

FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA" - CENTRO
UNIVERSITÁRIO "EURÍPIDES DE MARÍLIA"
UNIVEM
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO.

SILAS MARQUES RIBEIRO

**PARAMETRIZAÇÃO E ANIMAÇÃO DE ROTAÇÕES NO
ESPAÇO POR QUATÉRNIOS**

MARÍLIA

2005

SILAS MARQUES RIBEIRO

PARAMETRIZAÇÃO E ANIMAÇÃO DE ROTAÇÕES NO ESPAÇO
POR QUATÉRNIOS

Monografia apresentada ao Conselho do
Curso de Bacharelado em Ciência da Com-
putação do Centro Universitário Eurípides
de Marília, como requisito para a obtenção
do grau de Bacharel em Computação

Orientador:

Prof. Dr. Luiz Fernandes Galante

MARÍLIA

2005

SILAS MARQUES RIBEIRO

**PARAMETRIZAÇÃO E ANIMAÇÃO DE ROTAÇÕES NO ESPAÇO
POR QUATÉRNIOS**

Banca examinadora da monografia apresentada ao Conselho do Curso de Bacharelado em Computação do UNIVEM/F.E.S.R., como requisito para a obtenção do grau de Bacharel em Computação.

Resultado: _____

ORIENTADOR : Prof. Dr. Luiz Fernandes Galante:

1º EXAMINADOR : Prof. Dr. José Remo Ferreira Brega

2º EXAMINADOR : Prof. Dr. Antonio Carlos Sementille

Marília , 30 de Novembro de 2005

RIBEIRO, Silas Marques. **Parametrização e Animação de Rotações no Espaço por Quatérnios**. 2005, 46 f. Trabalho de Conclusão de Curso - Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2005.

RESUMO

Neste trabalho apresentamos um estudo da parametrização do espaço das rotações 3D pelo conjunto dos quatérnios, com ênfase às vantagens que esta representação tem em relação a outras mais populares como ângulo-eixo, ângulos de Euler e por matrizes. Um estudo comparativo baseado em critérios de custo computacional, operacionalidade e qualidade gráfica de animações obtidas por interpolações, bem como algumas implementações de procedimentos utilizados são apresentados.

Palavras-chave: rotação, quatérnios, interpolação, animação

RIBEIRO, Silas Marques. **Parametrização e Animação de Rotações no Espaço por Quatérnios. 2005, 46 f.** Trabalho de Conclusão de Curso - Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2005.

ABSTRACT

In this work we presented a study about rotations parametrization by quaternions, with emphasis to the advantages that this representation has in relationship the others more used as angle-axis, Euler angles and matrices. A comparative study based on criteria of computations cost , functionality and graphic quality of animations obtained by interpolation, as well as some implementation of used procedures are presented.

Keywords: rotations, quaternions, interpolation, animation

SUMÁRIO

	Página
RESUMO	ii
ABSTRACT	iii
INTRODUÇÃO	1
1 QUATÉRNIOS	2
1.1 Operações Básicas	2
1.2 Propriedades Estruturais	4
1.3 Conjugados, Norma e Módulo.	5
1.4 Unitários e Forma Polar.	7
2 ROTAÇÕES 3D	9
2.1 Preliminares	9
2.2 Expressão Vetorial de $\mathbf{R}(\theta, \hat{\mathbf{u}})$ - Fórmula de Rodrigues	9
2.3 Linearidade e Representação Matricial de $\mathbf{R}(\theta, \hat{\mathbf{u}})$	10
2.4 Rotações em torno dos Eixos Coordenados	12
2.5 Parametrização por Ângulos de Euler	13
2.6 Parametrização de Rotações por Quatérnios	14
2.7 Matriz de uma Rotação parametrizada por Quatérnio	17
3 ANIMAÇÃO E INTERPOLAÇÃO DE ROTAÇÕES	20
3.1 Preliminares.	20
3.2 Interpolação Linear de Rotações representadas por Matrizes	20
3.3 Interpolação Linear de Rotações representadas por Ângulos de Euler.	23
3.4 Interpolação Linear de Rotações representadas por Quatérnios.	25
3.5 Interpolação Linear Esférica de Rotações representadas por Quatérnios.	26

CONCLUSÕES 28

A APÊNDICE

Implementações 32

A.1	Adição de Quatérnios: $Ad(q,p)$	32
A.2	Multiplicação de Escalar por Quatérnio : $Me(s,q)$	32
A.3	Multiplicação de Quatérnios - Forma vetorial : $M(p,q)$	32
A.4	Multiplicação de Quatérnios - Forma matricial: $M^*(p,q)$	33
A.5	Conjugação de Quatérnios: $Conj(q)$	33
A.6	Norma: $Nor(q)$	33
A.7	Módulo de um quatérnio: $Mod(q)$	33
A.8	Inverso de um Quatérnio: $Inv(q)$	34
A.9	Construindo o unitário de um Quatérnio	34
A.10	Forma Polar do Unitário de um Quatérnio: $Fpl(q)$)	34

B APÊNDICE

Implementações 35

B.1	Interpolação Linear de Matrizes - (Fig 1.4)	35
B.2	Interpolação Linear de Matrizes - (Fig 2.4)	35
B.3	Interpolação Linear de ângulos de Euler - (Fig 3.4)	36
B.4	Interpolação Linear de ângulos de Euler - (Fig 4.4)	36
B.5	Interpolação Linear de Quatérnios - (Fig 5.4)	37
B.6	Interpolação Esférica de Quatérnios - Fig(6.4)	38

REFERÊNCIAS 39

INTRODUÇÃO

Neste trabalho estudamos a parametrização de rotações no espaço \mathbb{R}^3 por quatérnios e as facilidades que esta representação introduz nas interpolações relacionadas à geração de animações por computador. Para efeito de comparação, apresentamos também as representações mais utilizadas de rotações 3D, a saber: por ângulo-eixo, por matrizes e por ângulos de Euler.

Na produção de animações, a representação por quatérnios destaca-se das demais, qualitativamente pela uniformidade com que gera quadros interpolados e, quantitativamente pelo número reduzido de operações exigidas, que se traduz em baixo custo computacional.

A escolha do tema se justifica pela importância que a transformação rotação tem em várias áreas do conhecimento, como, por exemplo, navegação aérea, visão computacional, robótica, mecânica clássica, fotogrametria e computação gráfica.

A fundamentação deste trabalho está nos artigos de (Shoemake, 1985), (Shoemake, 1991), (Erick et ali., 1998) e os livros de (Gomes et ali., 1998), (Verth et ali., 2004).

Na seção 1 construímos a álgebra dos quatérnios, destacando suas propriedades estruturais. Na seção 2 discutimos rotações 3D. Partindo da expressão eixo - ângulo, conhecida como Fórmula de Rodrigues, obtivemos as representações matricial e quatérnica. A representação por ângulos de Euler e algumas conversões também são discutidas nesta seção. Na seção 3 discutimos os métodos de interpolação e os problemas apresentados por cada uma das representações utilizadas na animação resultante.

Finalmente, apresentamos as conclusões. Nos apêndices, apresentamos os procedimentos em linguagem Maple, que implementam as principais operações com quatérnios e algumas das animações apresentadas nas figuras do texto.

1 QUATÉRNIOS

1.1 Operações Básicas

No espaço \mathbb{R}^4 , são bem conhecidas as operações **adição** e **multiplicação por escalar**. Se $\mathbf{q}_1 = (w_1, x_1, y_1, z_1)$, $\mathbf{q}_2 = (w_2, x_2, y_2, z_2)$ são elementos do \mathbb{R}^4 e $\alpha \in \mathbb{R}$, então:

$$\begin{aligned}\mathbf{q}_1 + \mathbf{q}_2 &= (w_1 + w_2, x_1 + x_2, y_1 + y_2, z_1 + z_2), \\ \alpha \mathbf{q}_1 &= (\alpha w_1, \alpha x_1, \alpha y_1, \alpha z_1).\end{aligned}$$

No que segue, vamos definir uma **multiplicação** no espaço \mathbb{R}^4 . Para tanto, consideremos a base canônica $B = \{\mathbf{1}, \mathbf{i}, \mathbf{j}, \mathbf{k}\}$, onde $\mathbf{1} = (1, 0, 0, 0)$, $\mathbf{i} = (0, 1, 0, 0)$, $\mathbf{j} = (0, 0, 1, 0)$, e $\mathbf{k} = (0, 0, 0, 1)$. Nesta base, os vetores \mathbf{q}_1 e \mathbf{q}_2 podem ser escritos na forma:

$$\begin{aligned}\mathbf{q}_1 &= w_1 \mathbf{1} + x_1 \mathbf{i} + y_1 \mathbf{j} + z_1 \mathbf{k} \\ \mathbf{q}_2 &= w_2 \mathbf{1} + x_2 \mathbf{i} + y_2 \mathbf{j} + z_2 \mathbf{k}\end{aligned}\tag{*}$$

Definimos o produto dos elementos desta base pela tabela:

\cdot	$\mathbf{1}$	\mathbf{i}	\mathbf{j}	\mathbf{k}
$\mathbf{1}$	$\mathbf{1}$	\mathbf{i}	\mathbf{j}	\mathbf{k}
\mathbf{i}	\mathbf{i}	-1	\mathbf{k}	$-\mathbf{j}$
\mathbf{j}	\mathbf{j}	$-\mathbf{k}$	-1	\mathbf{i}
\mathbf{k}	\mathbf{k}	\mathbf{j}	\mathbf{i}	-1

Usando a representação (*) obtém-se:

$$\begin{aligned}
\mathbf{q}_1\mathbf{q}_2 &= (w_1\mathbf{1} + x_1\mathbf{i} + y_1\mathbf{j} + z_1\mathbf{k})(w_2\mathbf{1} + x_2\mathbf{i} + y_2\mathbf{j} + z_2\mathbf{k}) = \\
&= w_1w_2 + w_1x_2\mathbf{i} + w_1y_2\mathbf{j} + w_1z_2\mathbf{k} \\
&+ x_1w_2\mathbf{i} + x_1x_2\mathbf{ii} + x_1y_2\mathbf{ij} + x_1z_2\mathbf{ik} \\
&+ y_1w_2\mathbf{j} + y_1x_2\mathbf{ji} + y_1y_2\mathbf{jj} + y_1z_2\mathbf{jk} \\
&+ z_1w_2\mathbf{k} + z_1x_2\mathbf{ki} + z_1y_2\mathbf{kj} + z_1z_2\mathbf{kk} \\
&= w_1w_2 - (x_1x_2 + y_1y_2 + z_1z_2) + w_1(x_1\mathbf{i} + y_1\mathbf{j} + z_1\mathbf{k}) + w_2(x_2\mathbf{i} + y_2\mathbf{j} + z_2\mathbf{k}) \\
&+ (y_1z_2 - y_2z_1)\mathbf{i} + (z_1x_2 - z_2x_1)\mathbf{j} + (x_1y_2 - x_2y_1)\mathbf{k}
\end{aligned}$$

Identificando $\mathbf{v}_1 = x_1\mathbf{i} + y_1\mathbf{j} + z_1\mathbf{k}$, $\mathbf{v}_2 = x_2\mathbf{i} + y_2\mathbf{j} + z_2\mathbf{k}$ com vetores do \mathbb{R}^3 , temos:

$$\begin{aligned}
\mathbf{v}_1 \cdot \mathbf{v}_2 &= x_1x_2 + y_1y_2 + z_1z_2 \quad (\text{produto escalar}), \\
\mathbf{v}_1 \wedge \mathbf{v}_2 &= (y_1z_2 - y_2z_1)\mathbf{i} + (z_1x_2 - z_2x_1)\mathbf{j} + (x_1y_2 - x_2y_1)\mathbf{k} \quad (\text{produto vetorial}),
\end{aligned}$$

e então:

$$\mathbf{q}_1\mathbf{q}_2 = w_1w_2 - \mathbf{v}_1 \cdot \mathbf{v}_2 + w_1\mathbf{v}_1 + w_2\mathbf{v}_2 + \mathbf{v}_1 \wedge \mathbf{v}_2$$

Definição. O conjunto \mathbb{R}^4 munido das três operações:

$$\begin{aligned}
\mathbf{q}_1 + \mathbf{q}_2 &= (w_1 + w_2, x_1 + x_2, y_1 + y_2, z_1 + z_2), \\
\alpha \cdot \mathbf{q}_1 &= (\alpha w_1, \alpha x_1, \alpha y_1, \alpha z_1), \\
\mathbf{q}_1\mathbf{q}_2 &= (w_1w_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, w_1\mathbf{v}_1 + w_2\mathbf{v}_2 + \mathbf{v}_1 \wedge \mathbf{v}_2),
\end{aligned}$$

onde $\mathbf{q}_1 = w_1\mathbf{1} + x_1\mathbf{i} + y_1\mathbf{j} + z_1\mathbf{k}$, $\mathbf{q}_2 = w_2\mathbf{1} + x_2\mathbf{i} + y_2\mathbf{j} + z_2\mathbf{k}$, $\mathbf{v}_1 = x_1\mathbf{i} + y_1\mathbf{j} + z_1\mathbf{k}$, e $\mathbf{v}_2 = x_2\mathbf{i} + y_2\mathbf{j} + z_2\mathbf{k}$ será chamado **Conjunto dos Quatérnios** e será indicado por **H**.

