

M. Eng. Fernando Henrique Gomes Zucatelli
Gabriela Bittencourt

Apostila de Quatérnios e Rotações em 3D com
Matlab®

Nota dos autores

Essa apostila foi elaborada a partir de um desejo meu, Gabriela, em saber mais sobre quatérnios e suas aplicações em quadricópteros. A partir da vontade da Gabriela, eu, Fernando, comecei a reunir material sobre quatérnios e suas aplicações em rotações em três dimensões.

As anotações feitas, os códigos criados e as figuras rascunhadas estão reunidos nesta apostila. As explicações apresentadas são aquelas que nos fizeram compreender os problemas apresentados e os exercícios propostos são aqueles que fixaram o conteúdo novo ou exploraram noções mais básicas, como as de álgebra linear, que se aplicam nos estudos de quatérnios.

A ideia dessa apostila é apresentar os quatérnios como um novo tipo de número, abrindo a mente dos leitores para um novo mundo de tipos diferentes de números por assim dizer, que expandem as noções básicas aprendidas até o início do curso superior, em resumo, dos naturais aos complexos. Uma segunda ideia desenvolvida na apostila é apresentar o software Matlab[®] como ferramenta para se executar cálculos com os quatérnios, de tal forma a criar toda uma biblioteca de funções de quatérnios e rotações tridimensionais. A criação dessa biblioteca contribui para que o estudante entenda que pode compartimentalizar seus códigos e acessá-los conforme a necessidade. Com essa segunda ideia em mente, fornecemos todos os códigos usados nas figuras geradas em Matlab[®].

Talvez para um especialista em quatérnios ou em dinâmica de rotações espaciais os tópicos abordados sejam simples demais. Talvez para um especialista em Matlab[®] os códigos feitos sejam simples demais. Porém, para um leitor que esteja se familiarizando com um tema ou com o outro, talvez seja essa apostila a reunião ideal dos conceitos de forma simples para um aprendizado eficiente.

Diante disso, mantivemos ao longo do texto diversas sugestões de materiais complementares, artigos e aulas sobre o assunto, além de curiosidades que circundam alguns dos tópicos abordados como forma de despertar o interesse dos leitores mostrando uma fração de um novo universo de possibilidades de forma atrativa e surpreendente, muitos desses comentários estão em notas de rodapé¹ para não interferirem com o texto principal.

¹Nós, os autores, gostamos de notas de rodapé.

Sumário

1	Aquecimento - Sistemas de Coordenadas	1
1.1	Coordenadas polares e cartesianas	1
1.2	Coordenadas esféricas e cartesianas	2
2	Introdução à Álgebra de Quatérnios	4
2.1	Revisão de conjuntos numéricos	4
2.2	Definição dos quatérnios	5
2.3	Operações com quatérnios	7
2.3.1	Parte escalar de um quatérnio	7
2.3.2	Parte vetorial de um quatérnio	7
2.3.3	Soma	7
2.3.4	Multiplicação por escalar	8
2.3.5	Notação de quatérnio com parte escalar e vetorial	8
2.3.6	Conjugado	9
2.3.7	Módulo	9
2.3.8	Multiplicação entre quatérnios	10
2.3.9	Divisão entre quatérnios	11
3	Quatérnios como “complexo de complexo”	13
3.1	Isomorfismo entre quatérnios, plano complexo e espaço real de 4 dimensões	13
3.2	Representação de quatérnios com matrizes	13
4	Rotações	16
4.1	Rotações 2D	16
4.2	Rotações 3D - Ângulos de Euler	16
4.2.1	Visualizações e Exemplos de Rotações 3D com Ângulos de Euler	19
4.3	Rotações 3D - Teorema de Euler	23
4.3.1	Teorema de Euler	23
4.4	Rotações 3D - <i>Fórmula de Rodrigues</i>	24
4.4.1	Entendendo o grupo de rotação $SO(3)$	28
4.5	Rotações 2D - Fórmula de Euler	28
4.6	Rotações 3D - Fórmula de Euler para Quatérnios	29
4.7	Rotações 3D - <i>Fórmula de Rodrigues</i> para Quatérnios	30
A	Resoluções de Exercícios Propostos	35
A.1	Exercício 1	35
A.2	Exercício 2	35
A.3	Exercício 3	35
A.4	Exercício 6	36
A.5	Exercício 7	36
A.6	Exercício 8	36
A.7	Exercício 10	37
A.8	Exercício 11	37
A.9	Exercício 12	37
A.10	Exercício 13	37

A.11 Exercício 16	38
A.12 Exercício 17	38
A.13 Exercícios 32, 33, 34, 35	38
A.13.1 Código de resolução	39
A.14 Exercício 40	39
A.15 Exercício 41	40
A.16 Exercício 44	40
A.17 Exercício 50	40
A.18 Exercício 56	41
A.19 Exercício 58	41
A.20 Exercício 59	41
B Normas	43
C Criando sua própria biblioteca em Matlab[®]	47
D Códigos em Matlab[®] para Álgebra de Quatérnios	48
D.1 Códigos com álgebra de quatérnios	48
D.2 Códigos com <i>plot</i> de coordenadas	50
D.3 Códigos com matrizes de rotação	53
D.4 Códigos auxiliares	55
D.5 Códigos geradores de Figuras	55
E Outros trabalhos e bibliotecas sobre Quatérnios em Matlab[®]	64

Lista de Figuras

1.1	Vetor \vec{v} em duas dimensões com tamanho ρ . Elaborado em Matlab®.	1
1.2	Vetor \vec{v} em três dimensões com tamanho ρ . Elaborado em Matlab®.	2
2.1	Diagrama de Venn dos principais subconjuntos numéricos com complexos.	4
2.2	Placa na ponte do <i>Canal Royal de Dublin</i> homenageando Hamilton.	6
3.1	Isomorfismos $\mathbb{C} \cong \mathbb{R}^2$ e entre $\mathbb{C}^2 \cong \mathbb{R}^4$. \mathbb{C}^2 pode ser interpretado como um plano em que cada eixo é um plano complexo \mathbb{C} .	14
4.1	Exemplo de rotação em 2D com uso de números complexos.	17
4.2	Definição matriz de rotação R_x .	17
4.3	Definição matriz de rotação R_y .	18
4.4	Definição matriz de rotação R_z .	18
4.5	Três passos da rotação XYZ : $\phi_X = 90^\circ$; $\theta_Y = 90^\circ$; $\psi_Z = 90^\circ$.	20
4.6	Três passos da rotação XYZ : $\phi_X = 45^\circ$; $\theta_Y = 90^\circ$; $\psi_Z = 45^\circ$.	20
4.7	Três passos da rotação XZY : $\phi_X = 90^\circ$; $\theta_Z = 90^\circ$; $\psi_Y = 90^\circ$.	21
4.8	Três passos da rotação YXZ : $\phi_X = 90^\circ$; $\theta_Y = 90^\circ$; $\psi_Z = 90^\circ$.	21
4.9	Três passos da rotação ZYX : $\phi_Z = 90^\circ$; $\theta_Y = 90^\circ$; $\psi_X = 90^\circ$.	22
4.10	Três passos da rotação ZYZ : $\phi_Z = 90^\circ$; $\theta_Y = 90^\circ$; $\psi_Z = 90^\circ$.	22
4.11	Ilustração do Teorema de Euler.	23
4.12	Rotação do vetor \vec{v} , decomposto em suas projeções ortogonal e perpendicular ao vetor \hat{n} , cujo resultado é o vetor \vec{v}' . Repare que as projeções paralelas são iguais antes e após a rotação ao redor de \hat{n} .	25
4.13	Representação do grupo $SO(3)$ como esfera de raio π .	28
4.14	Resultado da rotação de \hat{e}_3 ao redor de \hat{e}_2 por um ângulo de $\frac{\pi}{2}$ rad.	31
B.1	Desigualdade triangular.	43
B.2	Ilustração dos casos com $\ \cdot\ _p = 1$ para as normas $p = \{1, 2, \infty\}$.	44
B.3	Exemplo de normas.	44
C.1	Adicionando pastas e subpastas para criar biblioteca do usuário.	47
C.2	Chamando funções a partir da biblioteca.	47

Lista de Algoritmos

2.1	Exemplo de extração da parte escalar de quatérnios em Matlab [®]	7
2.2	Exemplo de extração da parte vetorial de quatérnios em Matlab [®]	7
2.3	Exemplo soma de quatérnios em Matlab [®]	8
2.4	Exemplo produto por escalar de quatérnios em Matlab [®]	8
2.5	Exemplo de soma de escalar com vetor no Matlab [®]	8
3.1	Exemplo do isomorfismo descrito pela Eq. (3.6).	14
3.2	Exemplo do isomorfismo descrito pela Eq. (3.9).	15
4.1	Exemplo de teste básico de tempo de execução em Matlab [®]	19
A.1	Resolução dos Exercícios 34, 33, 34 e 35 com uso de simbólico no Matlab [®] e integração com L ^A T _E X	39
A.2	Código de resolução com Matlab [®] simbólico do Exercício 50.	41
D.1	Rotação de um vetor por um ângulo com quatérnios.	48
D.2	Conjugado de quatérnio.	48
D.3	Divisão de quatérnio à direita.	48
D.4	Divisão de quatérnio à esquerda.	48
D.5	Inverso de quatérnio.	48
D.6	Módulo de quatérnio.	49
D.7	Exponencial de quatérnio.	49
D.8	Plot coordenadas 3D com quatérnios.	49
D.9	Produto de quatérnios.	50
D.10	Rotação de quatérnios.	50
D.11	Plot de coordenadas 2D.	50
D.12	Plot de coordenadas 3D.	50
D.13	Plot setas 2D.	51
D.14	Plot setas 3D.	52
D.15	Plot de vetor em 3D com identificação.	52
D.16	Rotação Fórmula de Rodrigues.	53
D.17	Matriz de rotação sobre eixo X	53
D.18	Matriz de rotação sobre eixo Y	53
D.19	Matriz de rotação sobre eixo Z	53
D.20	Matriz de rotação na sequência XYZ	53
D.21	Matriz de rotação na sequência XZY	53
D.22	Matriz de rotação na sequência YXZ	54
D.23	Matriz de rotação na sequência YZX	54
D.24	Matriz de rotação na sequência ZXY	54
D.25	Matriz de rotação na sequência ZYX	54
D.26	Matriz de rotação na sequência ZXZ	54
D.27	Matriz de rotação na sequência ZYZ	54
D.28	Encapsulamento da função que gera arquivos em L ^A T _E X a partir de uma <i>string</i>	55
D.29	Nomenclatura e padronização dos rótulos dos eixos XYZ	55
D.30	Figura do exemplo de coordenadas 2D em Matlab [®]	55
D.31	Inserção da ponta da seta em um ponto qualquer em 2D.	56
D.32	Figura do exemplo de coordenadas 3D em Matlab [®]	56
D.33	Inserção da ponta da seta em um ponto qualquer em 3D.	57
D.34	Figura do isomorfismo $\mathbb{R}^2 \cong \mathbb{C}$	57

D.35 Figura do isomorfismo $\mathbb{R}^4 \cong \mathbb{C}^2$ 58

D.36 Figura do exemplo de rotação 2D em Matlab[®]. 58

D.37 Figura sobre o Teorema de Euler. 60

D.38 Figura do exemplo de rotação 3D em Matlab[®]. 60

D.39 Geração dos exemplos de 3 a 8. 62

Capítulo 1

Aquecimento - Sistemas de Coordenadas

1.1 Coordenadas polares e cartesianas

A descrição da localização de um ponto em um plano qualquer pode ser feita em diversos sistemas de coordenadas. Aqui apresentamos os dois mais comuns. As coordenadas cartesianas, pelo par (x, y) , e as coordenadas polares, representadas pelo par (ρ, θ) .

A notação com uso explícito das bases C , Cartesiana, e P , Polar.

$$\vec{v} = (x_1 - x_0, y_1 - y_0)_C \equiv (\rho, \theta)_P \quad (1.1)$$

Nesta apostila também será usada a notação fasorial para a forma polar: $\rho \angle \theta$.

