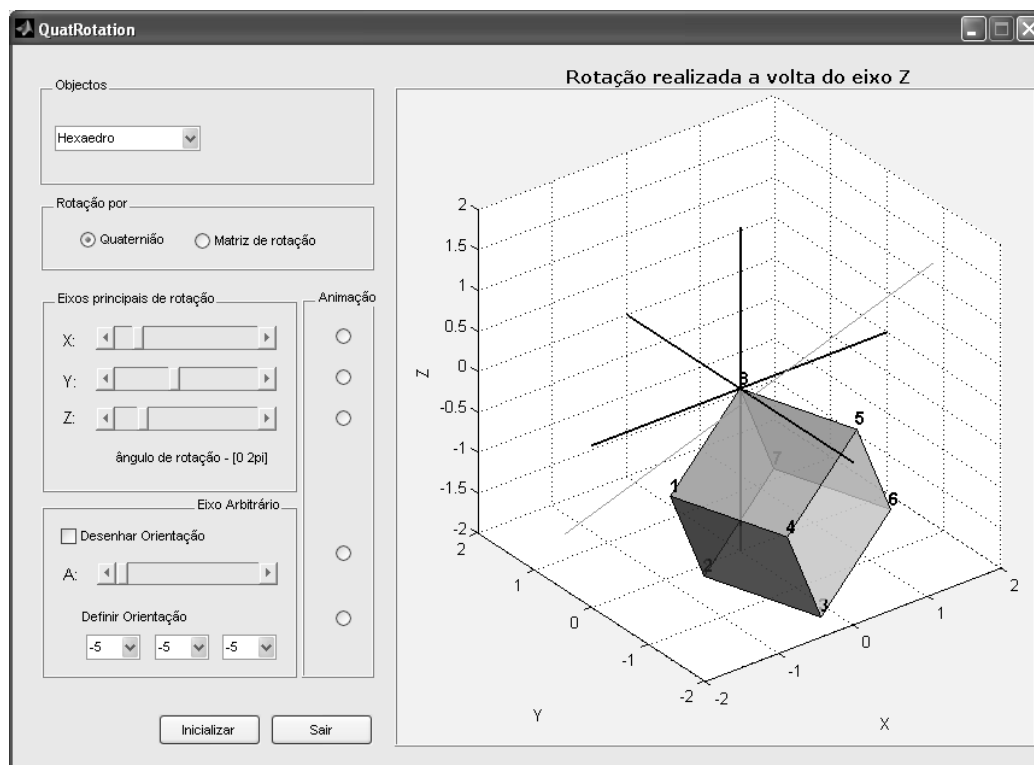


Figura 4.5: Rotação manual ou dinâmica de um hexaedro

A rotação de cada objecto pode ser realizada à volta de um dos três eixos principais (X, Y, Z) ou à volta de um eixo arbitrário. A orientação do eixo arbitrário pode ser desenhada graficamente pelo utilizador ou definida manualmente.

A calculadora QuatCalc, o QuatRotation e todas as funções em MATLAB desenvolvidas no âmbito deste trabalho são do domínio público e podem ser encontradas em <http://www.math.uminho.pt/~mmc/dissertacoes/quaternions.htm>.

Bibliografia

- [Alt89] Simon L. Altmann. Hamilton, Rodrigues and the Quaternion Scandal. *Mathematics Magazine*, Vol 62,5,pp. 291-308, 1989.
- [Bai96] Éric Bainville. *Modélisation géométrique et dynamique d'un geste chirurgical*. PhD thesis, Université Joseph Fourier - Grenoble I, Março 1996.
- [BDS82] F. Brackx, R. Delanghe e F. Sommen. *Clifford analysis*. 76. Pitman, Boston-London-Melbourne, 1982.
- [Bur96] Andrew Burbanks. *Quaternions in C++*. 1996.
- [Cay45] Arthur Cayley. On Algebraical Couples. *Phyl. Magazine XXVII*, pages 38-40, 1845.
- [CDP93] P. Capolsini, S. Dalmas e Y. Papegay. Quaterman: A library for computing with quaternions in Maple. Technical report, INRIA-I3S - Université de Nice Sophia Antipolis, 148, 1993.
- [CS03] John H. Conway e Derek A. Smith. *On Quaternions and Octonions: their geometry, arithmetic and symmetry*. A. K. Peters Ltd, 2003.
- [DKL98] Erik B. Dam, Martin Koch e Martin Lillholm. *Quaternions, Interpolation and Animation*. Technical report, Department of Computer Science, University of Copenhagen, Denmark, 1998.