Definição. Dado $\mathbf{q} = (w, x, y, z) = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \in \mathbf{H}$, a primeira componente w é chamada **parte real de q**, e o vetor $\mathbf{v} = (x, y, z) = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ é chamada **parte imaginária de q**.

É usual representar um quatérnio $\mathbf{q} = (w, x, y, z) = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ na forma

compacta $[w, \mathbf{v}]$, onde w é a **parte real** e $\mathbf{v} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ é a **parte imaginária**.

Se

$$\begin{aligned}\mathbf{q}_1 &= w_1\mathbf{1} + x_1\mathbf{i} + y_1\mathbf{j} + z_1\mathbf{k} \\ \mathbf{q}_2 &= w_2\mathbf{1} + x_2\mathbf{i} + y_2\mathbf{j} + z_2\mathbf{k}, \text{ e} \\ \alpha &\in \mathbb{R},\end{aligned}$$

então as formas compactas da **soma**, **múltiplo** e **produto** de \mathbf{q}_1 por \mathbf{q}_2 são definidas por:

$$\begin{aligned}\mathbf{q}_1 + \mathbf{q}_2 &= [w_1 + w_2, \mathbf{v}_1 + \mathbf{v}_2] \\ \alpha\mathbf{q}_1 &= [\alpha w_1, \alpha\mathbf{v}_1] \\ \mathbf{q}_1\mathbf{q}_2 &= [w_1w_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, w_1\mathbf{v}_1 + w_2\mathbf{v}_2 + \mathbf{v}_1 \wedge \mathbf{v}_2]\end{aligned}$$

1.2 Propriedades Estruturais

Definição.

$$\begin{aligned}\mathbf{0} &\stackrel{def}{=} [0, 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}], \text{ é chamado } \mathbf{quatérnio nulo} \\ \mathbf{1} &\stackrel{def}{=} [1, 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}], \text{ é chamado } \mathbf{quatérnio unidade} \\ -\mathbf{q}_1 &\stackrel{def}{=} [-w_1, -\mathbf{v}_1], \text{ é chamado } \mathbf{oposto} \text{ de } \mathbf{q}_1 = [w_1, \mathbf{v}_1] \\ \mathbf{q}_1^{-1} &\stackrel{def}{=} \left[\frac{w_1}{\|\mathbf{q}_1\|}, \frac{\mathbf{v}_1}{\|\mathbf{q}_1\|} \right], \text{ é chamado } \mathbf{inverso} \text{ de } \mathbf{q}_1, \text{ onde} \\ &\|\mathbf{q}_1\| = w_1^2 + x_1^2 + y_1^2 + z_1^2, \text{ e } \mathbf{q}_1^{-1} = w_1\mathbf{1} + x_1\mathbf{i} + y_1\mathbf{j} + z_1\mathbf{k} \in \mathbf{H} \text{ é não nulo.}\end{aligned}$$

Proposição. valem as propriedades abaixo para as operações de \mathbf{H} .

Para quaisquer que sejam $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3 \in \mathbf{H}$, têm-se:

$$\begin{aligned}
(A_1) \quad \mathbf{q}_1 + (\mathbf{q}_2 + \mathbf{q}_3) &= (\mathbf{q}_1 + \mathbf{q}_2) + \mathbf{q}_3, & (\text{associativa da adi\c{c}ao}) \\
(A_2) \quad \mathbf{q}_1 + \mathbf{q}_2 &= \mathbf{q}_2 + \mathbf{q}_1, & (\text{comutativa da adi\c{c}ao}) \\
(A_3) \quad \mathbf{0} + \mathbf{q}_1 &= \mathbf{q}_1 + \mathbf{0} = \mathbf{q}_1, & (\text{Neutro Aditivo}) \\
(A_4) \quad \mathbf{q}_1 + -\mathbf{q}_1 &= -\mathbf{q}_1 + \mathbf{q}_1 = \mathbf{0}, & (\text{Exist\ec{e}ncia do Oposto}) \\
(M_1) \quad \mathbf{q}_1(\mathbf{q}_2\mathbf{q}_3) &= (\mathbf{q}_1\mathbf{q}_2)\mathbf{q}_3, & (\text{Assoc. da Multiplica\c{c}ao}) \\
(M_2) \quad \mathbf{q}_1\mathbf{1} &= \mathbf{1}\mathbf{q}_1 = \mathbf{q}_1, & (\text{Neutro multiplicativo}) \\
(M_3) \quad \mathbf{q}_1\mathbf{q}_1^{-1} &= \mathbf{q}_1^{-1}\mathbf{q}_1 = \mathbf{1} & (\text{Exist\ec{e}ncia do Inverso para } \mathbf{q}_1 \neq \mathbf{0}) \\
(D_1) \quad \mathbf{q}_1(\mathbf{q}_2 + \mathbf{q}_3) &= \mathbf{q}_1\mathbf{q}_2 + \mathbf{q}_1\mathbf{q}_3 & (\text{Distributiva \`a direita}) \\
(D_2) \quad (\mathbf{q}_1 + \mathbf{q}_2)\mathbf{q}_3 &= \mathbf{q}_1\mathbf{q}_3 + \mathbf{q}_2\mathbf{q}_3 & (\text{Distributiva \`a esquerda})
\end{aligned}$$

Observa\c{c}o. A multiplica\c{c}o de quat\ec{e}rnios n\~ao \u00e9 **comutativa**, tomando-se por exemplo $\mathbf{q}_1 = [2, \mathbf{i} + \mathbf{j} + \mathbf{k}]$, $\mathbf{q}_2 = [3, 2\mathbf{i} + \mathbf{j} - \mathbf{k}]$, tem-se $\mathbf{q}_1\mathbf{q}_2 = [4, 5\mathbf{i} + 2\mathbf{j} + 0\mathbf{k}]$ e $\mathbf{q}_2\mathbf{q}_1 = [4, 9\mathbf{i} + 2\mathbf{j} + 2\mathbf{k}]$

1.3 Conjugados, Norma e M\u00f3dulo.

Defini\c{c}o. O **conjugado** de um quat\ec{e}rnio $\mathbf{q} = [w, \mathbf{v}] = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ \u00e9 o quat\ec{e}rnio $\mathbf{q}^* \stackrel{def}{=} [w, -\mathbf{v}] = w - x\mathbf{i} - y\mathbf{j} - z\mathbf{k}$, obtido de \mathbf{q} negativando-se a parte imagin\u00e1ria.

Defini\c{c}o. A **norma** de um quat\ec{e}rnio $\mathbf{q} = [w, \mathbf{v}] = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$, indicado por $\|\mathbf{q}\|$ ou **Nor**(\mathbf{q}), \u00e9 o n\u00famero real n\u00e3o negativo definido por :

$$\|\mathbf{q}\| \stackrel{def}{=} \mathbf{q}\mathbf{q}^* = w^2 + x^2 + y^2 + z^2.$$

Definição. O **módulo** de $\mathbf{q} = [w, \mathbf{v}] = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ é o número real não negativo:

$$|\mathbf{q}| \stackrel{def}{=} \sqrt{\|\mathbf{q}\|} = \sqrt{\mathbf{q}\mathbf{q}^*} = \sqrt{w^2 + x^2 + y^2 + z^2}$$

Proposição. A conjugação de quatérnios tem as seguintes propriedades:

- (c₁) $(\mathbf{q}^*)^* = \mathbf{q}$ (o conjugado do conjugado é o próprio quatérnio)
- produto** (c₂) $(\mathbf{qp})^* = \mathbf{p}^*\mathbf{q}^*$ (o conjugado do produto é o dos conjugados na ordem inversa)
- (c₃) $(\mathbf{q} + \mathbf{p})^* = \mathbf{q}^* + \mathbf{p}^*$ (o conjugado da soma é soma dos conjugados)

Prova. Pondo $\mathbf{q} = [w, \mathbf{v}]$ e $\mathbf{p} = [w', \mathbf{v}']$ temos:

$$(c_1) \quad (\mathbf{q}^*)^* = ([w, -\mathbf{v}])^* = [w, -(-\mathbf{v})] = [w, \mathbf{v}] = \mathbf{q}$$

$$(c_2) \quad (\mathbf{qp})^* = ([w, \mathbf{v}][w', \mathbf{v}'])^* = [ww' - \mathbf{v} \cdot \mathbf{v}', -w\mathbf{v}' - w'\mathbf{v} - \mathbf{v} \wedge \mathbf{v}']^* \\ = [w', -\mathbf{v}'] [w, -\mathbf{v}] = \mathbf{p}^*\mathbf{q}^*$$

$$(c_3) \quad (\mathbf{q} + \mathbf{p})^* = ([w, \mathbf{v}] + [w', \mathbf{v}'])^* = ([w + w', \mathbf{v} + \mathbf{v}'])^* \\ = [w + w', -(\mathbf{v} + \mathbf{v}')] = [w, -\mathbf{v}] + [w', -\mathbf{v}'] = \mathbf{q}^* + \mathbf{p}^*$$

Proposição. Norma e Módulo de quatérnios têm as seguintes propriedades:

$$(n_1) \quad \|\mathbf{q}\| = \|\mathbf{q}^*\|$$

$$|\mathbf{q}| = |\mathbf{q}^*|$$

$$(n_2) \quad \|\mathbf{qp}\| = \|\mathbf{q}\| \|\mathbf{p}\|$$

$$|\mathbf{qp}| = |\mathbf{q}| |\mathbf{p}|$$

Prova. Como antes, pondo $\mathbf{q} = [w, \mathbf{v}]$ e $\mathbf{p} = [w', \mathbf{v}']$ temos:

$$\begin{aligned}
 (\mathbf{n}_1) \quad \|\mathbf{q}\| &= w^2 + x^2 + y^2 + z^2 = w^2 + (-x)^2 + (-y)^2 + (-z)^2 = \|\mathbf{q}^*\| \\
 |\mathbf{q}| &= \sqrt{\|\mathbf{q}\|} = \sqrt{\|\mathbf{q}^*\|} = |\mathbf{q}^*|
 \end{aligned}$$

$$\begin{aligned}
 (\mathbf{n}_2) \quad \|\mathbf{qp}\| &= (\mathbf{qp})(\mathbf{qp})^* = (\mathbf{qp})(\mathbf{p}^*\mathbf{q}^*) = \mathbf{q}(\mathbf{p}\mathbf{p}^*)\mathbf{q}^* = \|\mathbf{q}\| \|\mathbf{p}\| \\
 |\mathbf{qp}| &= \sqrt{\|\mathbf{qp}\|} = \sqrt{\|\mathbf{q}\| \|\mathbf{p}\|} = \sqrt{\|\mathbf{q}\|} \sqrt{\|\mathbf{p}\|}
 \end{aligned}$$

1.4 Unitários e Forma Polar.

Definição. Dizemos que um quatérnio \mathbf{q} é unitário se $|\mathbf{q}| = 1$.

A cada $\mathbf{q} \neq \mathbf{0} \in \mathbf{H}$ corresponde um quatérnio unitário $\hat{\mathbf{q}}$, chamado **unitário de \mathbf{q}** , definido por $\hat{\mathbf{q}} = \frac{\mathbf{q}}{|\mathbf{q}|}$.

Definição. O subconjunto dos quatérnios unitários:

$$\mathbf{S} = \{\mathbf{q} \in \mathbf{H} \mid |\mathbf{q}| = 1\},$$

é chamado **Esfera Unitária de \mathbf{H}** .