A Figura 1.1 ilustra as relações entre esses dois sistemas de coordenadas.¹

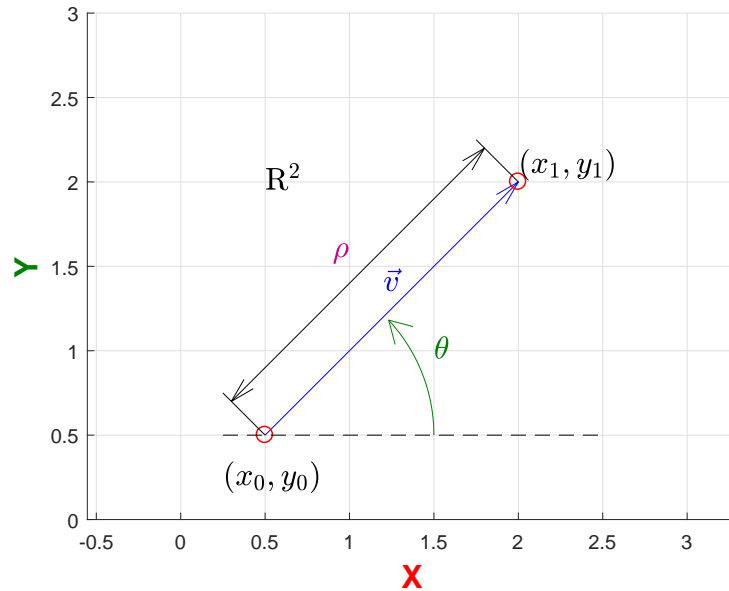


Figura 1.1: Vetor \vec{v} em duas dimensões com tamanho ρ . Elaborado em Matlab[®].

A transformação de coordenadas polares para cartesianas é dada por

$$\begin{cases} x = \rho \cos(\theta) \\ y = \rho \sin(\theta) \end{cases} \quad (1.2)$$

¹O código gerador da figura encontra-se no Apêndice D.5, Algoritmo D.30.

e a de coordenadas cartesianas para polares por

$$\begin{cases} \rho = \sqrt{x^2 + y^2} \\ \theta = \tan^{-1} \left(\frac{y}{x} \right) = \text{atan2}(y, x) \end{cases} \quad (1.3)$$

a função² (ou comando) `atan2`³ é conhecida como e seu resultado retorna um ângulo no intervalo $[-\pi, +\pi]$.

Código 1. Faça um código para realizar a conversão de coordenadas polares para cartesianas:

`[x,y] = fc_polar_para_cartesiana(ρ, θ)`

Código 2. Faça um código para realizar a conversão de coordenadas cartesianas para polares:

`[ρ, θ] = fc_cartesiana_para_polar(x, y)`

Código 3. Faça um código que receba vetores contendo x_0, y_0, ρ e θ , gere o gráfico da trajetória descrita pelo par (x_0, y_0) e a cada par de coordenadas desenha a seta dada pelo par (ρ, θ) :

`fc_setas_2D(x0, y0, ρ, θ)`

Compare os resultados com o comando **quiver** do Matlab[®].

1.2 Coordenadas esféricas e cartesianas

As principais coordenadas usadas em espaço de 3 dimensões são: cartesianas, cilíndricas e esféricas. Nesta apostila não serão abordadas as coordenadas cilíndricas.

As coordenadas cartesianas, pela tripla (x, y, z) , e as coordenadas esféricas, representadas pela tripla (ρ, θ, φ) .

Aqui destaca-se que há diferenças entre físicos e matemáticos, estadunidenses e europeus, no uso das letras para indicar os ângulos (WEISSTEIN, 2015c).

A notação com uso explícito das bases C , Cartesiana, e E , Esféricas.

$$\vec{v} = (x_1 - x_0, y_1 - y_0, z_1 - z_0)_C \equiv (\rho, \theta, \varphi)_E \quad (1.4)$$

A Figura 1.2 ilustra as relações entre esses dois sistemas de coordenadas.⁴

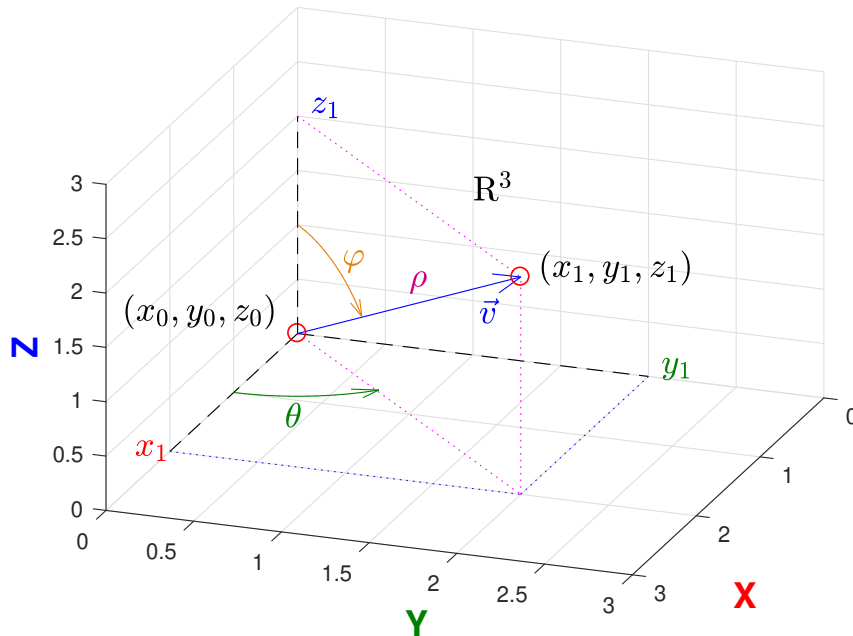


Figura 1.2: Vetor \vec{v} em três dimensões com tamanho ρ . Elaborado em Matlab[®].

²Neste apostila as funções nos exemplos terão o prefixo “fc.lib_” por pertencerem à biblioteca (*library*) pessoal dos autores. As funções solicitadas como exercícios não contém o “lib_”.

³Verifique no Matlab[®]: “help atan2”.

⁴O código gerador da figura encontra-se no Apêndice D.5, Algoritmo D.32.

Nesta apostila usaremos a notação normalmente adotada por matemáticos com ρ para a grandeza radial, θ para o ângulo azimutal e φ para o ângulo polar, também conhecido como elevação.

A transformação de coordenadas esféricas para cartesianas é dada por

$$\begin{cases} x = \rho \cos(\theta) \sin(\varphi) \\ y = \rho \sin(\theta) \sin(\varphi) \\ z = \rho \cos(\varphi) \end{cases} \quad (1.5)$$

e a de coordenadas cartesianas para esféricas por

$$\begin{cases} \rho = \sqrt{x^2 + y^2 + z^2} \\ \theta = \tan^{-1}\left(\frac{y}{x}\right) = \text{atan2}(y, x) \\ \varphi = \cos^{-1}\left(\frac{z}{\rho}\right) \end{cases} \quad (1.6)$$

e os limites de cada variável são

$$0 \leq \rho < \infty; \quad 0 \leq \theta \leq 2\pi; \quad 0 \leq \varphi < \pi \quad (1.7)$$

Código 4. Faça um código para realizar a conversão de coordenadas esféricas para cartesianas:

`[x,y,z] = fc_esfericas_par_cartesiana(ρ, θ, φ)`

Código 5. Faça um código para realizar a conversão de coordenadas cartesianas para esféricas:

`[ρ, θ, φ] = fc_cartesiana_par_esfericas(x, y, z)`

Código 6. Faça um código que receba vetores contendo $x_0, y_0, z_0, \rho, \theta$ e φ , gere o gráfico da trajetória descrita pelo par (x_0, y_0, z_0) e a cada par de coordenadas desenha a seta dada pelo par (ρ, θ, φ) :

`fc_setas_3D($x_0, y_0, z_0, \rho, \theta, \varphi$)`

Compare os resultados com o comando **quiver3** do Matlab[®]. **Dica:** “*help plot*”.

Capítulo 2

Introdução à Álgebra de Quatérnios

2.1 Revisão de conjuntos numéricos

Iniciemos relembando as relações dos conjuntos numéricos até os complexos

$$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C} \subset \dots \quad (2.1)$$

Uma aula muito boa e fortemente recomendada sobre os conjuntos numéricos encontra-se em (PARKER, 2019). De forma simplificada, tem-se o conjunto dos naturais \mathbb{N} , que são basicamente a ideia central da contagem; os inteiros \mathbb{Z} , que adicionam os negativos; os racionais \mathbb{Q} , que acrescem a divisão entre inteiros; os reais \mathbb{R} , em que o número já não é mais descrito como uma forma racional e os complexos \mathbb{C} , que expandem o conjunto dos reais para conter todas as soluções de equações polinomiais conforme descrito pelo Teorema Fundamental da Álgebra (Gauss, 1816) (TOLEDO, 2017; SALVADO, 2016), o qual prova que para todo polinômio

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \quad (2.2)$$

com $a_i \in \mathbb{R}$ e $\text{grau}(p) = n \geq 1, n \in \mathbb{N}$, as raízes encontram-se **sempre** no conjunto dos complexos \mathbb{C} .

A Figura 2.1 apresenta um diagrama muito útil para compreender as relações entre esses conjuntos numéricos.¹

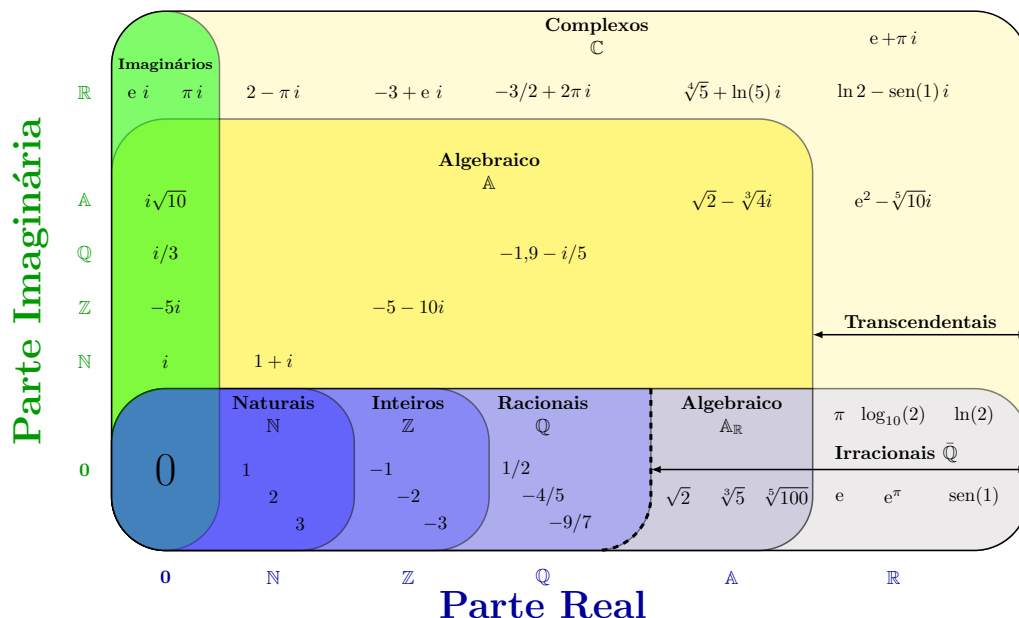


Figura 2.1: Diagrama de Venn dos principais subconjuntos numéricos com complexos.

Fonte: Traduzido e adaptado de (TULEY, 2017).

¹Os autores reconhecem o debate sobre o zero ser ou não natural. Como este trabalho envolve o uso de álgebra de espaços vetoriais preferimos então considerar o zero como natural para todos os efeitos.

Na [Figura 2.1](#) foram acrescentados dois conjuntos em relação aos apresentados em (2.1) com características interessantes, a denominar os *algebraicos* \mathbb{A} e o seu complemento em relação aos complexos, os *transcendentais* ([PAMPENA, 2013](#)). Os algebraicos são todos os números que são soluções de (2.1) com a condição de $a_i \in \mathbb{Q}$, ou ainda, apenas com $a_i \in \mathbb{Z}$, dado que é sempre possível reescrever os coeficientes racionais em uma forma inteira por meio de um múltiplo comum. Os números que não podem ser obtidos com essa restrição são os denominados transcendentais ([WEISSTEIN, 2015a](#); [FISCHLER, 2000](#); [WIKIPÉDIA, 2018a](#)).