- [Dor95] Fadi Dornaïka. *Contribution à l'intégration vision/robotique : calibrage, localisation et asservissement*. Phd thesis, Institut national polytechnique de Grenoble, 1995.
- [Ebe04] David Eberly. *Rotation Representations and Performance Issues*. Technical report, Geometric Tools, Inc., 2004.
- [Fue28] R. Fueter. Über Funktionen einer Quaternionenvariablen. *Atti Congr. Int. Mat., Bologna*, 1928.
- [Fue32] R. Fueter. *Analytische Funktionen einer Quaternionenvariablen*. *Comment. Math. Helv.*, 9–20, 1932.
- [Fue35] R. Fueter. Die Funktionentheorie der Differentialgleichungen $\Delta u = 0$ und $\Delta\Delta u = 0$ mit vier reellen Variablen. *Comm. Math. Helv.*, (7):307–330, 1935.
- [GHS06] K. Gürlebeck, K. Habetha e W. Sprössig. *Funktionentheorie in der Ebene und im Raum*. Grundstudium Mathematik, Birkhäuser, 2006.
- [Gra48] James Graves. *Transaction of the Irish Academy* 21. 338, 1848.
- [GS97] Klaus Gurlebeck e Wolfgang Sprossig. *Quaternionic and Clifford Calculus for Physicists and Engineers*. John Wiler & Sons, 1997.
- [Ham53] William Rowan Hamilton. *Lectures on Quaternions*. 1853.
- [Ham67] William Rowan Hamilton. *The Mathematical Papers of Sir William Rowan Hamilton*. 1967.
- [Ham69] William Rowan Hamilton. *Elements of Quaternions*. 1969.
- [Han01] Andrew J. Hanson. *Visualizing Quaternions*. Siggraph Tutorial, 2001.
- [Han06] Andrew J. Hanson. *Visualizing Quaternions*. The Morgan Kaufmann Series in Interactive 3D Technology. Elsevier, 2006.

- [Hoatm] David Hoag. Apollo Guidance and Navigation Considerations of Apollo Inertial Measurement Unit (IMU) Gimbal Lock. Mit instrumentation laboratory document e-1344, Massachusetts Institute of Tecnology, 1963, webpage: www.hq.nasa.gov/office/pao/History/alsj/e-1344.htm.
- [Kui02] Jack B. Kuipers. *Quaternions and Rotation Sequences*. Princeton University Press, 41 William Street, Princeton, New Jersey 08540, 2002.
- [Lan98] Jeff Lander. Better 3D: The writing Is on the Wall. *Game Developer*, 1998.
- [Len02] Eric Lengyel. *Mathematics for 3D Game Programming and Computer Graphics*. Charles River Media, 2002.
- [LS87] F.S Leite e J. Saraiva. Generalization of the De Moivre Formulas for Quaternions and Octonions. Estudos de matemática in the honour of Luís de Albuquerque, Universidade de Coimbra, 1987.
- [Mai90] Patrick-Gilles Maillot. *Using quaternions for coding 3d transformations*. Graphics Gems (Andrew Glassner, edt.), Academic Press, Inc., Boston, 1990.
- [Mal03] Helmuth R. Malonek. Quaternions in applied sciences - a historical perspective of a mathematical concept. *in: K. Gürlebeck, L. Hempel, C. Könke (Editors): Proceedings 16th International Conference on the Applications of Computer Science and Mathematics in Architecture and Civil Engineering*, 2003.
- [McCtm] Michael McCool. Orientation Interaction Tester, 1995, webpage: www.cgl.uwaterloo.ca/gallery/image-html/gimbal.jpg.htm.
- [Muk02] R. Mukundan. Quaternions: From Classical Mechanics to Computer Graphics, and Beyond. 2002.
- [Pau80] Wolfgang Pauli. *General Principles of Quantum Mechanics*. Berlin: Springer-Verlag, 1980.

- [PD95] Y. Papegay e S. Dalmas. Quaternica: A package for manipulating expressions involving quaternions. Technical report, Computacional mechanics publications, Mathematics with vision: proceedings of the First Internacional Mathematica Symposium, 291-298, 1995.
- [Rod40] Olinde Rodrigues. Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements consideres independamment des causes qui peuvent les produire. *Journal de Mathématiques Pures et Appliquées*, 5:380–440, 1840.
- [SD92] Ken Shoemake e Tom Duff. Matrix animation and polar decomposition. pages 258–264, 1992.
- [Sho85] Ken Shoemake. Animating Rotation with Quaternion Curves. In “*Proceedings of SIGGRAPH 1985*, volume 19, pages 245–254. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, July 22–26 1985.
- [Sho92] Ken Shoemake. ARCBALL: A User Interface for Specifying Three-Dimensional Orientation Using a Mouse. In “*Proceedings of Graphics Interface '92*, pages 151–156, Vancouver, Canada, 1992.
- [SN93] M.D. Shuster e G.A. Natanson. Quaternion computation from a geometric point of view. *Journal of Astronautical Sciences*, (41(4)):545–556, 1993.
- [Sud79] A. Sudbery. Quaternionic analysis. *Math. Proc. Camb. Phil. Soc.*, (85):197–225, 1979.
- [TA05] Joseph Teran e John Anderson. Visualize whirled peas: Efficient simulation of many colliding deformable solids. 2005.
- [Vic01] Leandra Vicci. Quaternions and Rotations in 3-space: The algebra and its geometric interpretation. Technical Report TR01-014, 27 2001.