Proposição. valem as seguintes propriedades para quatérnios unitários:

- (s₁) $\mathbf{qp} \in \mathbf{S}$, $\forall \mathbf{q}, \mathbf{p} \in \mathbf{S}$ (produto de unitários é unitário)
- (s₂) $\mathbf{q}^{-1} \in \mathbf{S}$, $\forall \mathbf{q} \in \mathbf{S}$, (inverso de unitário é unitário)
- (s₃) $\mathbf{1} \in \mathbf{S}$ (o neutro multiplicativo é unitário)

Prova.

- (s₁) $|\mathbf{qp}| = |\mathbf{q}| |\mathbf{p}| = 1 \cdot 1 = 1$
- (s₂) $|\mathbf{q}^{-1}| = \left| \frac{\mathbf{q}}{\|\mathbf{q}\|} \right| = \frac{|\mathbf{q}|}{1} = |\mathbf{q}| = 1$
- (s₃) $|\mathbf{1}| = |[1, 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}]| = \sqrt{1} = 1$

Da proposição acima segue o:

Corolário. A esfera unitária \mathbf{S} é um subgrupo do grupo multiplicativo dos quatérnios não nulos:

$$\mathbf{H}^* = \{\mathbf{q} \in \mathbf{H} / \mathbf{q} \neq 0\}.$$

Proposição. (Forma Polar) Para cada $\mathbf{q} = [w, \mathbf{v}] \in \mathbf{S} = \{\mathbf{q} \in \mathbf{H} / |\mathbf{q}| = 1\}$, existe $\theta \in (-\pi, \pi]$ e um vetor unitário $\hat{\mathbf{u}} \in \mathbb{R}^3$ tal que,

$$\mathbf{q} = [\cos(\theta), \sin(\theta)\hat{\mathbf{u}}]$$

Prova. Seja $\mathbf{q} = [w, \mathbf{v}]$ unitário. Temos dois casos a considerar:

$$\mathbf{v} = 0ib + 0jb + 0kb = 0 \quad \text{e } \mathbf{v} \neq 0.$$

Se $\mathbf{v} = 0$ então $\mathbf{q} = [1, 0]$. Tomando-se $\theta = 0$ e $\hat{\mathbf{u}}$ um vetor unitário qualquer do \mathbb{R}^3 teremos: $\mathbf{q} = [1, 0] = [\cos(0), \sin(0)\hat{\mathbf{u}}]$.

Se $\mathbf{v} \neq 0$, então podemos escrever $\mathbf{q} = [w, b\hat{\mathbf{u}}]$, onde $b = |\mathbf{v}|$ e $\hat{\mathbf{u}} = \frac{\mathbf{v}}{|\mathbf{v}|}$, ou $b = -|\mathbf{v}|$ e $\hat{\mathbf{u}} = -\frac{\mathbf{v}}{|\mathbf{v}|}$. Sendo $|\mathbf{q}| = w^2 + b^2 = 1$ temos que o ponto (w, b) está na circunferência unitária do \mathbb{R}^2 , portanto existe $\theta \in (-\pi, \pi)$, tal que, $\cos(\theta) = w$ e $\sin(\theta) = b$. Assim temos:

$$\mathbf{q} = [\cos(\theta), \sin(\theta)\hat{\mathbf{u}}]$$

Observação. A representação acima não é única.

2 ROTAÇÕES 3D

2.1 Preliminares

Os parâmetros de uma rotação no espaço são: um eixo \mathbf{E} pela origem com direção $\hat{\mathbf{u}}$, chamado **eixo de rotação** e um ângulo de medida θ chamado **ângulo de rotação**. Nosso objetivo é contruir uma função $\mathbf{R}(\theta, \hat{\mathbf{u}}) : \mathbb{R}^3 \mapsto \mathbb{R}^3$ que associa a cada ponto $p(x, y, z)$ do \mathbb{R}^3 um único ponto $p'(x, y, z)$ do \mathbb{R}^3 , que é o rotacionado de p , em torno do eixo \mathbf{E} , de ângulo θ .

Vamos inicialmente obter uma expressão vetorial para função $\mathbf{R}(\theta, \hat{\mathbf{u}})$ que permitirá detectarmos sua linearidade e, em seguida, determinaremos sua representação matricial na base canônica do \mathbb{R}^3 . Posteriormente iremos obter uma representação de $\mathbf{R}(\theta, \hat{\mathbf{u}})$ por quatérnios.

2.2 Expressão Vetorial de $\mathbf{R}(\theta, \hat{\mathbf{u}})$ - Fórmula de Rodrigues

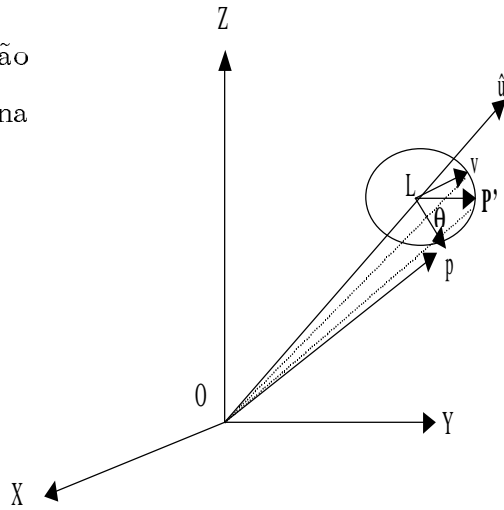
Na figura ao lado, o vetor \vec{OL} é a projeção do vetor \vec{Op} sobre o eixo \mathbf{E} pela origem na direção de $\hat{\mathbf{u}}$, portanto

$$\vec{OL} = \text{proj}_{\hat{\mathbf{u}}}(\vec{Op}) = (p \cdot \hat{\mathbf{u}}) \hat{\mathbf{u}}.$$

Estamos indicando por \vec{v} o vetor

$(\hat{\mathbf{u}} \wedge \vec{Lp})$, isto é,

$$\begin{aligned} \vec{v} &= \hat{\mathbf{u}} \wedge \vec{Lp} = \hat{\mathbf{u}} (p - (p \cdot \hat{\mathbf{u}}) \hat{\mathbf{u}}) = \\ &= \hat{\mathbf{u}} \wedge p. \end{aligned}$$



Observamos que $|\vec{v}| = |\hat{\mathbf{u}} \wedge \vec{Lp}| = |\hat{\mathbf{u}}| |\vec{Lp}| \text{sen}(\pi/2) = |\vec{Lp}|$. Também sendo $p' = \mathbf{R}(\theta, \hat{\mathbf{u}})p$, temos que $|\vec{Lp'}| = |\vec{Lp}|$, pois $\mathbf{R}(\theta, \hat{\mathbf{u}})$ preserva distâncias. Assim, $|\hat{\mathbf{v}}| = |\vec{Lp'}| = |\vec{Lp}|$ e $|\hat{\mathbf{v}}| \perp |\vec{Lp}$, logo

$$\vec{Lp'} = \cos(\theta) \vec{Lp} + \text{sen}(\theta) \vec{v},$$

portanto

$$\begin{aligned}
 p' &= \mathbf{R}(\theta, \hat{\mathbf{u}})p \\
 &= \vec{Op'} \\
 &= \vec{OL} + \vec{Lp'} \\
 &= (p \cdot \hat{\mathbf{u}}) \hat{\mathbf{u}} + \cos(\theta)(\vec{Lp}) + \text{sen}(\theta) \vec{\mathbf{v}} \\
 &= (p \cdot \hat{\mathbf{u}}) \hat{\mathbf{u}} + \cos(\theta)(\vec{Op} - \vec{OL}) + \text{sen}(\theta) \vec{\mathbf{v}} \\
 &= (p \cdot \hat{\mathbf{u}}) \hat{\mathbf{u}} + \cos(\theta)p - \cos(\theta)(p \cdot \hat{\mathbf{u}}) \hat{\mathbf{u}} + \text{sen}(\theta)(\hat{\mathbf{u}} \wedge p) \\
 &= \cos(\theta)p + (1 - \cos(\theta))(p \cdot \hat{\mathbf{u}}) \hat{\mathbf{u}} + \text{sen}(\theta)(\hat{\mathbf{u}} \wedge p)
 \end{aligned}$$

Provamos acima o seguinte:

Teorema. *A rotação de ângulo θ , de um ponto $p \in \mathbb{R}^3$ em torno de um eixo pela origem \mathbf{E} , com vetor diretor unitário $\hat{\mathbf{u}}$, é dada pela expressão:*

$$\mathbf{R}(\theta, \hat{\mathbf{u}})p = \cos(\theta)p + (1 - \cos(\theta))(p \cdot \hat{\mathbf{u}}) \hat{\mathbf{u}} + \text{sen}(\theta)(\hat{\mathbf{u}} \wedge p)$$

2.3 Linearidade e Representação Matricial de $\mathbf{R}(\theta, \hat{\mathbf{u}})$

Da expressão acima vemos que a rotação $\mathbf{R}(\theta, \hat{\mathbf{u}}) : \mathbb{R}^3 \mapsto \mathbb{R}^3$ é uma aplicação linear.

Indicando por $B = \{e_1 = (1, 0, 0), e_2 = (0, 1, 0), e_3 = (0, 0, 1)\}$ a base canônica do

\mathbb{R}^3 e assumindo que nesta base $\hat{\mathbf{u}} = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}$, temos:

$$\begin{aligned}
[\mathbf{R}(\theta, \hat{\mathbf{u}})]e_1 &= \cos(\theta) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + (1 - \cos(\theta))u_x \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} + \text{sen}(\theta) \begin{pmatrix} 0 \\ u_z \\ -u_y \end{pmatrix} \\
&= \begin{pmatrix} \cos(\theta) + (1 - \cos(\theta))u_x^2 \\ u_x u_y (1 - \cos(\theta)) + u_z \text{sen}(\theta) \\ u_x u_z (1 - \cos(\theta)) - u_y \text{sen}(\theta) \end{pmatrix}
\end{aligned}$$

Analogamente temos:

$$\begin{aligned}
[\mathbf{R}(\theta, \hat{\mathbf{u}})]e_2 &= \cos(\theta) \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + (1 - \cos(\theta))u_x \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} + \text{sen}(\theta) \begin{pmatrix} -u_z \\ 0 \\ u_x \end{pmatrix} \\
&= \begin{pmatrix} (1 - \cos(\theta))u_y u_x - \text{sen}(\theta)u_z \\ \cos(\theta) + (1 - \cos(\theta))u_y^2 \\ (1 - \cos(\theta))u_y u_z + \text{sen}(\theta)u_x \end{pmatrix} e
\end{aligned}$$

$$\begin{aligned}
[\mathbf{R}(\theta, \hat{\mathbf{u}})]e_3 &= \cos(\theta) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + (1 - \cos(\theta))u_x \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} + \text{sen}(\theta) \begin{pmatrix} -u_y \\ -u_x \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} (1 - \cos(\theta))u_z u_x + \text{sen}(\theta)u_y \\ (1 - \cos(\theta))u_z u_y - \text{sen}(\theta)u_x \\ \cos(\theta) + (1 - \cos(\theta))u_z^2 \end{pmatrix}, \text{ portanto}
\end{aligned}$$

$$[\mathbf{R}(\theta, \hat{\mathbf{u}})] =$$

$$\begin{pmatrix} \cos(\theta) + (1 - \cos(\theta))u_x^2 & (1 - \cos(\theta))u_y u_x - \text{sen}(\theta)e_z & (1 - \cos(\theta))u_z u_x + \text{sen}(\theta)e_y \\ u_x u_y (1 - \cos(\theta)) + u_z \text{sen}(\theta) & \cos(\theta) + (1 - \cos(\theta))u_y^2 & (1 - \cos(\theta))u_z u_y - \text{sen}(\theta)u_x \\ u_x u_z (1 - \cos(\theta)) - u_y \text{sen}(\theta) & (1 - \cos(\theta))u_y u_z + \text{sen}(\theta)u_x & \cos(\theta) + (1 - \cos(\theta))u_z^2 \end{pmatrix}$$

Pondo $c = \cos(\theta)$ e $s = \text{sen}(\theta)$ obtemos a matriz simplificada:

$$[\mathbf{R}(\theta, \hat{\mathbf{u}})] = \begin{pmatrix} c + (1 - c)u_x^2 & (1 - c)u_x u_y - s u_z & (1 - c)u_x u_z + s u_y \\ (1 - c)u_x u_y + s u_z & c + (1 - c)u_y^2 & (1 - c)u_y u_z - s u_x \\ (1 - c)u_x u_z - s u_y & (1 - c)u_y u_z + s u_x & c + (1 - c)u_z^2 \end{pmatrix} \quad (*)$$