Esse detalhe é particularmente interessante por mostrar que os números até o conjunto dos algebraicos estão associados com problemas elementares de matemática e, em especial, com resolução de equações polinomiais. Reais e complexos estendem essa fronteira mas ainda se relacionam com resoluções de equações. Os quatérnios estão além disso.

Os quatérnios, denotados por \mathbb{H} , são um tipo de número dos hipercomplexos. Os hipercomplexos extrapolam a noção dos complexos, não são mais associados à números obtidos com soluções de equações, sejam polinomiais ou não.

Há ao menos duas definições para os hipercomplexos, (i) eles são números de altas dimensões que não necessariamente compartilham propriedades dos reais e dos complexos porém, são construídos a partir deste e, (ii) são números cujas propriedades divergem das propriedades dos reais e complexos. Além dos quatérnios, são exemplos de hipercomplexos: bi-quatérnios, álgebras exteriores, álgebras de grupo, matrizes e octônios ([WEISSTEIN, 2015b](#)).

Além dos conjuntos apresentados na Equação (2.1) e os hipercomplexos há os seguintes conjuntos de números que lidam com problemas específicos da matemática. Ao lado de cada conjunto abaixo encontram-se vídeos didáticos e introdutórios aos temas como uma forma de primeiro contato *amigável* com o assunto.

- p-adic ([SANDERSON, 2015](#))
- Números transfinitos ([STEVENS, 2016](#))
- Surreais ([WIKIPÉDIA, 2019e](#))
- Super-reais ([WIKIPÉDIA, 2019a](#))

Na elaboração dessa apostila, foram utilizados diversos materiais citados entre os quais: ([ROSA, 2011](#); [CONCI, 2014](#); [BIASI, 2002](#); [BUCHMANN, 2011](#)).

Também foram utilizadas algumas videoaulas e tópicos de curiosidades sobre quatérnios citadas a seguir: ([ACADEMICS, 2014](#); [ROBOTICS, 2017a](#); [ROBOTICS, 2017b](#); [JACKSON, 2017](#); [MATHOMA, 2015](#); [MATHOMA, 2016](#)).

Destacamos os vídeos **extremamente recomendados** do canal de YouTube® **3Blue1Brown** ([SANDERSON, 2018a](#); [SANDERSON, 2018b](#)) que explicam sobre quatérnios do ponto de vista geométrico, apresentando, através de belas visualizações e didática ímpar, diversas noções de como as operações com quatérnios se relacionam com o mundo.

2.2 Definição dos quatérnios

Seja $q \in \mathbb{H}$ um quatérnio dado por

$$q = a + xi + yj + zk \quad (2.3)$$

onde $a, x, y, z \in \mathbb{R}$, equivalentemente utilizando a notação com elementos de uma base vetorial

$$q = q_0 e_0 + q_1 e_1 + q_2 e_2 + q_3 e_3 \quad (2.4)$$

onde $q_0, q_1, q_2, q_3 \in \mathbb{R}$, $e_0 = 1$ e $e_1 = i, e_2 = j, e_3 = k$ são unidades imaginárias, i.e.

$$e_1^2 = i^2 = -1; \quad e_2^2 = j^2 = -1; \quad e_3^2 = k^2 = -1 \quad (2.5)$$

e acresce-se a seguinte regra

$$e_1 e_2 e_3 = ijk = -1; \quad (2.6)$$

tal que a combinação das Equações (2.5) e (2.6) definem de forma mínima e precisa todas as propriedades do conjunto \mathbb{H} conforme W. R. Hamilton² formulou em 1844 (HAMILTON, 1844).

Hamilton chegou à solução para multiplicação de quatérnios enquanto passeava pela ponte do *Canal Royal de Dublin*. Orgulhoso de sua descoberta, Hamilton interrompeu sua caminhada para cravar em uma pedra da ponte a simples e elegante solução, posteriormente foi feita uma placa sobre o feito apresentada na Figura 2.2.

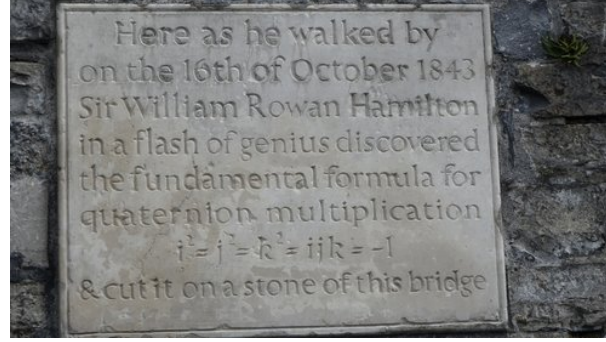


Figura 2.2: Placa na ponte do *Canal Royal de Dublin* homenageando Hamilton. Na placa lê-se: “Here as he walked by on the 16th of October 1843 Sir William Rowan Hamilton in a flash of genius discovered the fundamental formula for quaternion multiplication $i^2 = j^2 = k^2 = ijk = -1$ & cut it on a stone of this bridge.”

As notações estabelecidas nas Equações (2.3) e (2.4) serão usadas ao longo da apostila sendo aplicadas conforme a necessidade.

Usaremos ao longo da apostila extensivamente a notação com os elementos de base $e_{(\cdot)}$ por facilitar o entendimento das operações entre as unidades imaginárias como operações entre os elementos de base do espaço vetorial \mathbb{R}^3 . A lógica por trás dela segue a lógica do épsilon de *Levi-Civita* (ε_{ijk}) comumente utilizado no produto vetorial (WIKIPÉDIA, 2019b; GUIO, 2011).

$$\varepsilon_{ijk} = \begin{cases} +1 & , \text{ se } (i, j, k) = \{(1, 2, 3), (2, 3, 1), (3, 1, 2)\} \\ -1 & , \text{ se } (i, j, k) = \{(1, 3, 2), (2, 1, 3), (3, 2, 1)\} \\ 0 & , \text{ c.c. (caso contrário)} \end{cases} \quad (2.7)$$

Repare para o valor positivo estamos rotacionando a posição dos índices (1, 2, 3) para a esquerda. A rotação para a direita produziria o mesmo resultado porém na ordem inversa. O valor negativo é obtido por meio da comutação entre dois índices. Em particular, cada comutação entre dois índices implica na inversão do sinal.

A partir da Eq. (2.7) construímos as relações entre os produtos das unidades imaginárias com os elementos de base, ou seja, $(i, j, k) \leftrightarrow (e_1, e_2, e_3)$.

$$\begin{aligned} & \begin{cases} e_1 e_2 = e_3 \\ ij = k \end{cases}; \quad \begin{cases} e_2 e_3 = e_1 \\ jk = i \end{cases}; \quad \begin{cases} e_3 e_1 = e_2 \\ ki = j \end{cases} \\ & \begin{cases} e_2 e_1 = -e_3 \\ ji = -k \end{cases}; \quad \begin{cases} e_3 e_2 = -e_1 \\ kj = -i \end{cases}; \quad \begin{cases} e_1 e_3 = -e_2 \\ ik = -j \end{cases} \end{aligned} \quad (2.8)$$

Repare atentamente na ordem em que os índices de $e_{(\cdot)}$ foram escritos. A ordem concorda perfeitamente com a descrita em (2.7).

A partir desses resultados elaboramos a Tabela 2.1 (ROSA, 2011; CONCI, 2014) com o produto dos elementos da base.

Exercício 1. *Mostre que $ijk = -1$.*

Exercício 2. *Calcule (a) jki , (b) kij .*

²É o mesmo Hamilton da função hamiltoniana de energia.

Tabela 2.1: Produtos entre os elementos base do espaço dos quatérnios \mathbb{H}

	$e_0 = 1$	$e_1 = i$	$e_2 = j$	$e_3 = k$
$1 = e_0$	1	i	j	k
$i = e_1$	i	-1	k	$-j$
$j = e_2$	j	$-k$	-1	i
$k = e_3$	k	j	$-i$	-1

2.3 Operações com quatérnios

2.3.1 Parte escalar de um quatérnio

A parte escalar de um quatérnio é o termo que não está associado às unidades imaginárias.

$$S(q) = S(a + xi + yj + zk) = a \in \mathbb{R} \quad (2.9)$$

Código 7. No Matlab[®] usaremos o quatérnio q escrito como elemento do \mathbb{R}^4 . A validade dessa relação será detalhada no [Capítulo 3](#).

Em Matlab[®], basta extrair a primeira posição do elemento conforme o exemplo:

Algoritmo 2.1: Exemplo de extração da parte escalar de quatérnios em Matlab[®]

```
1 q = [10, 20, 30, 40];
2 a = q(1)
```

cujo resultado é:

a =

10

2.3.2 Parte vetorial de um quatérnio

A parte vetorial de um quatérnio é composta pelo vetor associado às unidades imaginárias.

$$V(q) = V(a + xi + yj + zk) = xi + yj + zk = \vec{v} \in \mathbb{R}^3 \quad (2.10)$$

Código 8. Em Matlab[®], basta extrair o vetor composto da segunda à quarta posição do elemento conforme o exemplo:

Algoritmo 2.2: Exemplo de extração da parte vetorial de quatérnios em Matlab[®]

```
1 q = [10, 20, 30, 40];
2 v = q(2:4)
```

cujo resultado é:

v =

20 30 40

2.3.3 Soma

Sejam $p, q \in \mathbb{H}$, a soma de dois quatérnios é definida como a soma sobre cada componente

$$\begin{aligned} q &= q_0 e_0 + q_1 e_1 + q_2 e_2 + q_3 e_3 \\ p &= p_0 e_0 + p_1 e_1 + p_2 e_2 + p_3 e_3 \\ (q \pm p) &= (q_0 \pm p_0) e_0 + (q_1 \pm p_1) e_1 + (q_2 \pm p_2) e_2 + (q_3 \pm p_3) e_3 \end{aligned} \quad (2.11)$$

Código 9. A operação de soma se preserva entre os quatérnios e o \mathbb{R}^4 conforme segue exemplo em Matlab[®]

Algoritmo 2.3: Exemplo soma de quatérnios em Matlab[®]

```
1 q = [1, 2, 3, 4];
2 p = [5, 6, 7, 8];
3 r = q + p
```

cujo resultado é:

r =

6 8 10 12

2.3.4 Multiplicação por escalar

Seja $c \in \mathbb{K} = \mathbb{R}$, o produto por escalar é definido.³

$$p = cq = (cq_0)e_0 + (cq_1)e_1 + (cq_2)e_2 + (cq_3)e_3 \quad (2.12)$$

Código 10. A operação de produto por escalar se preserva entre os quatérnios e o \mathbb{R}^4 conforme segue exemplo em Matlab[®]

Algoritmo 2.4: Exemplo produto por escalar de quatérnios em Matlab[®]

```
1 q = [1, 2, 3, 4];
2 c = 5;
3 p = c*q
```

cujo resultado é:

p =

5 10 15 20

2.3.5 Notação de quatérnio com parte escalar e vetorial

A partir das notações das partes escalares e reais é comum o uso da seguinte notação para quatérnios:

$$q = a + \vec{v} \quad (2.13)$$

essa notação faz uso do fato dos quatérnios serem formados pela soma de uma parte escalar com uma parte vetorial descrita através das unidades imaginárias (i, j, k) , todavia, ao longo da apostila será usado o Matlab[®] como *software* para auxiliar as operações e, dentro do Matlab[®], a operação de soma de um vetor com um escalar tem um comportamento bem específico, o Matlab[®] adiciona o valor do escalar a cada componente do vetor conforme exemplificado no Algoritmo 2.5.