2.4 Rotações em torno dos Eixos Coordenados

Se o eixo de rotação é um dos eixos coordenados, digamos $E = Ox$, então o vetor diretor é $\hat{\mathbf{u}} = (u_x, u_y, u_z) = (1, 0, 0)$. Substituindo em (*) obtemos a conhecida matriz de rotação em torno de Ox :

$$[R(\theta, Ox)] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\text{sen}(\theta) \\ 0 & \text{sen}(\theta) & \cos(\theta) \end{pmatrix}$$

Analogamente, para os casos $E = Oy$ e $E = Oz$, temos $\hat{\mathbf{u}} = (0, 1, 0)$ e $\hat{\mathbf{u}} = (0, 0, 1)$, valores que substituídos em (*) fornecem as matrizes:

$$[R(\theta, Oy)] = \begin{pmatrix} \cos(\theta) & 0 & \text{sen}(\theta) \\ 0 & 1 & 0 \\ -\text{sen}(\theta) & 0 & \cos(\theta) \end{pmatrix} e$$

$$[R(\theta, Oz)] = \begin{pmatrix} \cos(\theta) & -\text{sen}(\theta) & 0 \\ \text{sen}(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2.5 Parametrização por Ângulos de Euler

Uma rotação \mathbf{R} transforma a base canônica $B = \{e_1, e_2, e_3\}$ do \mathbb{R}^3 em outra base ortonormal positiva $B' = \{b_1, b_2, b_3\}$, isto é, $\mathbf{R}(e_1) = f_1$, $\mathbf{R}(e_2) = f_2$, $\mathbf{R}(e_3) = f_3$. Pode-se mostrar (Gomes et ali., 1998), que existem três rotações sucessivas em torno de eixos distintos do espaço, de ângulos respectivamente ϕ , θ e ψ , tais que,

$$\mathbf{R} = \mathbf{R}_\phi \circ \mathbf{R}_\theta \circ \mathbf{R}_\psi$$

Define-se ϕ , θ e ψ , como os ângulos de Euler da rotação \mathbf{R} , os quais são largamente usados em Mecânica Clássica.

A definição de ângulos de Euler dada acima explora a idéia de rotação como mudança de coordenadas. O conceito de ângulos de Euler utilizados em Robótica, Computação Gráfica e Aviação é resultante de outra abordagem, que mostraremos a seguir.

Seja \mathbf{R} uma rotação de ângulo θ , em torno de um eixo \mathbf{E} pela origem. É possível mostrar (Gomes et ali., 1998) que existem rotações \mathbf{R}_{θ_x} , \mathbf{R}_{θ_y} , e \mathbf{R}_{θ_z} , em torno dos eixos coordenados Ox , Oy e Oz respectivamente, tais que,

$$\mathbf{R} = \mathbf{R}(\theta_x, \theta_y, \theta_z) \stackrel{def}{=} \mathbf{R}_{\theta_x} \circ \mathbf{R}_{\theta_y} \circ \mathbf{R}_{\theta_z}$$

A representação acima apresenta vários problemas. Um deles é a falta de unicidade, o que significa que uma mesma terna de ângulos de Euler pode representar várias rotações. Isto ocorre porque a representação depende da fixação da ordem

dos eixos coordenados (anisotropia). Existem doze possibilidades que tornam a representação ambígua. Outro problema é que a determinação dos ângulos de Euler de uma dada rotação não é simples. Também a parametrização que se obtém com estes ângulos não é global. Um sério problema, talvez o mais perturbador, é o fenômeno conhecido com "Gimbal Lock", que pode ser traduzido matematicamente como perda de um grau de liberdade e ocorre sempre que um dos ângulos parâmetro tem 90° . Para exemplificar, consideremos a rotação $\mathbf{R} = \mathbf{R}(\theta_x, 90, \theta_z)$. Usando relações trigonométricas mostra-se que $\mathbf{R} = \mathbf{R}((\theta_x - \theta_z), 90, (\theta_x - \theta_z))$. ou seja, perdeu-se um grau de liberdade. Um exemplo concreto, de fácil verificação geométrica, é que a rotação $\mathbf{R}(90,90,90)$ tem o mesmo efeito que a rotação $\mathbf{R}(0,90,0)$.

2.6 Parametrização de Rotações por Quatérnios

Lembremos que, conforme demonstrado em 3.2, uma rotação de ângulo θ em torno de um eixo E com vetor diretor unitário $\hat{\mathbf{u}}$, é dada pela função: $\mathbf{R}(\theta, \hat{\mathbf{u}}) : \mathbb{R}^3 \mapsto \mathbb{R}^3$, definida por:

$$\mathbf{R}(\theta, \hat{\mathbf{u}})p = \cos(\theta)p + (1 - \cos(\theta))(p \cdot \hat{\mathbf{u}})\hat{\mathbf{u}} + \sin(\theta)(\hat{\mathbf{u}} \wedge p), \quad \forall p \in \mathbb{R}^3.$$

Vimos também em 2.4, que todo quatérnio unitário

$$\mathbf{q} = [w, (x, y, z)] = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \in \mathbf{S},$$

pode ser escrito na forma "polar"

$$\mathbf{q} = [\cos(\theta), \sin(\theta)\hat{\mathbf{u}}],$$

onde $\theta \in (-\pi, \pi]$ e $\hat{\mathbf{u}}$ é um vetor unitário do \mathbb{R}^3 .

Daremos a seguir a definição de uma função que desempenhará papel fundamental na representação de rotações por quatérnios.

Definição . Para cada $\mathbf{q} \in \mathbf{S}$, $\varphi_{\mathbf{q}}$ indicará a função que associa a cada quatérnio puro $\mathbf{p} = [0, (x, y, z)] = 0 + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ o quatérnio puro $\varphi_{\mathbf{q}}(\mathbf{p}) = \mathbf{qpq}^{-1}$.

Para dar consistência à definição acima demonstraremos a seguinte:

Proposição. Se $\mathbf{q} = [w, \mathbf{v}] \in \mathbf{S}$ e $\mathbf{p} = [0, \mathbf{v}]$ é um quatérnio puro, então o quatérnio $\mathbf{p}' = \mathbf{q}\mathbf{p}\mathbf{q}^{-1}$ é puro, isto é, tem parte real nula.

Prova. Indicando-se por $Re(\mathbf{q})$ a parte real de um quatérnio \mathbf{q} , mostra-se facilmente a relação semelhante a dos números complexos: $2Re(\mathbf{q}) = \mathbf{q} + \mathbf{q}^*$. Logo,

$$\begin{aligned}
 2Re(\mathbf{q}) &= (\mathbf{p}' + \mathbf{p}'^*) \\
 &= (\mathbf{q}\mathbf{p}'\mathbf{q}^{-1}) + (\mathbf{q}\mathbf{p}'\mathbf{q}^{-1})^* \\
 &= (\mathbf{q}\mathbf{p}'\mathbf{q}^*) + (\mathbf{q}\mathbf{p}'\mathbf{q}^*)^* \\
 &= (\mathbf{q}\mathbf{p}'\mathbf{q}^*) + (\mathbf{q}\mathbf{p}'\mathbf{q}^*)^* \\
 &= (\mathbf{q}\mathbf{p}'\mathbf{q}^*) + (\mathbf{q}\mathbf{p}'^*\mathbf{q}^*) \\
 &= \mathbf{q}(\mathbf{p}' + \mathbf{p}'^*)\mathbf{q}^* \\
 &= \mathbf{q} \mathbf{0} \mathbf{q}^* \\
 &= \mathbf{0} \\
 &\equiv 0
 \end{aligned}$$

Mostraremos a seguir o seguinte resultado fundamental:

Teorema. Se $\mathbf{q} = [w, \mathbf{v}] = [\cos(\theta) \text{ sen}(\theta)\hat{\mathbf{u}}] \in \mathbf{S}$, então, identificando \mathbb{R}^3 com o subconjunto dos quatérnios com parte real nula, tem-se

$$\varphi_{\mathbf{q}} = \mathbf{R}(2\theta, \hat{\mathbf{u}}).$$

Prova. Seja $p \in \mathbb{R}^3$. Identificaremos p com o quatérnio $\mathbf{p} = [0, p]$. Temos:

$$\begin{aligned}
 \varphi_{\mathbf{q}}(p) &= \mathbf{q}\mathbf{p}\mathbf{q}^{-1} \\
 &= \mathbf{q}\mathbf{p}\mathbf{q}^* \\
 &= [w, \mathbf{v}][0, p][w, \mathbf{v}] \\
 &= [w, \mathbf{v}][p \cdot \mathbf{v}, wp - (p \wedge \mathbf{v})] \\
 &= [w(p \cdot \mathbf{v}) - \mathbf{v} \cdot (wp - (p \wedge \mathbf{v})), w(wp - (p \wedge \mathbf{v})) + (p \cdot \mathbf{v})\mathbf{v} + \mathbf{v} \wedge (wp - (p \wedge \mathbf{v}))] \\
 &= [0, w^2p - w(p \wedge \mathbf{v}) + (p \cdot \mathbf{v})\mathbf{v} - w(p \wedge \mathbf{v}) + (\mathbf{v} \cdot p)\mathbf{v} - (\mathbf{v} \cdot \mathbf{v})p] \\
 &= [0, (w^2 - (\mathbf{v} \cdot \mathbf{v}))p + 2(p \cdot \mathbf{v})\mathbf{v} + 2w(\mathbf{v} \wedge p)]
 \end{aligned}$$

Sendo $w = \cos(\theta)$, $\mathbf{v} = \text{sen}(\theta)\hat{\mathbf{u}}$ temos:

$$\begin{aligned}
\varphi_{\mathbf{q}}(p) &= \mathbf{qpq}^{-1} \\
&= [0, (\cos^2(\theta) - \text{sen}^2(\theta))p + 2\text{sen}^2(\theta)(p \cdot \hat{\mathbf{u}})\hat{\mathbf{u}} + 2\cos(\theta)\text{sen}(\theta)(\hat{\mathbf{u}} \wedge p)] \\
&= [0, \cos(2\theta)p + (1 - \cos(2\theta))(p \cdot \hat{\mathbf{u}})\hat{\mathbf{u}} + \text{sen}(2\theta)(\hat{\mathbf{u}} \wedge p)] \\
&\equiv \cos(2\theta)p + (1 - \cos(2\theta))(p \cdot \hat{\mathbf{u}})\hat{\mathbf{u}} + \text{sen}(2\theta)(\hat{\mathbf{u}} \wedge p) \\
&= \mathbf{R}(2\theta, \hat{\mathbf{u}})(p).
\end{aligned}$$

Corolário. Se $\mathbf{q} = [w, \mathbf{v}] = [\cos(\theta/2), \text{sen}(\theta/2)\hat{\mathbf{u}}] \in \mathbf{S}$, então, identificando \mathbb{R}^3 com o subconjunto dos quatérnios com parte real nula, tem-se

$$\varphi_{\mathbf{q}} = \mathbf{R}(\theta, \hat{\mathbf{u}})$$

A proposição seguinte mostra que a rotação parametrizada por um quatérnio \mathbf{q}_2 , seguida de uma rotação parametrizada por um quatérnio \mathbf{q}_1 é a rotação parametrizada pelo quatérnio produto $\mathbf{q}_1\mathbf{q}_2$. A facilidade para compor rotações representadas por quatérnios, entre outras, é uma das boas vantagens desta parametrização.

Proposição. Se $\mathbf{q}_1, \mathbf{q}_2 \in \mathbf{S}$, então

$$\varphi_{\mathbf{q}_1}\varphi_{\mathbf{q}_2} = \varphi_{\mathbf{q}_1\mathbf{q}_2}$$

Prova. Seja $\mathbf{p} = [0, p] \in \mathbf{R}^3$. Tem-se:

$$\begin{aligned}
(\varphi_{\mathbf{q}_1}\varphi_{\mathbf{q}_2})(\mathbf{p}) &= \varphi_{\mathbf{q}_1}(\varphi_{\mathbf{q}_2}(\mathbf{p})) \\
&= \varphi_{\mathbf{q}_1}(\mathbf{q}_2\mathbf{p}\mathbf{q}_2^{-1}) \\
&= \mathbf{q}_1(\mathbf{q}_2\mathbf{p}\mathbf{q}_2^{-1})\mathbf{q}_1^{-1} \\
&= (\mathbf{q}_1\mathbf{q}_2)\mathbf{p}(\mathbf{q}_1\mathbf{q}_2)^{-1} \\
&= (\varphi_{\mathbf{q}_1\mathbf{q}_2})(\mathbf{p})
\end{aligned}$$

2.7 Matriz de uma Rotação parametrizada por Quatérnio

A seguir vamos determinar a matriz ortogonal positiva correspondente à rotação $\varphi_{\mathbf{q}}$, parametrizada pelo quatérnio \mathbf{q} . Mais precisamente vamos mostrar o seguinte resultado:

Proposição. Se $\mathbf{q} = [w, (x, y, z)] \in \mathbf{S}$, então a matriz da rotação $\varphi_{\mathbf{q}}$ em coordenadas cartesianas é definida por:

$$[\varphi_{\mathbf{q}}] = \begin{pmatrix} 1 - 2(y^2 + z^2) & 2xy - 2wz & 2xz + 2yw \\ 2xy + 2wz & 1 - 2(x^2 + z^2) & 2yz - 2wz \\ 2xz - 2yw & 2yz + 2wz & 1 - 2(x^2 + y^2) \end{pmatrix}$$

Prova. Para quaisquer $\mathbf{q}_1 = [w_1, (x_1, y_1, z_1)]$ e $\mathbf{q}_2 = [w_2, (x_2, y_2, z_2)]$ fixados em \mathbf{H} , podemos considerar as funções lineares $\mathbf{L}_{\mathbf{q}_1}$ e $\mathbf{R}_{\mathbf{q}_2}$ de \mathbf{H} em \mathbf{H} , definidas por:

$$\begin{aligned} \mathbf{L}_{\mathbf{q}_1}(q) &= \mathbf{q}_1 \mathbf{q}, \quad \forall \mathbf{q} \in \mathbf{H} \quad (\text{multiplicação à esquerda}) \\ \mathbf{R}_{\mathbf{q}_2}(q) &= \mathbf{q} \mathbf{q}_2, \quad \forall \mathbf{q} \in \mathbf{H} \quad (\text{multiplicação à direita}) \end{aligned}$$

O produto de dois quatérnios $\mathbf{q}_1 \mathbf{q}_2$, onde $\mathbf{q}_1 = [w_1, (x_1, y_1, z_1)]$ e $\mathbf{q}_2 = [w_2, (x_2, y_2, z_2)]$, é conforme 2.1, definido por:

$$\begin{aligned} \mathbf{q}_1 \mathbf{q}_2 &= w_1 w_2 - x_1 x_2 - y_1 y_2 - z_1 z_2 \\ &+ (w_1 x_2 + w_2 x_1 + y_1 z_2 - z_1 y_2) \mathbf{i} \\ &+ (w_1 y_2 + w_2 y_1 + z_1 x_2 - x_1 z_2) \mathbf{j} \\ &+ (w_1 z_2 + w_2 z_1 + x_1 y_2 - y_1 x_2) \mathbf{k} \end{aligned}$$

Decorre daí que na base $\{\mathbf{i}, \mathbf{j}, \mathbf{k}, \mathbf{1}\}$ as funções $\mathbf{L}_{\mathbf{q}_1}$ e $\mathbf{R}_{\mathbf{q}_2}$ podem ser representadas pelas matrizes:

$$[\mathbf{L}_{\mathbf{q}_1}] = \begin{pmatrix} w_1 & -z_1 & y_1 & x_1 \\ z_1 & w_1 & -x_1 & y_1 \\ -y_1 & x_1 & w_1 & z_1 \\ -x_1 & -y_1 & -z_1 & w_1 \end{pmatrix}, \quad [\mathbf{R}_{\mathbf{q}_2}] = \begin{pmatrix} w_2 & z_2 & -y_2 & x_2 \\ -z_2 & w_2 & x_2 & y_2 \\ y_2 & -x_2 & w_2 & z_2 \\ -x_2 & -y_2 & -z_2 & w_2 \end{pmatrix},$$

O produto $\mathbf{q}\mathbf{p}\mathbf{q}^{-1}$ pode ser visto como uma multiplicação de \mathbf{p} por \mathbf{q}^{-1} à direita, seguido de uma multiplicação por \mathbf{q} à esquerda.

Portanto temos:

$$\begin{aligned} \varphi_{\mathbf{q}}(\mathbf{p}) &= \mathbf{q}(\mathbf{p}\mathbf{q}^{-1}) \\ &= \mathbf{L}_{\mathbf{q}}(\mathbf{R}_{\mathbf{q}^{-1}}(\mathbf{p})) \end{aligned}$$

Segue que :

$$\begin{aligned} [\varphi_{\mathbf{q}}] &= [\mathbf{L}_{\mathbf{q}} \mathbf{R}_{\mathbf{q}^{-1}}] \\ &= [\mathbf{L}_{\mathbf{q}}][\mathbf{R}_{\mathbf{q}^{-1}}] \\ &= \begin{pmatrix} w^2 + x^2 - y^2 - z^2 & 2xy - 2wz & 2xz + 2wy & 0 \\ 2xy + 2wz & y^2 - z^2 + w^2 - x^2 & 2yz - 2wz & 0 \\ 2xz - 2wz & 2yz + 2wz & z^2 - y^2 - x^2 + w^2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

Sendo \mathbf{q} unitário temos as identidades:

$$\begin{aligned} w^2 + x^2 + y^2 + z^2 &= 1 \\ w^2 + x^2 &= 1 - (y^2 + z^2) \\ w^2 + y^2 &= 1 - (x^2 + z^2) \\ w^2 + z^2 &= 1 - (x^2 + y^2) \end{aligned}$$

Substituindo estes valores na última matriz obtemos a matriz de $\varphi_{\mathbf{q}}$ em coordenadas homogêneas:

$$[\varphi_{\mathbf{q}}]_h = \begin{pmatrix} 1 - 2(y^2 + z^2) & 2xz - 2wz & 2xz + 2wy & 0 \\ 2xy + 2wz & 1 - 2(x^2 + z^2) & 2yz - 2wz & 0 \\ 2xz - 2wz & 2yz + 2wz & 1 - 2(x^2 + y^2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Eliminando a última linha e a última coluna obtemos a matriz de $\varphi_{\mathbf{q}}$ em coordenadas cartesianas:

$$[\varphi_{\mathbf{q}}] = \begin{pmatrix} 1 - 2(y^2 + z^2) & 2xy - 2wz & 2xz + 2yw \\ 2xy + 2wz & 1 - 2(x^2 + z^2) & 2yz - 2wz \\ 2xz - 2yw & 2yz + 2wz & 1 - 2(x^2 + y^2) \end{pmatrix}$$

3 ANIMAÇÃO E INTERPOLAÇÃO DE ROTAÇÕES

3.1 Preliminares.

Nas seções anteriores, estudamos quatro alternativas para representar uma rotação: Ângulo-eixo, Matrizes, Ângulos de Euler e Quatérnios. A seguir vamos examinar a adequação destes parâmetros com relação aos métodos de interpolação e animação de rotações.

Definição. Seja \mathbf{M} um espaço métrico. Uma interpolação entre dois pontos A e B de \mathbf{M} é uma curva contínua $\gamma(A, B, t) : [0, 1] \mapsto M$, tal que,

$$\begin{aligned}\gamma(A, B, 0) &= A \quad \text{e,} \\ \gamma(A, B, 1) &= B.\end{aligned}$$

Nas aplicações que faremos, \mathbf{M} será um dos espaços: \mathbb{R}^3 , $SO(3)$, $M_3(\mathbb{R})$ ou a esfera unitária \mathbf{S} de \mathbf{H} . A curva $\gamma(A, B, [0, 1])$ será um segmento de reta ou um grande arco sobre a esfera \mathbf{S} , ligando o ponto A ao ponto B .

3.2 Interpolação Linear de Rotações representadas por Matrizes

Quando representamos rotações por matrizes ortogonais positivas, estamos fazendo uso do conjunto $SO(3)$ como espaço de parâmetros. A interpolação mais simples entre duas matrizes $A, B \in SO(3)$, é definida pela função:

$$\text{LinMat}(A, B, t) = (1 - t)A + tB : [0, 1] \mapsto M_3(\mathbb{R})$$

chamada **Interpolação Linear de Matrizes**.

Uma séria restrição ao uso desta modalidade de interpolação é que o conjunto

$SO(3)$ não é fechado em relação às operações adição e multiplicação por escalar. Em consequência a matriz interpolada $(1-t)A + tB$ pode não ser ortogonal positiva, portanto pode não representar uma rotação.

Este problema está presente mesmo nos casos mais simples de interpolação de rotações em torno de um mesmo eixo. Na figura 1.4 abaixo, mostramos uma sequência de 24 retângulos, gerados pela aplicação ao retângulo original $M := \{(2,0,0), (5,0,0), (2,0,5), (5,0,5)\}$, das rotações interpoladas por $\mathbf{LinMat}(\mathbf{A},\mathbf{B},t)$ entre as matrizes

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad e \quad B = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Estas matrizes representam rotações em torno do eixo Oz , respectivamente de 0° e 90° . O resultado da animação é completamente insatisfatório.

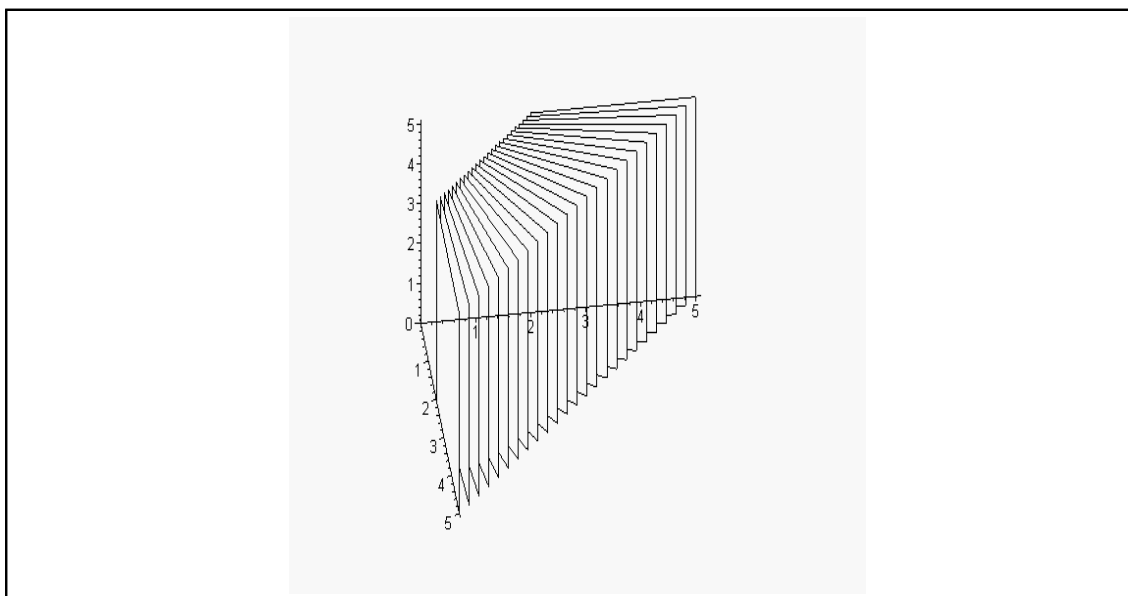


Figura 1.4 : Animação gerada por Interpolação Linear de Matrizes, representantes de rotações em torno de um mesmo eixo.

No caso de interpolação de matrizes que representam rotações em torno de eixos diferentes, o resultado também é insatisfatório pelas mesmas razões já expostas.

Na figura 2.4 apresentamos uma sequência de rotações aplicadas ao retângulo M , geradas pela interpolação linear entre as matrizes:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \quad e \quad B = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix},$$

as quais representam respectivamente, rotação de 90° em torno do eixo Ox e rotação de 180° em torno do eixo E pela origem, na direção do vetor $\hat{\mathbf{u}} = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0)$.

Pode-se verificar que

$$\det(\mathbf{LinMat}(\mathbf{A}, \mathbf{B}, t)) = 1 - 3t + 3t^2,$$

e então as matrizes interpoladas ($t \neq 0$ e $t \neq 1$) não são matrizes ortogonais positivas, logo não representam rotações.