Algoritmo 2.5: Exemplo de soma de escalar com vetor no Matlab[®]

```
1 a = 1;
2 v = [10 20 30];
3 w = a + v
```

cujo resultado é

w =

11 21 31

Acrescenta-se ao exposto sobre o Matlab[®] que estritamente do ponto de vista da álgebra linear a soma de um escalar com um vetor não faz sentido, não é uma operação definida a começar pelas dimensões envolvidas. Todavia, o Matlab[®] redefine essa soma para facilitar algumas operações do uso. No caso dos quatérnios a notação explicita o fato de um quatérnio ser formado pelas partes escalar e vetorial. Assim, para evitar conflitos e ambiguidades, a notação que usaremos para evidenciar as partes escalar e vetorial e evitar qualquer problema de interpretação será

$$q = (a, \vec{v}); \quad a \in \mathbb{R}, \quad \vec{v} \in \mathbb{R}^3, \quad q \in \mathbb{H} \quad (2.14)$$

³O corpo \mathbb{K} utilizado ao longo da apostila será o corpo dos reais \mathbb{R} . A abordagem com quatérnios poderia considerar o corpo $\mathbb{K} = \mathbb{C}$, mas preferimos, por simplicidade, usar escalares no corpo dos reais.

Código 11. Crie uma função que retorna um **texto** com a escrita de um quatérnio na base $\{1, i, j, k\}$ conforme definido na Eq. (2.3).

`fc_quat_text(q).`

Exemplo de uso:

```
>> fc_quat_text([1;0;-2;0])
ans =

'1 + 0 i - 2 j + 0 k'
```

2.3.6 Conjugado

O conjugado de um quatérnio é o oposto aditivo de cada unidade imaginária, o que é equivalente a trocar o sinal da parte vetorial.

$$\bar{q} = (a, -\vec{v}) = a - xi - yj - zk \quad (2.15)$$

Código 12. Crie uma função que retorna o conjugado de um quatérnio.

`r = fc_quat_conj(q).`

2.3.7 Módulo

Por desambiguação, a operação de módulo ($|\cdot|$) é um operador aplicado a números e a operação de norma ($\|\cdot\|$) é um operador aplicado a vetores.⁴

Quatérnios aqui serão considerados números, portanto será aplicado o verbete módulo.

O módulo de $q \in \mathbb{H}$ é igual a

$$|q| = \sqrt{a^2 + x^2 + y^2 + z^2} \quad (2.16)$$

Repare que $|q|$ é sempre um número real positivo, i.e., $|q| \in \mathbb{R}^+, \forall q \in \mathbb{H}$, e que $|q| = 0 \Leftrightarrow q = 0 + 0i + 0j + 0k$, o quatérnio nulo, o qual será denotado por $\mathbf{0}_{\mathbb{H}}$.⁵

Código 13. Crie uma função que retorna o módulo de um quatérnio.

`r = fc_quat_modulo(q)`

Dica: “*help norm*”.

Exercício 3. Mostre que $|\lambda q| = |\lambda||q|, \forall \lambda \in \mathbb{R}$.

Exercício 4. Mostre que a operação que extrai a parte escalar de um quatérnio é linear.

Exercício 5. Mostre que a operação que extrai a parte vetorial de um quatérnio é linear.

Exercício 6. Mostre que $q + \bar{q} = 2S(q), \forall q \in \mathbb{H}$.

Exercício 7. Mostre que $q - \bar{q} = 2V(q), \forall q \in \mathbb{H}$.

Exercício 8. Sejam $p, q, r \in \mathbb{H}$ tais que $p = 4 + 5i - 5j + 5k$ e $q = 2 + 2j - 10k$. Calcule

- (a) $r = p + q - \bar{q}$,
- (b) $r = 3 \cdot p + 5 \cdot \bar{q} - |p|$,
- (c) $r = S(p + q) - 2 \cdot S(\bar{q})$,
- (d) $r = |p|^2 S(q)p + |q|^2 S(p)\bar{q}$,
- (e) $r = V(p) + V(2q) - 2 \cdot V(\bar{q})$.

Exercício 9. Sejam $p, q, r \in \mathbb{H}$ tais que $p = 1 + 2i + 3j - 5k$ e $q = -2 + 2i + 2j - 2k$. Calcule

- (a) $r = pi + qj$,
- (b) $r = 3 \cdot pk + 5 \cdot \bar{q}k - |p|(i + j)$,
- (c) $r = S(pi + qj) - 2 \cdot S(\bar{q}k)$,
- (d) $r = |pi|^2 S(qj)pk + |qj|^2 S(pi)\bar{q}j$,
- (e) $r = V(pk) + V(2qk) - 2V(\bar{q}k)$.

⁴Mais precisamente, norma é aplicado a qualquer elemento de espaço vetorial, o que inclui matrizes, tensores e funções.

⁵ \mathbb{R}^+ representa o conjunto dos reais não-negativos.

2.3.8 Multiplicação entre quatérnios

A afirmação mais importante sobre a multiplicação entre quatérnios é que **a multiplicação entre quatérnios não é comutativa**, ou seja, $pq \neq qp$, exceto casos particulares. Por outro lado, vale a propriedade distributiva.⁶

Sejam $p, q \in \mathbb{H}$ tais que

$$\begin{aligned} q &= a_1 + x_1e_1 + y_1e_2 + z_1e_3 = (a_1, \vec{v}_1) \\ p &= a_2 + x_2e_1 + y_2e_2 + z_2e_3 = (a_2, \vec{v}_2) \end{aligned} \quad (2.17)$$

realizaremos a multiplicação organizando os números tabuladamente.⁷

$$\begin{aligned} qp &= \begin{array}{l} a_1(a_2 + x_2e_1 + y_2e_2 + z_2e_3) \\ + x_1e_1(a_2 + x_2e_1 + y_2e_2 + z_2e_3) \\ + y_1e_2(a_2 + x_2e_1 + y_2e_2 + z_2e_3) \\ + z_1e_3(a_2 + x_2e_1 + y_2e_2 + z_2e_3) \end{array} = \begin{array}{l} a_1a_2 + a_1x_2e_1 + a_1y_2e_2 + a_1z_2e_3 \\ + x_1a_2e_1 + x_1x_2e_1^2 + x_1y_2e_1e_2 + x_1z_2e_1e_3 \\ + y_1a_2e_2 + y_1x_2e_2e_1 + y_1y_2e_2^2 + y_1z_2e_2e_3 \\ + z_1a_2e_3 + z_1x_2e_3e_1 + z_1y_2e_3e_2 + z_1z_2e_3^2 \end{array} \end{aligned} \quad (2.18)$$

apliquemos os resultados da Tabela 2.1

$$\begin{aligned} qp &= \begin{array}{l} a_1a_2 + a_1x_2e_1 + a_1y_2e_2 + a_1z_2e_3 \\ + x_1a_2e_1 + x_1x_2(-1) + x_1y_2(e_3) + x_1z_2(-e_2) \\ + y_1a_2e_2 + y_1x_2(-e_3) + y_1y_2(-1) + y_1z_2(e_1) \\ + z_1a_2e_3 + z_1x_2(e_2) + z_1y_2(-e_1) + z_1z_2(-1) \end{array} \end{aligned} \quad (2.19)$$

reorganizando os termos escalares e os vetoriais de acordo com o elemento da base

$$\begin{aligned} qp &= \begin{array}{l} a_1a_2 - x_1x_2 - y_1y_2 - z_1z_2 \\ + (x_1a_2 + a_1x_2 + y_1z_2 - z_1y_2)e_1 \\ + (y_1a_2 + a_1y_2 + z_1x_2 - x_1z_2)e_2 \\ + (z_1a_2 + a_1z_2 + x_1y_2 - y_1x_2)e_3 \end{array} \end{aligned} \quad (2.20)$$

para simplificar a notação da multiplicação entre dois quatérnios, utilizaremos as notações de produto interno e de produto vetorial definidos a seguir.

Sejam $\vec{v} = [v_1, v_2, v_3]^T$ e $\vec{u} = [u_1, u_2, u_3]^T \in \mathbb{R}^3$, define-se o produto interno entre $\vec{v}, \vec{u} \in \mathbb{R}^3$

$$\langle \vec{v}, \vec{u} \rangle = v_1u_1 + v_2u_2 + v_3u_3 \quad (2.21)$$

e o produto vetorial entre $\vec{v}, \vec{u} \in \mathbb{R}^3$

$$\vec{v} \times \vec{u} = (v_2u_3 - v_3u_2)e_1 + (v_3u_1 - v_1u_3)e_2 + (v_1u_2 - v_2u_1)e_3 \quad (2.22)$$

Os resultados da Eq. (2.20) são então reescritos com as definições das Equações (2.21) e (2.22)

$$qp = (a_1a_2 - \langle \vec{v}_1, \vec{v}_2 \rangle, a_1\vec{v}_2 + a_2\vec{v}_1 + \vec{v}_1 \times \vec{v}_2) = (b, \vec{w}); \quad b \in \mathbb{R}, \vec{w} \in \mathbb{R}^3 \therefore (qp) \in \mathbb{H} \quad (2.23)$$

de onde conclui-se que o produto de dois quatérnios é um quatérnio. Destaca-se que o produto de dois quatérnios contém o produto interno de seus vetores na parte escalar do resultado e o produto vetorial na sua parte vetorial. Se as partes escalares de q e p forem nulas, i.e., $a_1 = a_2 = 0$, tem-se que o resultado do produto de dois quatérnios é

$$qp = (-\langle \vec{v}_1, \vec{v}_2 \rangle, \vec{v}_1 \times \vec{v}_2) \quad (2.24)$$

Código 14. Crie uma função que retorna o produto de dois quatérnios.

`r = fc_quat_produto(q,p)`

Dica(1): “help dot”. **Dica(2):** “help cross”.

Exercício 10. Mostre que no caso geral $qp \neq pq$.

⁶OBS: é comum encontrar a notação $q \otimes p$ para o produto de quatérnios (CONCI, 2014). Optamos por não usar o símbolo \otimes para manter a notação mais limpa e deixar claro que o produto de quatérnios é um produto “normal” com a propriedade distributiva.

⁷Os autores gostam e recomendam essa forma de escrita tabulada por entenderem que ela facilita consideravelmente a compreensão das relações importantes da operação.

Exercício 11. Apresente um exemplo em que $qp \neq pq$.

Exercício 12. Mostre que o produto interno é comutativo, i.e., $\langle \vec{v}, \vec{u} \rangle = \langle \vec{u}, \vec{v} \rangle, \forall \vec{v}, \vec{u} \in \mathbb{R}^3$.

Exercício 13. (a) Mostre que o produto vetorial é anticomutativo, i.e., $\vec{v} \times \vec{u} = -\vec{u} \times \vec{v}, \forall \vec{v}, \vec{u} \in \mathbb{R}^3$.

(b) O que significa então o fato de dois quatérnios q, p comutarem na multiplicação?

(c) Apresente um exemplo de quatérnios q, p cuja multiplicação comuta.

(d) Dado o quatérnio $q = 3 + 4i + 5j - 9k$, encontre a forma de todos os quatérnios $p = a + bi + cj + dk$, tais que $qp = pq$.

Exercício 14. Mostre que $|pq| = |p||q|, \forall p, q \in \mathbb{H}$.

Exercício 15. (Desafio) Como verificar numericamente que $\forall q, p \in \mathbb{H}, S(qp) = S(pq)$?

Exercício 16. Mostre que $\bar{q}q = q\bar{q}, \forall q \in \mathbb{H}$.

Exercício 17. Mostre que $\bar{q}q = |q|^2, \forall q \in \mathbb{H}$.

Exercício 18. Mostre que o produto de quatérnios é associativo, i.e., $\forall q, p, r \in \mathbb{H}, (qp)r = q(pr)$.

Dica(1): $\forall \vec{v}, \vec{u}, \vec{w} \in \mathbb{R}^3 \quad \vec{v} \times (\vec{u} \times \vec{w}) = (\langle \vec{v}, \vec{w} \rangle) \vec{u} - (\langle \vec{v}, \vec{u} \rangle) \vec{w}$.

Dica(2): Identidade de Jacobi: $\vec{v} \times (\vec{u} \times \vec{w}) + \vec{u} \times (\vec{w} \times \vec{v}) + \vec{w} \times (\vec{v} \times \vec{u}) = \mathbf{0}_{\mathbb{R}^3}$.

Exercício 19. (Desafio) Mostre que $\overline{(qp)} = \bar{p}\bar{q}$.

Exercício 20. (Bônus) Usemos por um breve momento vetores que admitem componentes complexas. Seja $\langle q, p \rangle$ o produto interno de quatérnio tal que $\langle q, p \rangle = \underline{q}^H \underline{p}$, onde $\underline{q} = [a, xi, yj, zk]^\top$ e $\underline{q}^H = \overline{(\underline{q}^\top)} = (\bar{q})^\top$ é o **transposto conjugado**⁸ de q , i.e., calcula-se o conjugado do transposto de q que por sua vez é igual ao transposto do conjugado de q .