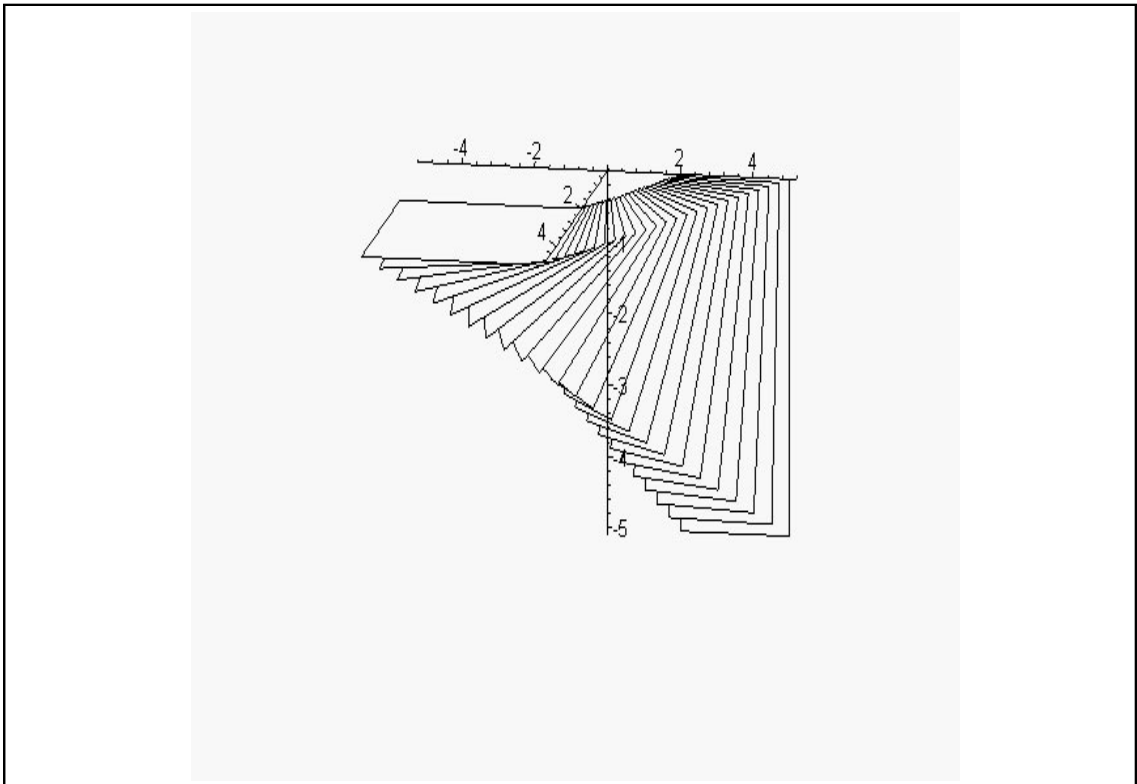


Figura 2.4 : Animação gerada por Interpolação Linear de Matrizes, representantes de rotações em torno de eixos diferentes. .

3.3 Interpolação Linear de Rotações representadas por Ângulos de Euler.

Uma rotação expressa por Ângulos de Euler tem a forma: $R = R(\theta_x, \theta_y, \theta_z)$. É usual fixar a ordem das rotações parâmetros em torno dos eixos coordenados como sendo da direita para a esquerda, ou seja, o efeito da rotação R é obtido aplicando-se ao objeto primeiro a rotação de ângulo θ_z em torno de Oz , seguida da rotação de ângulo θ_y em torno do eixo Oy , seguida da rotação de ângulo θ_x em torno do eixo Ox .

A terna de números $(\theta_x, \theta_y, \theta_z)$ pertence ao espaço \mathbb{R}^3 . Por esta razão dizemos que uma rotação especificada por ângulos de Euler está parametrizada pelo espaço \mathbb{R}^3 . Assim, interpolar estas rotações equivale a interpolar pontos do \mathbb{R}^3 .

Dados dois pontos $P(x_0, y_0, z_0)$ e $Q(x_1, y_1, z_1) \in \mathbb{R}^3$, a interpolação definida pela curva:

$$\text{LinEuler}(\mathbf{P}, \mathbf{Q}, t) = (1 - t)\mathbf{P} + t\mathbf{Q} : [0, 1] \mapsto \mathbb{R}^3$$

é chamada **Interpolação Linear de Ângulos de Euler**

Na figura 3.4 mostramos uma sequência de triângulos, obtida pela aplicação ao triângulo original $Tr := \{(1, 1, 1), (3, 3, 1), (2, 2, 4)\}$ das rotações interpoladas por $\text{LinEuler}(\mathbf{P}, \mathbf{Q}, t)$, entre as rotações em torno do eixo Ox , definidas por: $R_1 = R(0, 0, 0)$ e $R_2 = R(360, 0, 0, 0)$. Neste caso simples o resultado é satisfatório. A razão é que as rotações interpoladas $((1 - t)R_1 + tR_2)_{(0 < t < 1)}$ são bem formadas, isto é, são transformações lineares ortogonais positivas.

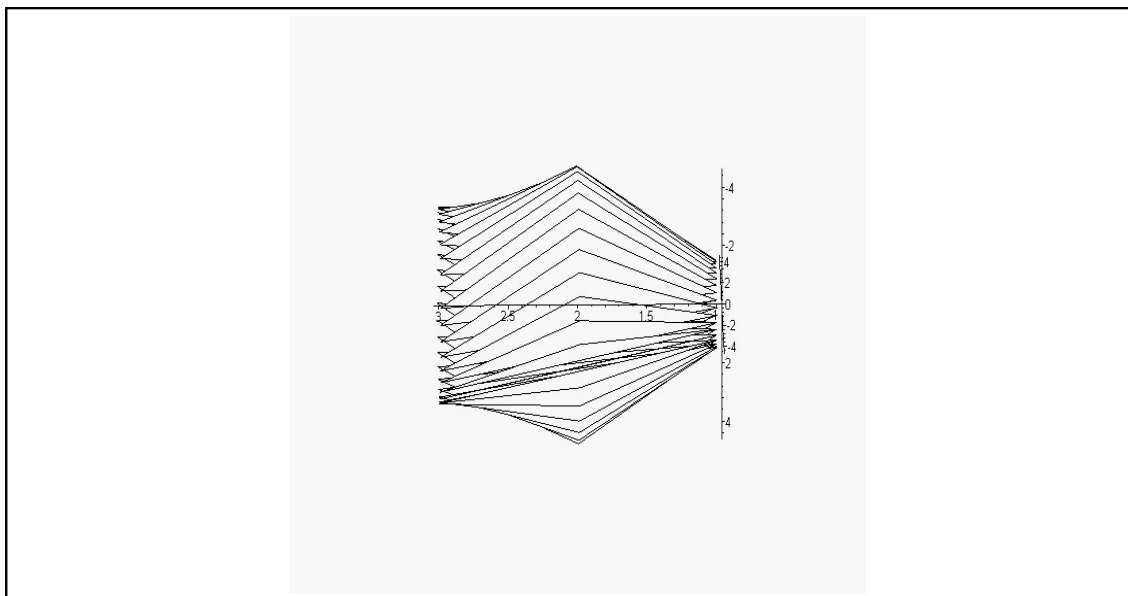


Figura 3.4 : Animação gerada por Interpolação Linear de ângulos de Euler, representantes de rotações em torno de um mesmo eixo. .

Nos casos em que os eixos de rotação são diferentes o resultado pode não ser satisfatório. Em geral quando se interpola rotações em torno de eixos diferentes, parametrizadas por Ângulos de Euler, as imagens tendem a se mover para os lados, gerando animações fora da realidade.

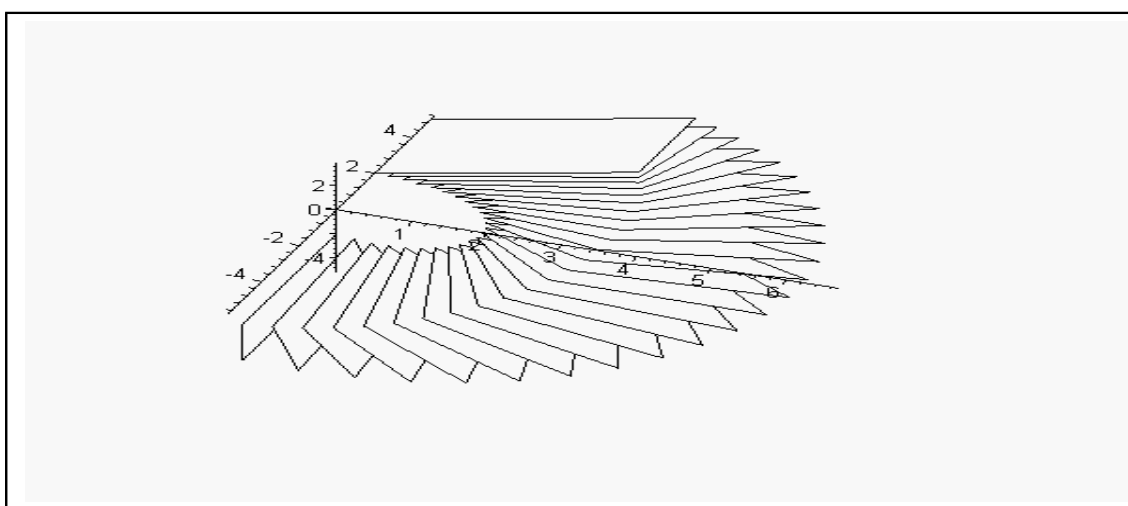


Figura 4.4 : Animação gerada por Interpolação Linear de ângulos de Euler, representantes de rotações em torno de eixos diferentes.

Na figura 4.4 mostramos a animação de uma sequência de imagens do retângulo M , cujas orientações foram interpoladas por $\mathbf{LinEuler}(\mathbf{P}, \mathbf{Q}, t)$, entre as rotações definidas pelos ângulos de Euler $R_1 = (90, 0, -90)$ e $R_2 = (0, 45, 90)$.

3.4 Interpolação Linear de Rotações representadas por Quatérnios.

Vimos anteriormente que quatérnios unitários parametrizam rotações no espaço tridimensional. Mais precisamente, se $\mathbf{q} = [w, (x, y, z)] = [\cos(\theta), \text{sen}(\theta)\hat{\mathbf{u}}]$, onde $\hat{\mathbf{u}}$ é um vetor unitário do \mathbb{R}^3 , então

$$\varphi_{\mathbf{q}} = \mathbf{R}(2\theta, \hat{\mathbf{u}}).$$

Dados \mathbf{q}_0 e \mathbf{q}_1 na esfera unitária \mathbf{S} de \mathbf{H} , a interpolação entre \mathbf{q}_0 e \mathbf{q}_1 definida por

$$\mathbf{Lerp}(\mathbf{q}_0, \mathbf{q}_1, t) = (1-t)\mathbf{q}_0 + t\mathbf{q}_1 : [0, 1] \mapsto \mathbf{H}$$

é chamada **Interpolação Linear Quatérnica**.

O conjunto $\{(1-t)\mathbf{q}_0 + t\mathbf{q}_1, t \in [0, 1]\}$ é um segmento em \mathbf{H} , ligando \mathbf{q}_0 a \mathbf{q}_1 , com todos os pontos (exceto os extremos) interiores a esfera \mathbf{S} . Assim os quatérnios interpolados não possuem módulos unitários. Embora representem a mesma rotação que seus correspondentes em \mathbf{S} , o fato de não serem unitários interfere na velocidade da curva interpoladora, provocando o aparecimento de quadros com espaçamentos irregulares. Este efeito indesejável pode ser corrigido com operações de normalização de altíssimo custo computacional.

Na figura 5.4 exibimos a animação de uma sequência de imagens do retângulo M , obtidos pela interpolação linear dos quatérnios unitários:

$$\begin{aligned} \mathbf{q}_0 &= \left[\frac{1}{2}, \left(\frac{1}{2}, \frac{1}{2}, \frac{-1}{2} \right) \right] \\ \mathbf{q}_1 &= [0.6532, (0.2705, 0.2705, 0.6532)] \end{aligned}$$

os quais parametrizam as rotações da animação anterior, respectivamente $R_1 = R(90, 0, -90)$ e $R_2 = (0, 45, 90)$.

Observa-se claramente a aglutinação de quadros na metade da curva, onde a velocidade é máxima.