(a) Mostre que $\langle \cdot, \cdot \rangle : \mathbb{H} \times \mathbb{H} \rightarrow \mathbb{C}$,

(b) Mostre que $\langle \underline{q}, \underline{q} \rangle = |q|^2$.

2.3.9 Divisão entre quatérnios

Sejam $q, p \in \mathbb{H}$. Faz sentido a operação $\frac{q}{p}$? Se fizer, então o que é dividir por um vetor?

Dividir por vetor (e analogamente por matriz) gera complicações quanto às dimensões envolvidas e, consequentemente, o tipo de objeto do resultado, seria o resultado um escalar, outro vetor ou uma matriz?

Analisando como exemplo a “divisão” por matriz, tem-se que na verdade é realizada uma multiplicação pela inversa da matriz em questão seja à esquerda ou à direita conforme a situação, lembrando que somente matrizes quadradas admitem inversa.

Desta forma, a operação de divisão entre vetor e matriz é dita “não bem definida”. Algo similar acontecerá com os quatérnios.

Seja \hat{q} um quatérnio unitário, tal que seu módulo $|\hat{q}| = 1$. Para se obter um quatérnio unitário é necessário dividir o quatérnio por seu módulo, $\hat{q} = \frac{q}{|q|}$, assim

$$|\hat{q}|^2 = 1 \Rightarrow (\hat{q})(\bar{\hat{q}}) = 1 \quad (2.25)$$

com base no Exercício 16, fica claro que a definição de divisão entre quatérnios na verdade deve seguir a ideia da inversa de uma matriz. Seja q^{-1} o quatérnio inverso de q tal que

$$q^{-1}q = qq^{-1} = 1; \quad \forall q \in \mathbb{H} \quad (2.26)$$

cujos resultados combinados com o Exercício 17 tem-se

$$qq^{-1} = 1 \Rightarrow \bar{q}qq^{-1} = \bar{q}1 \Rightarrow |q|^2 q^{-1} = \bar{q} \Rightarrow q^{-1} = \frac{\bar{q}}{|q|^2} \quad (2.27)$$

de onde nota-se claramente que se q for unitário, seu inverso será o próprio conjugado

$$\hat{q}^{-1} = \frac{\bar{\hat{q}}}{|\hat{q}|^2} = \frac{\bar{\hat{q}}}{1} = \bar{\hat{q}} \quad (2.28)$$

⁸Também denotado por $(\cdot)^\dagger$. E também denominada *adjunta hermitiana*.

cuja propriedade é idêntica às das matrizes de rotação ($R^{-1} = R^T$).

Portanto a “divisão” de quatérnios deve ser feita multiplicando pelo inverso respeitando a posição “à esquerda” ($p^{-1}q$) ou “à direita” (qp^{-1}). A notação $\frac{q}{p}$ é ambígua e inadequada.

Código 15. *Crie uma função que retorna o inverso de um quatérnio.*

`r = fc_quat_inverso(q)`

Código 16. *Crie uma função que retorna o produto pelo inverso de um quatérnio à esquerda.*

`r = fc_quat_divisao_esq(p,q)`

Dica: *Utilize as funções que calculam inverso de um quatérnio e o produto entre dois quatérnios. .*

Código 17. *Crie uma função que retorna o produto pelo inverso de um quatérnio à direita.*

`r = fc_quat_divisao_dir(q,p)`

Dica: *Utilize as funções que calculam inverso de um quatérnio e o produto entre dois quatérnios. .*

Exercício 21. *Sejam $q = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}i$ e $p = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}j$, calcule*

- (a) $r_1 = q^{-1}p$,
- (b) $r_2 = pq^{-1}$,
- (c) $s_1 = qp^{-1}$,
- (d) $s_2 = p^{-1}q$.

Exercício 22. *Calcule o inverso de*

- (a) $q = \frac{1}{2} + \frac{1}{2}i + \frac{1}{2}j + \frac{1}{2}k$,
- (b) $p = 0 + \frac{\sqrt{3}}{3}i + \frac{\sqrt{3}}{3}j + \frac{\sqrt{3}}{3}k$.

Exercício 23. *Repita a demonstração da Eq. (2.27) para a condição $q^{-1}q$.*

Capítulo 3

Quatérnios como “complexo de complexo”

3.1 Isomorfismo entre quatérnios, plano complexo e espaço real de 4 dimensões

Ao longo do [Capítulo 2](#), utilizamos a relação de isomorfismo entre \mathbb{H} e \mathbb{R}^4 para ser possível construir as funções em Matlab[®] para realizar as operações definidas. Nesta seção serão apresentadas as relações para a escrita de quatérnios como elementos de espaços vetoriais ([ROSA, 2011](#)).

Seja $q \in \mathbb{H}$

$$q = \alpha + \beta j; \quad \alpha, \beta \in \mathbb{C}, \quad j^2 = -1 \quad (3.1)$$

tal que

$$\alpha = q_0 + q_1 i; \quad \beta = q_2 + q_3 i; \quad i^2 = -1; \quad q_1, q_2, q_3, q_4 \in \mathbb{R} \quad (3.2)$$

então

$$q = q_0 + q_1 i + (q_2 + q_3 i)j = q_0 + q_1 i + q_2 j + q_3 i j = q_0 + q_1 i + q_2 j + q_3 k \in \mathbb{H} \quad (3.3)$$

de onde se conclui que quatérnios podem ser descritos como elementos de um espaço vetorial de duas dimensões complexas \mathbb{C}^2 .

A [Figura 3.1](#) apresenta uma comparação entre os isomorfismos de \mathbb{R}^2 e \mathbb{C} com o isomorfismo entre \mathbb{R}^4 e \mathbb{C}^2 . A [Figura 3.1\(a\)](#) exibe um plano cartesiano em que cada eixo ordenado é uma reta real, assim um vetor $\vec{v} = [v_x, v_y]^T \in \mathbb{R}^2$ é equivalente a um número $z = v_x + v_y i \in \mathbb{C}$. A [Figura 3.1\(b\)](#) é análoga com a variação de que cada eixo ordenado agora é um plano complexo por si.¹

Sejam π_1 e π_2 os planos complexos que representam os eixos das abcissas e das ordenadas respectivamente, tais que a origem é a única intersecção de cada plano complexo

$$\pi_1 = \{1, e_0\}, \quad \pi_2 = \{e_2, e_3\}, \quad \pi_1 \cap \pi_2 = [0, 0, 0, 0]^T \quad (3.4)$$

assim a descrição de $q \in \mathbb{C}^2$ é feita em função de $\alpha \in \pi_1$ e $\beta \in \pi_2$

$$q = \alpha + \beta j = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (3.5)$$

Sem entrarmos nos detalhes da prova, como já apresentamos a relação entre \mathbb{R}^2 e \mathbb{C} , podemos então concluir a seguinte cadeia de isomorfismos: $\mathbb{H} \cong \mathbb{C}^2 \cong \mathbb{R}^4$.

3.2 Representação de quatérnios com matrizes

Iniciemos com o isomorfismo entre os números complexos e um grupo de matrizes.

¹O código gerador da figura encontra-se no Apêndice [D.5](#), Algoritmos [D.34](#) e [D.35](#).

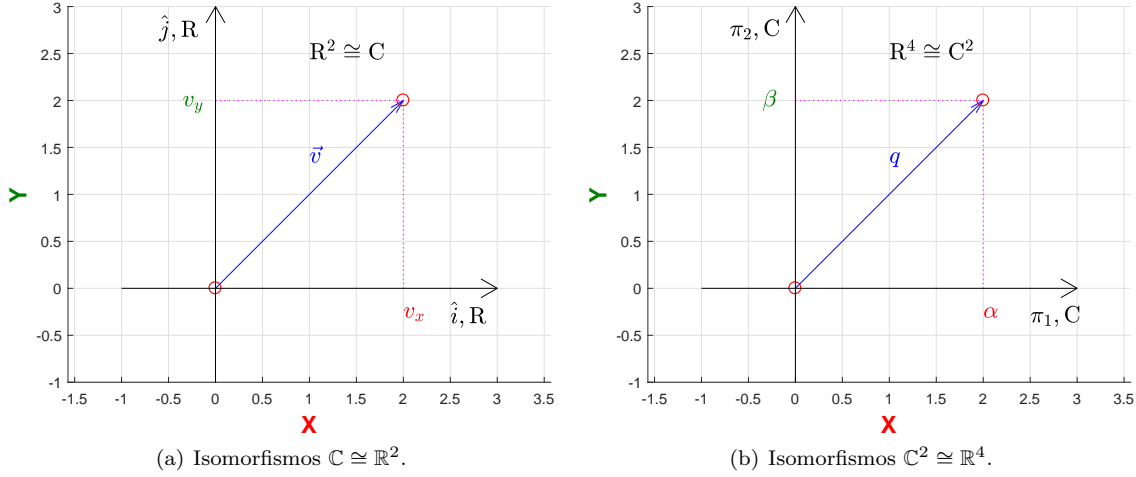


Figura 3.1: Isomorfismos $\mathbb{C} \cong \mathbb{R}^2$ e entre $\mathbb{C}^2 \cong \mathbb{R}^4$. \mathbb{C}^2 pode ser interpretado como um plano em que cada eixo é um plano complexo \mathbb{C} .

Considere o isomorfismo $\mathbb{C} \cong \mathbb{R}^2 \cong \mathbb{R}^{2 \times 2}$ com a seguinte regra

$$M_0 = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \quad (3.6)$$

As operações de soma e produto matriciais para o espaço formado com a regra de M_0 preservam nas respectivas componentes os resultados das mesmas operações com números complexos.

Exemplo 1. *Uso de matrizes para calcular números complexos no Matlab®*

Algoritmo 3.1: Exemplo do isomorfismo descrito pela Eq. (3.6).

```

1 a = 1; b = 1;
2 c = 3; d = 4;
3 z1 = a + b*1i
4 z2 = c + d*1i
5 z3 = z1*z2
6 A1 = [real(z1) -imag(z1); imag(z1) real(z1)]
7 A2 = [real(z2) -imag(z2); imag(z2) real(z2)]
8 A3 = A1*A2

```

O resultado é

$$z_1 = 1 + 1j; z_2 = 3 + 4j \Rightarrow z_3 = z_1 \cdot z_2 = -1 + 7j \quad (3.7)$$

$$A_1 = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}; A_2 = \begin{bmatrix} 3 & -4 \\ 4 & 3 \end{bmatrix} \Rightarrow A_3 = A_1 \cdot A_2 = \begin{bmatrix} -1 & -7 \\ 7 & -1 \end{bmatrix} \quad (3.8)$$

a primeira coluna de A_3 é exatamente o número complexo z_3 e, bônus, a primeira linha de A_3 é o seu complexo conjugado \bar{z}_3 .

Agora, vejamos a construção do isomorfismo entre um espaço vetorial de duas dimensões complexas, \mathbb{C}^2 , e um grupo de matrizes, tal como foi feito para complexos.

Isomorfismo $\mathbb{C}^2 \cong \mathbb{R}^4 \cong \mathbb{R}^{4 \times 4}$ com a seguinte regra

$$M_1 = \begin{bmatrix} \alpha & \beta \\ -\bar{\beta} & \bar{\alpha} \end{bmatrix} = \begin{bmatrix} (q_0 + q_1j) & (q_2 + q_3j) \\ (-q_2 + q_3j) & (q_0 - q_1j) \end{bmatrix} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \quad (3.9)$$

Tal como para M_0 em relação aos complexos \mathbb{C} , a soma e o produto matriciais, com a regra de M_1 , preservam as respectivas operações no espaço dos quatérnios \mathbb{H} .