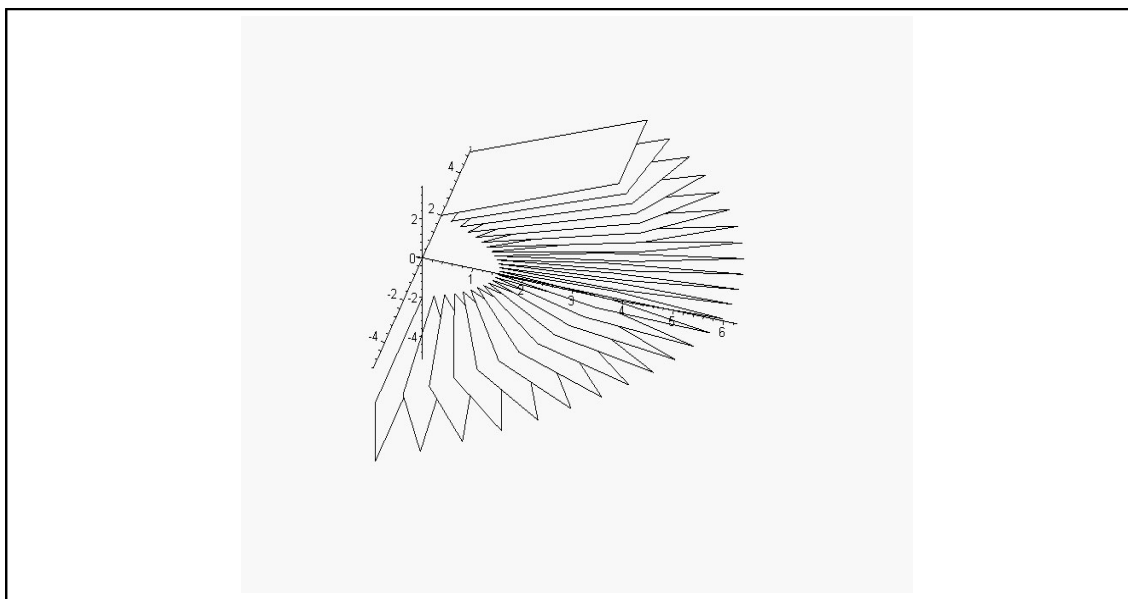


Figura 5.4 : Animação gerada por Interpolação Linear de quatérnios, representantes de rotações em torno de eixos diferentes.

3.5 Interpolação Linear Esférica de Rotações representadas por Quatérnios.

Conforme vimos na seção anterior, a interpolação linear entre dois quatérnios \mathbf{q}_0 e \mathbf{q}_1 , através de um segmento unindo \mathbf{q}_0 a \mathbf{q}_1 , tem o inconveniente de não manter constante a velocidade nem a norma unitária, afetando a regularidade da distribuição dos quadros.

Para evitar tais problemas, a idéia é interpolar estes pontos através de um caminho sobre a esfera unitária \mathbf{S} , ligando \mathbf{q}_0 a \mathbf{q}_1 , chamada "grande arco de interpolação". Este caminho é definido pelo gráfico da função:

$$\mathbf{SLerp}(\mathbf{q}_0, \mathbf{q}_1, t) = \frac{\text{sen}((1-t)\phi)}{\text{sen}(\phi)}\mathbf{q}_0 + \frac{\text{sen}(t\phi)}{\text{sen}(\phi)}\mathbf{q}_1 : [0, 1] \mapsto S^3$$

onde ϕ é o ângulo formado pelos vetores \mathbf{q}_0 e \mathbf{q}_1 , definido por $\phi = \cos^{-1}(\mathbf{q}_0 \cdot \mathbf{q}_1)$.

A função acima é chamada **Interpolação Linear Esférica de Quatérnios**.

A regularidade da animação de quadros interpolados por este método se justifica pelo fato de que a interpolação se dá em torno de um eixo, exatamente como estabelecido pelo Teorema de Euler, o qual em linguagem geométrica atual poderia ser enunciado como segue:

Dadas duas orientações O e O' de um objeto, existe um eixo $E \subset \mathbb{R}^3$ e um ângulo $\phi \in (-\pi, \pi]$, tal que, O' é o rotacionado de O , de ângulo ϕ em torno de E .

Para estabelecer comparação interpolamos os quatérnios da animação anterior usando a função $\mathbf{SLerp}(\mathbf{q}_0, \mathbf{q}_1, t)$. Os resultados exibidos na figura 6.4 abaixo mostram a eficiência deste método de interpolação.

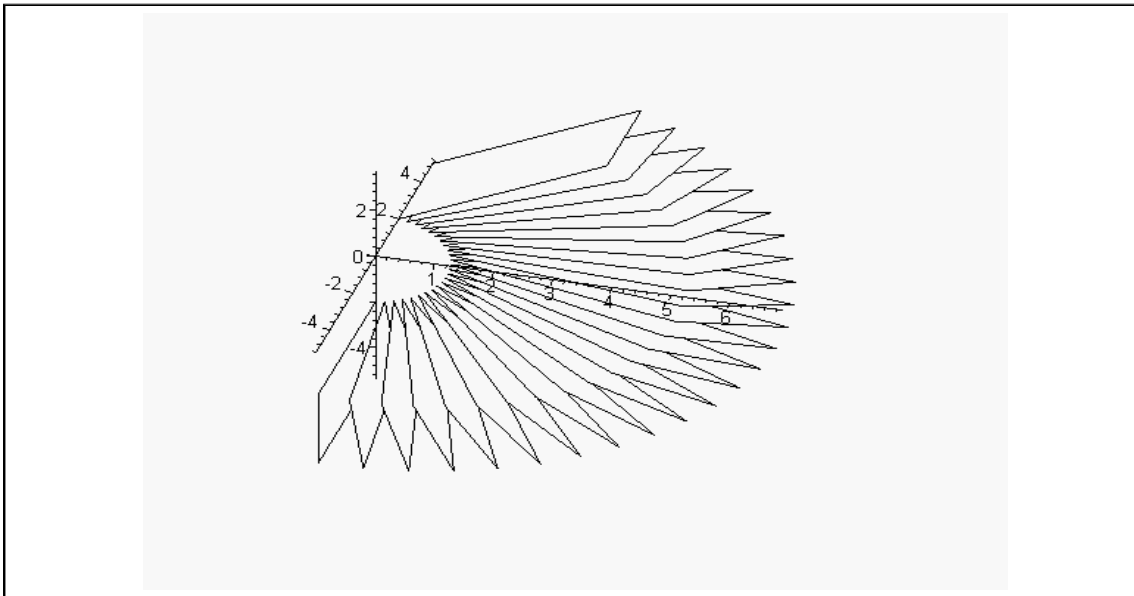


Figura 6.4 : Animação gerada por Interpolação Linear Esférica de quatérnios, representantes de rotações em torno de eixos diferentes.

CONCLUSÕES

Estudo Comparativo das representações

O ideal seria parametrizar o espaço das rotações por um conjunto cujos elementos pudessem ser caracterizados por apenas três valores. A estrutura algébrica desta parametrização hipotética deveria ser simples, de modo a permitir que as operações de armazenamento, concatenação e interpolação pudessem ser efetuadas com eficiência e baixo custo computacional.

A seguir, vamos analisar dentre as parametrizações estudadas: ângulo-eixo, matricial, ângulos de Euler e quatérnica, qual delas mais se aproxima da ideal.

Representação ângulo-eixo

Esta representação utiliza um vetor unitário $\hat{\mathbf{u}}$ e um ângulo θ , portanto a representação exige 4 parâmetros.

A concatenação ou composição de duas rotações representadas por ângulo-eixo se faz de maneira indireta. Convertem-se ambas para a forma matricial, multiplicam-se as matrizes e converte-se o produto à forma original ângulo-eixo. O custo deste procedimento é de: 58 multiplicações, 46 adições, 1 divisão e 6 chamadas a funções trigonométricas.

O rotacionamento de um vetor ou ponto do \mathbb{R}^3 é feito via Fórmula de Rodrigues a um custo de 31 operações, sendo 19 multiplicações e 12 adições.

Representação por Matrizes

A representação de rotações por matrizes 3×3 tem alto custo relativamente ao armazenamento. Exige 9 parâmetros ou 3 vetores de uma base ortonormal. A concatenação de duas rotações representadas por matrizes é realizada multiplicando-se as matrizes representantes. Este procedimento utiliza 27 multiplicações mais 18 adições. O custo total é de 45 operações.

O rotacionamento de um vetor é realizado pela multiplicação da matriz representante pela matriz das coordenadas do vetor. O custo é de 15 operações, sendo 9 multiplicações e 6 adições.

Representação por Ângulos de Euler

A representação por Ângulos de Euler, qualquer que seja a ordem dos eixos escolhida, utiliza 3 parâmetros, que são os ângulos θ_x, θ_y e θ_z .

A parametrização por ângulo de Euler além de ser ambígua e apresentar problemas do tipo "Gimbal Lock", conduz mal a concatenação de rotações. É feita por método indireto. Converte-se cada terna de ângulos representantes das rotações para o formato matricial, em seguida multiplicam-se as matrizes resultantes e converte-se a matriz produto ao formato original ângulos de Euler. Devido a problemas numéricos com esta última operação o resultado é aproximado. O custo computacional da concatenação nesta modalidade de representação é alto. São necessárias 34 multiplicações, 24 adições e 10 chamadas a funções trigonométricas.

O rotacionamento de um vetor ou ponto do \mathbb{R}^3 também não é eficiente. São necessárias 12 multiplicações, 6 adições e 6 chamadas a funções trigonométricas.

Parametrização por Quatérnios

A representação de rotações por quatérnios exige 4 parâmetros, um correspondente à parte real e 3 correspondente às componentes do vetor que constitui a parte imaginária do quatérnio.

A concatenação de duas rotações representadas por quatérnios, como no caso das matrizes, é realizada multiplicando-se os quatérnios representantes na ordem em que as rotações foram consideradas. O quatérnio produto será o parâmetro da rotação concatenada ou composta. O custo da concatenação de rotações com esta

representação é de 16 multiplicações e 12 adições.

O rotacionamento de um vetor ou ponto do \mathbb{R}^3 por uma rotação representada por um quatérnio é realizada como produto de três quatérnios. O custo deste procedimento é de 21 multiplicações e 12 adições; bem mais do que 9 multiplicações e 6 adições exigidos pela representação matricial, porém equivalente a 18 multiplicações e 12 adições exigidas pela representação ângulo-eixo via Fórmula de Rodrigues.

A tabela abaixo sintetiza nossa análise e nos leva a concluir que a representação quatérnica é a mais próxima da ideal.

	Formato da Representação			
Operação	Ângulo-eixo	Matricial	Ângulos de Euler	Quatérnica
Armazenamento	4	9	3	4
Concatenação	56M+46A+1D +6 Trig	27M+18A	34M+24A + 10 Trig	16M+12A
Rotacionamento	19M+12A	9M+6A	12M+6A+6Trig	21M+12A

Sobre Interpolação

A interpolação linear de rotações representadas por ângulo-eixo ou por ângulos de Euler, só produz resultados satisfatórios nos casos simples, em que as rotações a serem interpoladas ocorrem em torno de um mesmo eixo. No texto analisamos somente um dos casos devido a semelhança entre eles.

Nos casos em que as rotações a serem interpoladas linearmente, são especificadas por ângulos de Euler e não ocorrem em torno de um mesmo eixo, o resultado não é satisfatório. Por outro lado, rotações representadas por ângulo-eixo não são interpoláveis, isto é, a interpolação entre elas não está definida.

A interpolação linear de rotações representadas por matrizes ou por quatérnios, apresenta problemas semelhantes. Em geral as matrizes interpoladas não representam rotações e os quatérnios interpolados não são unitários. Estes problemas podem

ser contornados com procedimentos de reortogonalização e normalização que têm alto custo computacional.

Finalmente, a interpolação esférica de rotações representadas por quatérnios, apresenta um considerável ganho de qualidade visual e operacional, na medida em que todos os quatérnios parâmetros ficam sobre a esfera unitária e a curva que os liga tem velocidade constante. Na subseção anterior concluímos que quatérnios era a melhor opção para representar rotações relativamente aos aspectos de armazenamento, concatenação e rotacionamento. Podemos acrescentar agora que a representação quatérnica de rotações é também a melhor opção para produzir interpolações.

A APÊNDICE

Implementações

Nesta secção apresentaremos os procedimentos que implementam as principais funções da álgebra dos quatérnios usando a linguagem do sistema de computação algébrica **Maple**.