Exemplo 2. *Uso de matrizes para calcular quatérnios no Matlab®*

Algoritmo 3.2: Exemplo do isomorfismo descrito pela Eq. (3.9).

```

1 q0 = 1; q1 = 2; q2 = 3; q3 = 4;
2 p0 = 10; p1 = -1; p2 = -2; p3 = -3;
3 q = [q0; q1; q2; q3];
4 p = [p0; p1; p2; p3];
5 r = fc_lib_quat_produto(q, p)
6 Q = [q, [-q1 q0 q3 -q2]', [-q2 -q3 q0 q1]', [-q3 q2 -q1 q0]']
7 P = [p, [-p1 p0 p3 -p2]', [-p2 -p3 p0 p1]', [-p3 p2 -p1 p0]']
8 R = Q * P

```

Os resultados são

$$q = 1 + 2i + 3j + 4k; p = 10 - 1i - 2j - 3k \Rightarrow r = q \cdot p = 30 + 18i + 30j + 36k \quad (3.10)$$

$$Q = \begin{bmatrix} 1 & -2 & -3 & -4 \\ 2 & 1 & -4 & 3 \\ 3 & 4 & 1 & -2 \\ 4 & -3 & 2 & 1 \end{bmatrix}; P = \begin{bmatrix} 10 & 1 & 2 & 3 \\ -1 & 10 & 3 & -2 \\ -2 & -3 & 10 & 1 \\ -3 & 2 & -1 & 10 \end{bmatrix} \Rightarrow R = Q \cdot P = \begin{bmatrix} 30 & -18 & -30 & -36 \\ 18 & 30 & -36 & 30 \\ 30 & 36 & 30 & -18 \\ 36 & -30 & 18 & 30 \end{bmatrix} \quad (3.11)$$

a primeira coluna de R é exatamente o quatérnio r e, novamente há um bônus, a primeira linha de R contém o conjugado \bar{r} .

Exercício 24. Mostre que utilizando o grupo de matrizes definida pela matriz M_2 a seguir, também se preserva as operações de soma e produto de quatérnios.

$$M_2 = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \quad (3.12)$$

Mais precisamente há um total de 48 conjuntos de quádruplas de matrizes tais que a função que envia $1, i, j, k$ para as matrizes é um homeomorfismo, i.e., a função preserva a soma e os produtos dos quatérnios na soma e no produto das matrizes (FAREBROTHER; GROß; TROSCHKE, 2003).

Exercício 25. Mostre que $\det(M_1) = |q|^2$.

Exercício 26. Sejam os quatérnios $q = 1i + 2j + 3k = (0, \vec{u})$ e $p = 1i + 10j + 5k = (0, \vec{v})$.

- Calcule $r = qp$,
- Calcule as matrizes Q , P e R utilizando o isomorfismo entre \mathbb{H} e $\mathbb{R}^{4 \times 4}$ dado por M_1 da Eq. (3.9) para os quatérnios q, p e r respectivamente,
- Calcule $R_1 = Q \cdot P$ e compare R com R_1 ,
- Calcule o produto vetorial $\vec{w} = \vec{u} \times \vec{v}$,
- Calcule a matriz S a partir de $s = (0, \vec{w}) \in \mathbb{H}$ e compare com R ,
- O que pode ser dito da diferença $S - R$?

Exercício 27. Sejam os quatérnios $q = 0 + q_1i + 5j - 1k$ e $p = 1 + 0i + 1j - 2k$ e suas respectivas matrizes Q e P construídas a partir de M_1 . Calcule q_1 sabendo que

$$R = Q \cdot P = \begin{bmatrix} -7 & 8 & -7 & 0 \\ -8 & -7 & 0 & 7 \\ 7 & 0 & -7 & 8 \\ 0 & -7 & -8 & -7 \end{bmatrix} \quad (3.13)$$

Exercício 28. Sejam os quatérnios $q = 0 + q_1i + 1j + 3k$ e $p = -5 + 2i + p_2j + 1k$ e suas respectivas matrizes Q e P construídas a partir de M_1 . Calcule q_1 e p_2 sabendo que

$$R = Q \cdot P = \begin{bmatrix} -27 & 61 & 9 & -23 \\ -61 & -27 & -23 & -9 \\ -9 & 23 & -27 & 61 \\ 23 & 9 & -61 & -27 \end{bmatrix} \quad (3.14)$$

Capítulo 4

Rotações

4.1 Rotações 2D

Sejam $z_1, z_2 \in \mathbb{C}$

$$z_1 = x_1 + y_1 i; \quad z_2 = x_2 + y_2 i \quad (4.1)$$

cujas formas polares são

$$z_1 = (x_1, y_1) \equiv \rho_1 \angle \theta_1; \quad z_2 = (x_2, y_2) \equiv \rho_2 \angle \theta_2 \quad (4.2)$$

e seja z_3 o produto entre z_1 e z_2

$$\begin{aligned} z_3 = z_1 z_2 &= (x_1 + y_1 i) \cdot (x_2 + y_2 i) \\ &= (x_1 x_2 - y_1 y_2) + (x_1 y_2 - y_1 x_2) i \\ &= (x_1 x_2 - y_1 y_2, x_1 y_2 - y_1 x_2) \end{aligned} \quad (4.3)$$

O cálculo de z_3 em forma polar é bem mais simples e permite verificar que o módulo de z_3 é dado pelo produto dos módulos de z_1 e z_2 e o ângulo de z_3 pela soma dos ângulos de z_1 e z_2

$$\begin{aligned} z_3 &= (\rho_1 \angle \theta_1)(\rho_2 \angle \theta_2) \\ &= (\rho_1 \rho_2) \angle (\theta_1 + \theta_2) \end{aligned} \quad (4.4)$$

Por exemplo, uma rotação de 90° é obtida com $z_2 = i \equiv 1 \angle (\pi/2)$

$$\begin{aligned} z_3 &= (x_1 \cdot 0 - y_1 \cdot 1, x_1 \cdot 1 - y_1 \cdot 0) \\ &= (-y_1, x_1) \\ z_3 &= (\rho_1 \cdot 1) \angle (\theta_1 + 90^\circ) = \rho_1 \angle (\theta_1 + 90^\circ) \end{aligned} \quad (4.5)$$

O caso em que $z_1 = 2 + 2i$ é ilustrado na [Figura 4.1¹](#), cujo módulo $\rho_1 = \sqrt{2}$ e o ângulo $\theta_1 = 45^\circ$. Repare que o sentido positivo da rotação é no sentido anti-horário.

4.2 Rotações 3D - Ângulos de Euler

Os ângulos de Euler são ϕ , rolagem (*roll*) ao redor do eixo **X**, θ , arfagem (*pitch*) ao redor do eixo **Y** e ψ , guinada (*yaw*) ao redor do eixo **Z** ([CONCI, 2014](#)).

As matrizes de rotação sobre cada eixo **X**, **Y**, **Z** são:

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}; \quad R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}; \quad R_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

As Figuras [4.2](#), [4.3](#) e [4.4](#) apresentam respectivamente as rotações de um ângulo ϕ ao redor de x , de θ ao redor de y e ψ ao redor de z . Cada base do sistema de coordenadas é representada por uma aresta da respectiva cor (**X**, **Y**, **Z**), tamanho unitário e a rotação é positiva no sentido anti-horário “olhando para a ponta do vetor”.

¹O código gerador da figura encontra-se no Apêndice [D.5](#), Algoritmo [D.36](#).

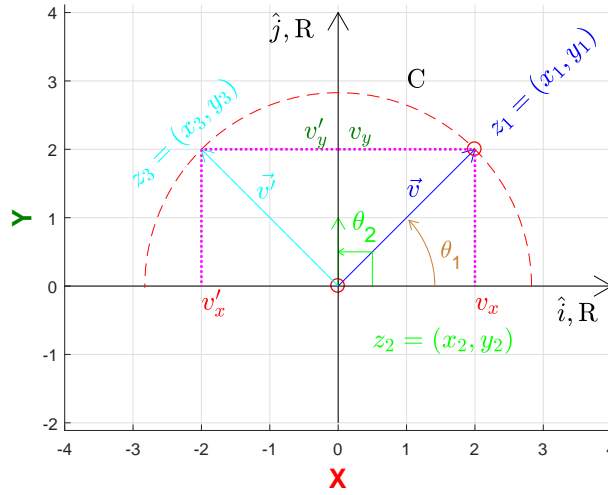
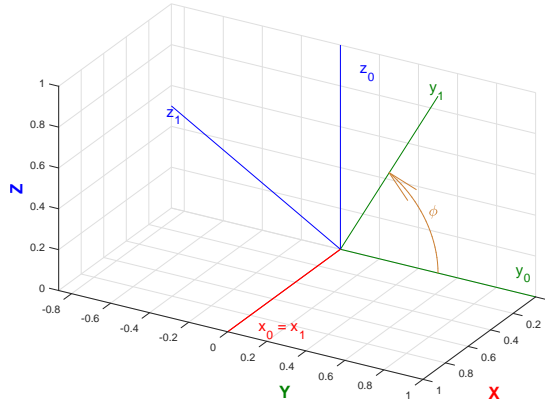


Figura 4.1: Exemplo de rotação em 2D com uso de números complexos.

Figura 4.2: Definição matriz de rotação R_x .

Exercício 29. Para $R = \{R_x, R_y, R_z\}$

- (a) Mostre que $RR^T = I$,
- (b) Mostre que $R^T R = I$.

Exercício 30. Mostre que $\det(R) = 1$. Isso significa que R é uma matriz do grupo especial (Special).

Exercício 31. Do Exercício 29, conclui-se que $R^T = R^{-1}$ como consequência do fato de R^T seguir a definição de uma matriz inversa. Mostre que $R^T = R^{-1}$ utilizando $R^{-1} = \frac{\text{adj}(R)}{\det(R)}$.

Isso significa que R é uma matriz do grupo ortogonal ($O(n)$). Em adição ao resultado do Exercício 30, tem-se que R é uma matriz do grupo especial e ortogonal, ou seja, $R \in \text{SO}(n)$, com $n = 3$. $\text{SO}(n) = \{R \in \mathbb{R}^{n \times n} \mid RR^T = R^T R = I, n \in \mathbb{N}_*\}$.²

Explicações sobre os grupos de matrizes especial, ortogonal e especial-ortogonal encontram-se em Baker (2000) e em Cameron (2006).

Código 18. Faça uma função que retorne a matriz de rotação sobre o eixo X dado um ângulo ϕ .

$R_x = \text{fc_Rx}(\phi)$

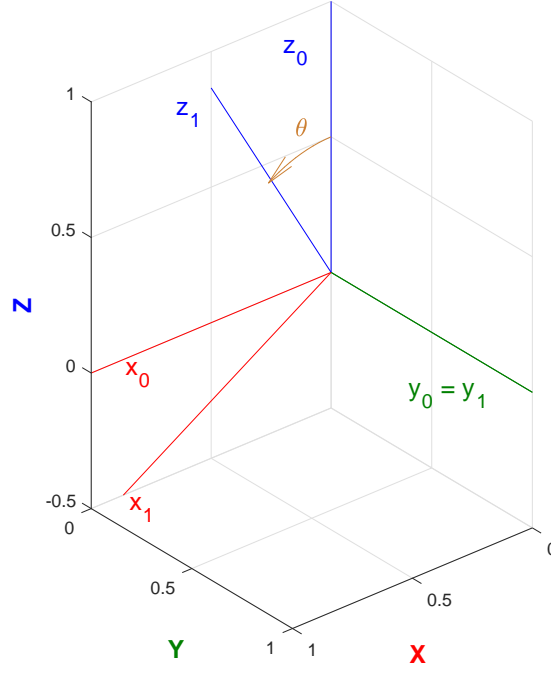
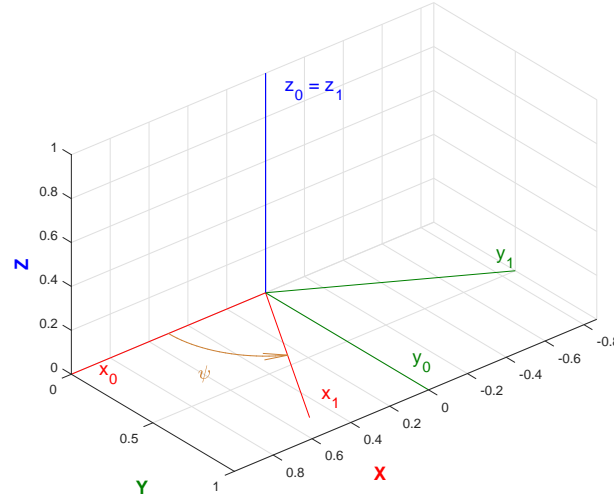
Código 19. Faça uma função que retorne a matriz de rotação sobre o eixo Y dado um ângulo θ .

$R_y = \text{fc_Ry}(\theta)$

Código 20. Faça uma função que retorne a matriz de rotação sobre o eixo Z dado um ângulo ψ .

$R_z = \text{fc_Rz}(\psi)$

² \mathbb{N}_* conjunto dos naturais sem o zero.

Figura 4.3: Definição matriz de rotação R_y .Figura 4.4: Definição matriz de rotação R_z .

A obtenção de rotações 3D utilizando os ângulos de Euler é obtida aplicando uma rotação após o resultado da anterior (WEISSTEIN, 2019; PERUMAL, 2014). Assim, uma sequência de rotação **XYZ** significa que a base original de coordenadas é rotacionada ao redor de **X**, em seguida ao redor de **Y** e por último ao redor de **Z**. A expressão matemática que traduz essa sequência de rotação é

$$R_{xyz}(\phi, \theta, \psi) = R_x(\phi)R_y(\theta)R_z(\psi) \quad (4.7)$$

O resultado da matriz de rotação para a sequência de rotação **XYZ** é

$$R_{xyz}(\phi, \theta, \psi) = \begin{bmatrix} \cos(\psi) \cos(\theta) & -\cos(\theta) \sin(\psi) & \sin(\theta) \\ \cos(\phi) \sin(\psi) + \cos(\psi) \sin(\phi) \sin(\theta) & \cos(\phi) \cos(\psi) - \sin(\phi) \sin(\psi) \sin(\theta) & -\cos(\theta) \sin(\phi) \\ \sin(\phi) \sin(\psi) - \cos(\phi) \cos(\psi) \sin(\theta) & \cos(\psi) \sin(\phi) + \cos(\phi) \sin(\psi) \sin(\theta) & \cos(\phi) \cos(\theta) \end{bmatrix} \quad (4.8)$$

e será utilizado nos Exemplos 3 e 4 a seguir. As matrizes dos demais exemplos serão apresentadas como exercícios.

Para se criar as matrizes de rotação de cada sequência é interessante aproveitar os códigos 18, 19 e 20 e usar variáveis simbólicas para se copiar o resultado e criar a função com base nele. Entretanto,

aconselhamos testar se é mais rápido utilizar as três funções já criadas ou se é mais rápido aplicar diretamente o resultado das três operações.

Código 21. O Algoritmo 4.1 apresenta um exemplo de como testar variações de códigos curtos.

O Matlab® possui uma ferramenta chamada **profile viewer** para analisar o tempo de execução de códigos. Essa ferramenta é acessível pela linha de comando e também pela opção **Run and Time**. Todavia essa ferramenta não é muito adequada para testar o tempo de execução de códigos curtos que são pensados para serem chamados diversas vezes por outras funções. Por outro lado, ela é ideal para indicar o tempo que o programa consome em cada função que for chamada durante a execução.

O “Teste 1” apresenta o resultado cujo cálculo é feito partir do produto das três funções previamente criadas conforme a Eq. (4.6), e o “Teste 2” apresenta o resultado cujo cálculo é feito a partir do resultado dos três produtos explicitamente descritos tal como na Eq. (4.8).

O comando **tic** dispara o relógio e o comando **toc** apresenta o total de tempo decorrido desde o último **tic**.

Ambos os testes apresentam 4 conjuntos de 10 mil iterações para indicar como cada pequena fração de tempo de chamada de função e de cálculo mais longo se acumulam.

Algoritmo 4.1: Exemplo de teste básico de tempo de execução em Matlab®

```

1 disp('Teste 1');
2 tic;
3 for i = 1:4*10^4
4     phi = rand; th = rand; psi = rand;
5     R = fc_lib_Rx(phi)*fc_lib_Ry(th)*fc_lib_Rz(psi);
6 end
7 toc
8 disp('Teste 2');
9 tic;
10 for i = 1:4*10^4
11     phi = rand; th = rand; psi = rand;
12     R = fc_lib_R_xyz(phi,th,psi);
13 end
14 toc

```

Um dos resultados foi:

```

Teste 1
Elapsed time is 1.549532 seconds.
Teste 2
Elapsed time is 0.469165 seconds.

```

de onde se conclui que o “Teste 2” foi mais eficiente neste teste em particular.

É interessante notar que deve-se repetir o teste algumas vezes para compensar os efeitos da aleatoriedade na geração dos dados e também os efeitos do carregamento das funções na memória do programa. Assim, a conclusão sobre o tempo de execução de cada estratégia é mais robusta.

4.2.1 Visualizações e Exemplos de Rotações 3D com Ângulos de Euler

Para todos os exemplos apresentados, a base de coordenadas inicial é

$$\xi_0 = [\hat{e}_1, \hat{e}_2, \hat{e}_3] = I_3$$

e se localiza na coordenada $\mathbf{X} = 0$.

A base após a primeira parte da sequência de rotações é denotada por ξ_1 e se localiza em $\mathbf{X} = 2$, a base após a segunda rotação é denotada por ξ_2 e localizada em $\mathbf{X} = 4$ e finalmente o resultado das três rotações combinadas é denotado por ξ_3 e localizada em $\mathbf{X} = 6$.

Essa forma de representação tem como objetivo permitir a visualização de cada etapa de rotação de forma sequencial em uma única imagem.

Exemplos em forma de animações podem ser encontradas em (ROBOTICS, 2017a).

Exemplo 3. A Figura 4.5 apresenta os passos da rotação \mathbf{XYZ} com 90° ao redor de cada eixo. A primeira rotação é ao redor de \mathbf{X} a segunda rotação ao redor de \mathbf{Y} e a terceira é ao redor de \mathbf{Z} .

Os resultados da base a cada rotação são apresentadas a seguir

$$\xi_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \xi_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}; \quad \xi_2 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}; \quad \xi_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (4.9)$$

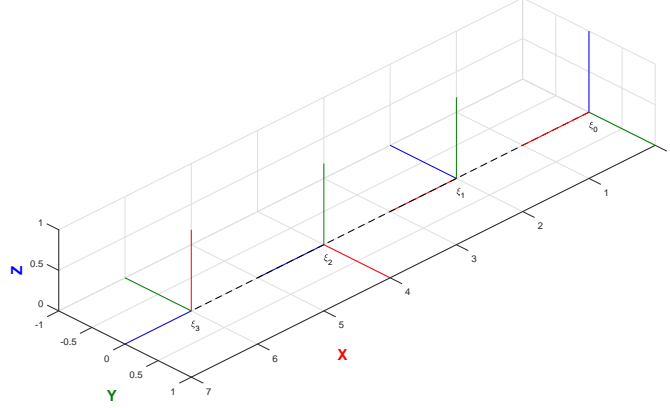


Figura 4.5: Três passos da rotação **XYZ**: $\phi_X = 90^\circ$; $\theta_Y = 90^\circ$; $\psi_Z = 90^\circ$.

Exemplo 4. A Figura 4.6 apresenta rotação **XYZ** em que o ângulo ao redor de **X** e **Z** são de 45° e o ângulo ao redor de **Y** é 90° . Destaca-se que essa escolha de valores deixa bem clara a cada passagem da sequência de rotação.

$$\xi_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \xi_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sqrt{2}/2 & -\sqrt{2}/2 \\ 0 & \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix}; \quad \xi_2 = \begin{bmatrix} 0 & 0 & 1 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ -\sqrt{2}/2 & \sqrt{2}/2 & 0 \end{bmatrix}; \quad \xi_3 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.10)$$

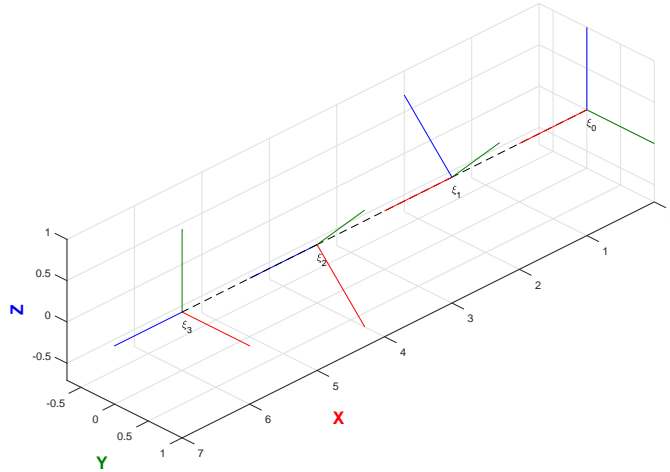


Figura 4.6: Três passos da rotação **XYZ**: $\phi_X = 45^\circ$; $\theta_Y = 90^\circ$; $\psi_Z = 45^\circ$.

Exemplo 5. A Figura 4.7 apresenta os passos da rotação **XZY** com 90° ao redor de cada eixo. A primeira rotação é ao redor de **X** a segunda rotação ao redor de **Z** e a terceira é ao redor **Y**.

Os resultados da base a cada rotação são apresentadas a seguir

$$\xi_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \xi_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}; \quad \xi_2 = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}; \quad \xi_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

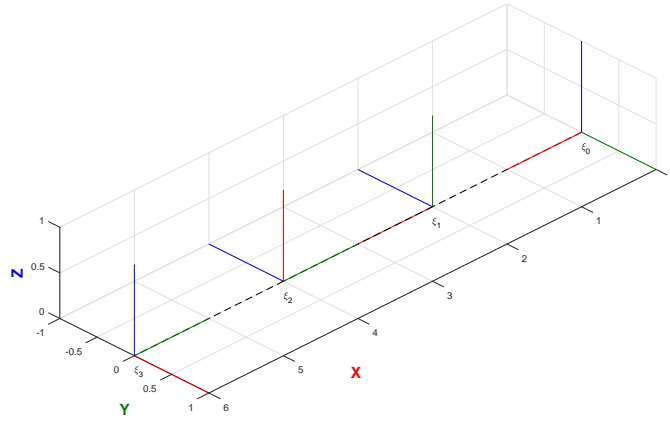


Figura 4.7: Três passos da rotação \mathbf{XYZ} : $\phi_X = 90^\circ$; $\theta_Z = 90^\circ$; $\psi_Y = 90^\circ$.

Exemplo 6. A Figura 4.8 apresenta os passos da rotação \mathbf{YXZ} com 90° ao redor de cada eixo. A primeira rotação é ao redor de \mathbf{Y} a segunda rotação ao redor de \mathbf{X} e a terceira é ao redor \mathbf{Z} .

Os resultados da base a cada rotação são apresentadas a seguir

$$\xi_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \xi_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}; \quad \xi_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix}; \quad \xi_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.12)$$

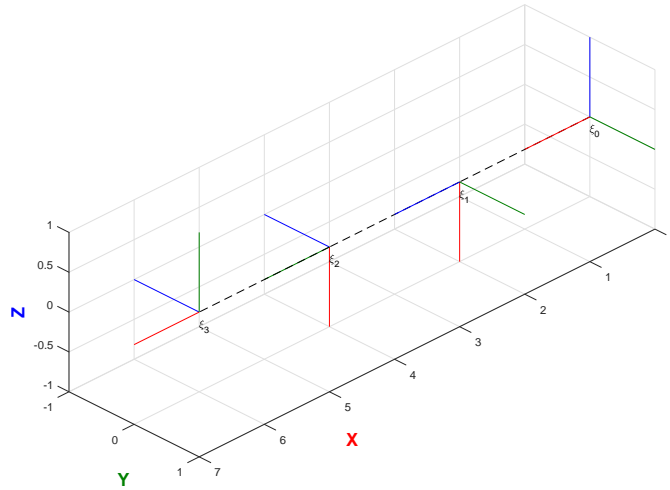


Figura 4.8: Três passos da rotação \mathbf{YXZ} : $\phi_X = 90^\circ$; $\theta_Y = 90^\circ$; $\psi_Z = 90^\circ$.

Exemplo 7. A Figura 4.9 apresenta os passos da rotação \mathbf{ZYX} com 90° ao redor de cada eixo. A primeira rotação é ao redor de \mathbf{Z} a segunda rotação ao redor de \mathbf{Y} e a terceira é ao redor \mathbf{X} .