A.1 Adição de Quatérnios: $Ad(q,p)$

Ad:=proc(q,p):

q:p:

[q[1]+p[1],[q[2]+p[2],q[3]+p[3],q[4]+p[4]]]: end proc:

Exemplo: Ad([1,1,1,1],[2,2,2,2])=[3,[3,3,3]]

A.2 Multiplicação de Escalar por Quatérnio : $Me(s,q)$

Me:=proc(s,q):

*[s*q[1],[s*q[2],s*q[3],s*q[4]]]:end proc:*

Exemplo: Me(2,[2,3,2,2]);=[4, [6, 4, 4]]

A.3 Multiplicação de Quatérnios - Forma vetorial : $M(p,q)$

M:=proc(q,b):

q: p:

s:=[q[2],q[3],q[4]]: t:=[b[2],b[3],b[4]]:

*evalf([q[1]*b[1]-dotprod(s,t),evalm(q[1]*t+b[1]*s+crossprod(s,t))]):*

end proc:

Exemplo: M([2,1,1,1],[3,2,1,-1])=[4., [5., 8., 0.]]

A.4 Multiplicação de Quatérnios - Forma matricial:

$M^*(p,q)$

```

M*:=proc(p,q)
w:=p[1]:x:=p[2]: y:=p[3]:z:=p[4]: w*:=q[1]:x*:=q[2]:y*:=q[3]: z*:=q[4]:
A:=matrix(4,4,[w,x,y,z, -x,w,z,-y, -y,-z,w,x, -z,y,-x,w]):
J:=transpose(A):
X:=matrix(4,1,[w*,x*,y*,z*]); L:=evalm(J δ * X);
[L[1,1],[L[2,1], L[3,1], L[4,1]]]: end proc:
Exemplo: M*([2,1,1,1],[3,2,1,-1])=[4, [5, 8, 0]]

```

A.5 Conjugação de Quatérnios: Conj(q)

```

Conj:=proc(p):
[ p[1], [-p[2],-p[3],-p[4]] ] : end proc:
Exemplo: Conj([3,1,0,8])=[3, [-1, 0, -8]]

```

A.6 Norma: Nor(q)

```

Nor:=proc(p):
p[1]^2 + p[2]^2 + p[3]^2 + p[4]^2:end proc:
Exemplo: Nor([1,3,3,1])= 20

```

A.7 Módulo de um quatérnio: Mod(q)

```

Mod:=proc(p):
sqrt(p[1]^2 + p[2]^2 + p[3]^2 + p[4]^2):end proc:
Exemplo: Mod([1,1,1,1])=2

```


A.8 Inverso de um Quatérnio: Inv(q)

Nor:=proc(p):

p[1]^2 + p[2]^2 + p[3]^2 + p[4]^2 :end proc:

Inv:=proc(q):

Conj(q)/Nor(q): end proc:

Exemplo: Inv([1,1,1,1])= [1/4, [-1/4, -1/4, -1/4]]

A.9 Construindo o unitário de um Quatérnio

Unit:=proc(q):

q:

[q[1]/Mod(q),[q[2]/Mod(q),q[3]/Mod(q),q[4]/Mod(q)]:end proc:

Exemplo: Unit([1,1,1,1])= [1/2, [1/2, 1/2, 1/2]]

A.10 Forma Polar do Unitário de um Quatérnio: Fpl(q)

Fpl:=proc(q):

Unit:=proc(q):

q: [q[1]/Mod(q),[q[2]/Mod(q),q[3]/Mod(q),q[4]/Mod(q)]: end proc:

U:=Unit(q):

x:=arccos(U[1]):

theta:=evalf(convert(x,degrees),4)

if U[2]=[0,0,0] then y=1 else y := sqrt((U[2][1])^2 + (U[2][2])^2 + (U[2][3])^2) end if:

z:=[U[2][1]/y,U[2][2]/y,U[2][3]/y]:

*Exemplo: Fpl([4,4,4,4])= [cos(60.*degrees), sen(60.*degrees)[1/(3√3), 1/(3√3), 1/(3√3)]]*

B APÊNDICE

Implementações

B.1 Interpolação Linear de Matrizes - (Fig 1.4)

```

restart; with(linalg): with(plots):
W:=proc(t):
F:=[[2,0,0],[5,0,0],[5,0,5],[2,0,5 ]]:
A:=matrix(3,3,[1,0,0, 0,1,0, 0,0,1 ]):
B:=matrix(3,3,[0,-1,0, 1,0,0, 0,0,1]):
G:=t→ evalm((1-t)*A+t*B ):
polygonplot3d( [ seq( evalm( G(t)&*F[j]), j=1..4)]):
end proc:
a:=display(W(0), W(1)):
b:=display(seq( W(j/24),j=0..24), color=white):
display(b);

```

B.2 Interpolação Linear de Matrizes - (Fig 2.4)

```

restart;with(linalg): with(plots):
Lim:=proc(t):
A:=matrix(3,3,[1,0,0, 0,0,-1,0,1,0]):
B:=matrix(3,3,[0,1,0,1,0,0,0,0,-1]):
F:=[[2,0,0],[5,0,0],[5,0,5],[2,0,5 ]]:
J:=t→ evalf( (1-t)*A+t*B ):
a:=polygonplot3d( [ evalm(J(t)&*F[1]), evalm(J(t)&*F[2]),evalm(J(t)&*F[3]),
evalm(J(t)&*F[4])]):
display(a): end proc:
b:=display( seq( Lim( j*1/1 ),j=0..1 ) , color=yellow ):

```

```
display(b):
display(b, seq( Lim( j*1/24 ),j=0..24 ),color=white );
```

B.3 Interpolação Linear de ângulos de Euler - (Fig 3.4)

```
restart;with(linalg): with(plots):
MR:=proc(theta1,theta2,theta3): R3:=matrix(3,3,[Cz,-Sz,0,Sz,Cz,0,0,0,1]):
R2:=matrix(3,3,[Cy,0,Sy,0,1,0,-y,0,Cy]):
R1:=matrix(3,3,[1,0, 0, 0, Cx,-Sx,0,Sx,Cx]):
Sx:=sin(theta1):Cx:=cos(theta1):Sy:=sin(theta2):
Cy:=cos(theta2):Sz:=sin(theta3): z:=cos(theta3):
M:= evalm(R1&*R2&*R3): end proc:
LinEuler:=proc(P,Q,t):
F:=[[1,1,1],[2,2,4],[3,3,1]]:
J:=t→ evalf( (1-t)*[P[1],P[2],P[3]]+t*[Q[1],Q[2],Q[3]] ,4):
a:=polygonplot3d( [ evalm(MR(J(t)[1],J(t)[2],J(t)[3])&*F[1]),
evalm(MR(J(t)[1],J(t)[2],J(t)[3])&*F[2]),
evalm(MR(J(t)[1],J(t)[2],J(t)[3])&*F[3]) ]):
display(a): end proc: b:=LinEuler( [0,0,0],[0,0,2*Pi],0 ):
display(b, seq( LinEuler( [0,0,0],[2*Pi,0,0],j*1/36 ),j=0..36));
```

B.4 Interpolação Linear de ângulos de Euler - (Fig 4.4)

```
restart;with(linalg): with(plots):
MR:=proc(theta1,theta2,theta3):
R3:=matrix(3,3,[Cz,-Sz,0,Sz,Cz,0,0,0,1]):
R2:=matrix(3,3,[Cy,0,Sy,0,1,0,-Sy,0,Cy]):
R1:=matrix(3,3,[1,0, 0, 0, Cx,-Sx,0,Sx,Cx]):
Sx:=sin(theta1):Cx:=cos(theta1):Sy:=sin(theta2):
```

```

Cy:=cos(theta2):Sz:=sin(theta3): Cz:=cos(theta3):
M:= evalm(R1 &*R2 &* R3): end proc:
Lim:=proc(P,Q,t):
F:=[[2,0,0],[5,0,0],[5,0,5],[2,0,5 ]]:
J:=t ↦ evalf( (1-t)*[P[1],P[2],P[3]]+t*[Q[1],Q[2],Q[3]] ):
U:=MR(J(t)[1],J(t)[2],J(t)[3]):
a:=polygonplot3d( [ evalm(U &*F[1]), evalm(U&*F[2]),
evalm(U &*F[3]),evalm(U &*F[4])] ):
display(a): end proc:
h:=display( seq( Lim( [Pi/2,0,-Pi/2],[0,Pi/4,Pi/2],j ),j=0..1),
view=[-7..7,-7..7,-7..7],color=yellow);
k:=display( seq( Lim( [Pi/2,0,-Pi/2],[0,Pi/4,Pi/2],j/24 ),j=0..24), color=white):
display(k);

```

B.5 Interpolação Linear de Quatérnios - (Fig 5.4)

```

restart; with(linalg): with(plots):
Linq:=proc(q1,q2,t):
F:=[[2,0,0],[5,0,0],[5,0,5],[2,0,5 ]]:
J:=t↦ (1-t)*q1+t*q2:
x:=evalf(J(t)[1]):y:=evalf(J(t)[2]):z:=evalf(J(t)[3]):w:=evalf(J(t)[4]):
M:=matrix(3,3,[1-2*(y^2+z^2), 2*x*y-2*z*w,2*x*z+2*y*w, 2*x*y+2*z*w,
1-2*(x^2+z^2), 2*y*z-2*x*w,2*x*z-2*y*w,2*y*z+2*x*w,1-2*(y^2+x^2)]):
a:=polygonplot3d([ evalm(M &*F[1]),evalm(M&*F[2]),
evalm(M&*F[3]),evalm(M&*F[4]) ] ):
end proc:
m1:=(Linq([1/2, 1/2,-1/2,1/2], [.2705980500, .2705980500, .6532814824,
.6532814824],0), color=red):

```

```

m2:=(Linq([1/2, 1/2,-1/2,1/2],[.2705980500, .2705980500, .6532814824,
.6532814824],1),
color=yellow):
m3:=display(seq( Linq([1/2, 1/2,-1/2,1/2],[.2705980500, .2705980500, .6532814824,
.6532814824],j*1/24),j=0..24), color=white ): display(m1,m2); display(m1,m2,m3);

```

B.6 Interpolação Esférica de Quatérnios - Fig(6.4)

```

with(linalg): with(plots):
Sl:=proc(q1,q2,t)
theta:=angle(q1,q2):
F:=[[2,0,0],[5,0,0],[5,0,5],[2,0,5 ]]:
J:=t→ evalf( (sin((1-t)*theta)/sin(theta) ) *q1+sin(t*theta)/sin(theta)*q2 ):
x:=evalf(J(t)[1]):y:=evalf(J(t)[2]):z:=evalf(J(t)[3]):w:=evalf(J(t)[4]):
M:=matrix(3,3,[1-2*(y^2+z^2), 2*x*y-2*z*w,2*x*z+2*y*w,
2*x*y+2*z*w, 1-2*(x^2+z^2),
2*y*z-2*x*w,2*x*z-2*y*w,2*y*z+2*x*w,1-2*(y^2+x^2)]):
a:=polygonplot3d([ evalm(M&*F[1]),evalm(M&*F[2]),
evalm(M&*F[3]),evalm(M&*F[4] ) ]):
end proc:
m1:=Sl([1/2, 1/2,-1/2,1/2],[.2705980500, .2705980500, .6532814824, .6532814824],0):
m2:=Sl([1/2, 1/2,-1/2,1/2],[.2705980500, .2705980500, .6532814824, .6532814824],1):
m3:=display( seq(Sl([1/2, 1/2,-1/2,1/2],[.2705980500, .2705980500, .6532814824,
.6532814824],j*1/24),j=0..24), color=white ): display(m1,m2); display(m3);

```

Referências

- [1] DAM,Erik; KOCH,M; LILLHOLM,M. **Quaternions interpolation an Animation.** Technical Report DIKU-TR-98/5, University of Copenhagen, Denmark, 1998.
- [2] SHOEMAKE,K. **Animating Rotation whith Quaternions Curves.** Computer Graphics Proc.do SIGGRAPH'85, vol.19, n.3, 1985, pp245-254.
- [3] GOMES,Jonas; VELHO,Luiz. **Computação Gráfica.:** IMPA,vol.1, 1998.
- [4] VERTH;James M.van; BISHOP, Lars M. **Essential Mathematics for Games and Interactive Applications.** Morgan Kaufmman, San Francisco, 2004.