Os resultados da base a cada rotação são apresentadas a seguir

$$\xi_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \xi_1 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \xi_2 = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}; \quad \xi_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad (4.13)$$

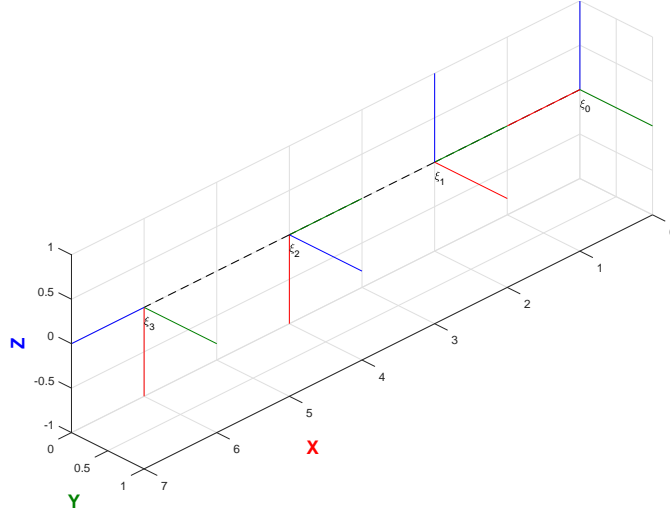


Figura 4.9: Três passos da rotação **ZYX**: $\phi_Z = 90^\circ$; $\theta_Y = 90^\circ$; $\psi_X = 90^\circ$.

Exemplo 8. A *Figura 4.10* apresenta os passos da rotação **ZYZ**. A primeira rotação é ao redor de **Z** a segunda rotação ao redor de **Y** e a terceira é novamente ao redor **Z**.

Os resultados da base a cada rotação são apresentadas a seguir

$$\xi_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \xi_1 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \xi_2 = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}; \quad \xi_3 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.14)$$

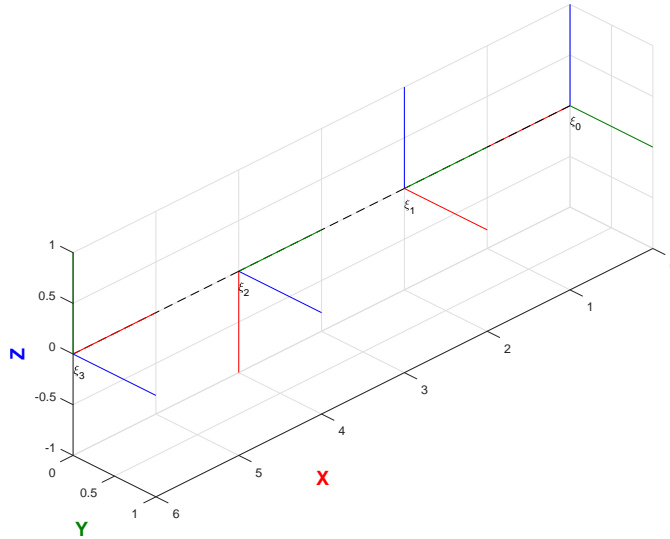


Figura 4.10: Três passos da rotação **ZYZ**: $\phi_Z = 90^\circ$; $\theta_Y = 90^\circ$; $\psi_Z = 90^\circ$.

Exercício 32. Obtenha a matriz de rotação $R_{xzy}(\phi, \psi, \theta) = R_x(\phi)R_z(\psi)R_y(\theta)$ do Exemplo 5.

Exercício 33. Obtenha a matriz de rotação $R_{yzx}(\phi, \theta, \psi) = R_y(\phi)R_x(\theta)R_z(\psi)$ do Exemplo 6.

Exercício 34. Obtenha a matriz de rotação $R_{zyx}(\psi, \phi, \theta) = R_z(\psi)R_y(\phi)R_x(\theta)$ do Exemplo 7.

Exercício 35. Obtenha a matriz de rotação $R_{zyz}(\psi, \phi, \theta) = R_z(\psi)R_y(\phi)R_z(\theta)$ do Exemplo 8.

Exercício 36. Refaça o Exemplo 5 com $\theta_X = 90^\circ$, $\phi_Z = 45^\circ$, $\psi_Y = 90^\circ$.

Exercício 37. Refaça o Exemplo 6 com $\theta_Y = 90^\circ$, $\phi_X = 45^\circ$, $\psi_Z = 90^\circ$.

Exercício 38. Refaça o Exemplo 7 com $\theta_Z = 90^\circ$, $\phi_Y = 45^\circ$, $\psi_X = 90^\circ$.

Exercício 39. Refaça o Exemplo 8 com $\theta_Z = 90^\circ$, $\phi_Y = 45^\circ$, $\psi_Z = 90^\circ$.

4.3 Rotações 3D - Teorema de Euler

Até a [Seção 4.2](#) foi entendido como as rotações são representadas e executadas com uso de matrizes de rotação e ângulos de Euler. A seguir veremos uma forma de realizar as rotações utilizando um eixo de rotação e um ângulo de rotação ao redor desse eixo.

4.3.1 Teorema de Euler

Qualquer posicionamento de um corpo rígido tal que um ponto o do corpo permaneça fixo, é equivalente à uma rotação sobre um eixo fixo através do ponto o . Sejam \underline{p} e \underline{q} dois pontos no espaço \mathbb{R}^3 , os vetores que descrevem as posições entre esses pontos e a origem o são

$$\begin{aligned}\vec{o\bar{p}} &= p_1 e_1 + p_2 e_2 + p_3 e_3 = \underline{p} \\ \vec{o\bar{q}} &= q_1 e_1 + q_2 e_2 + q_3 e_3 = \underline{q}\end{aligned}\quad (4.15)$$

Considerando que o ponto \underline{q} foi obtido ao se rotacionar o ponto \underline{p} mantendo o ponto o fixo, então é possível dizer que a relação entre os vetores \underline{q} e \underline{p} é dada por

$$\underline{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = R \underline{p} \quad (4.16)$$

onde $R \in \mathbb{R}^3$ e denota a matriz de rotação que rotaciona \underline{p} em \underline{q} . A [Figura 4.11](#) apresenta a situação descrita.³

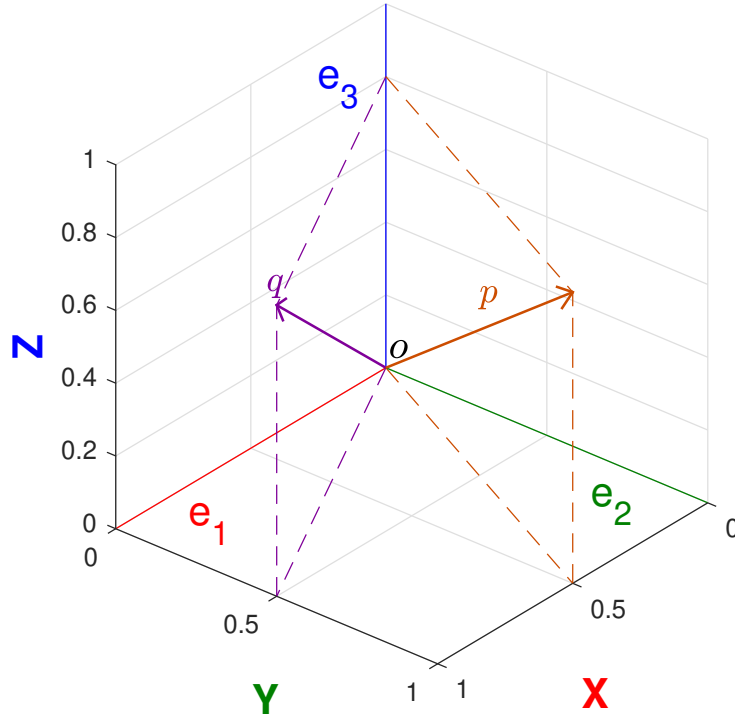


Figura 4.11: Ilustração do Teorema de Euler.

A pergunta que se faz é: existe um ponto \underline{p} tal que a rotação desse ponto resulte nele mesmo? Ou seja, se

$$\underline{p} = R \underline{p} \quad (4.17)$$

A resposta é sim, pois isso é equivalente a perguntar qual é o autovalor da matriz R .

Exercício 40. Verifique que $\lambda = 1$ é autovalor para qualquer matriz de rotação R da Eq. (4.17).

³O código gerador da figura encontra-se no Apêndice D.5, Algoritmo D.37.

Exercício 41. Mostre que se uma matriz M for usada para rotação, conforme indicado na Eq. (4.16), e tiver determinante (a) maior que 1, então o vetor resultante da operação terá norma maior que a norma do vetor original e (b) se for menor que 1, então a norma do vetor resultante será menor que a norma do inicial.

O teorema que prova que a existência da rotação de um vetor sobre um eixo dado um ângulo é conhecido como *Teorema (da rotação) de Euler*⁴ apresentado a seguir.

Teorema 1. Se $R \in \mathbb{R}^3$ que satisfaz $R^\top R = RR^\top = I$ e $\det(R) = 1$, então existe um vetor não nulo \vec{v} que satisfaz a relação $\vec{v} = R\vec{v}$.

Uma prova simplificada desse teorema é apresenta em Palais e Palais (2007).

4.4 Rotações 3D - Fórmula de Rodrigues

O problema inverso em relação ao apresentado na Seção 4.3 se constitui da seguinte pergunta: *Dado um eixo e um ângulo de rotação, como se encontra a matriz de rotação R ?*

Já sabemos como fazer isso se o eixo de rotação for um dos eixos coordenados (**X**, **Y**, **Z**), todavia, deseja-se agora generalizar a rotação de um dado ângulo ao redor de qualquer eixo arbitrário que seja fornecido.

A nomenclatura utilizada ao longo dessa seção é apresentada a seguir:

- \vec{n} : Vetor pivô da rotação;
- \vec{v} : Vetor a ser rotacionado;
- \vec{v}_\perp : Componente do vetor \vec{v} perpendicular ao vetor normal \hat{n} ;
- \vec{v}_\parallel : Componente do vetor \vec{v} paralela ao vetor normal \hat{n} ;
- \vec{v}' : Resultado da rotação do vetor \vec{v} ao redor do vetor \hat{n} por um ângulo ϕ ;
- \vec{v}'_\perp : Resultado da rotação do vetor \vec{v}_\perp ao redor do vetor \hat{n} por um ângulo ϕ ;
- \vec{v}'_\parallel : Resultado da rotação do vetor \vec{v}_\parallel ao redor do vetor \hat{n} por um ângulo ϕ ;
- \vec{w} : Resultado do produto vetorial $\hat{n} \times \vec{v}$.

Seja \hat{n} um vetor unitário, i.e., $\|\hat{n}\| = 1$, que representa um eixo de rotação arbitrário e seja ϕ um ângulo de rotação ao redor de \hat{n} .

Seja \vec{v} um vetor conectado no corpo rígido que deseja-se rotacionar por um ângulo ϕ ao redor de \hat{n} . As componentes paralela e perpendicular de \vec{v} em relação à \hat{n} são respectivamente \vec{v}_\parallel e \vec{v}_\perp dadas por

$$\vec{v}_\parallel = \langle \vec{v}, \hat{n} \rangle \cdot \hat{n} \quad (4.18)$$

$$\vec{v}_\perp = \vec{v} - \vec{v}_\parallel = \vec{v} - \langle \vec{v}, \hat{n} \rangle \cdot \hat{n} \quad (4.19)$$

Pode-se afirmar que o resultado da rotação de \vec{v} será a soma das rotações de cada componente. O resultado da rotação de cada vetor é denotado pela adição do símbolo ' ao vetor rotacionado, ou seja

$$\vec{v}' = \vec{v}'_\parallel + \vec{v}'_\perp \quad (4.20)$$

O vetor \vec{v}_\parallel permanece estático durante a rotação, o que faz sentido, dado que esse vetor é paralelo ao eixo de rotação e o eixo de rotação permanece estático, assim, tem-se a relação

$$\vec{v}'_\parallel = \vec{v}_\parallel \quad (4.21)$$

Para determinar o resultado da rotação da componente perpendicular \vec{v}_\perp . Seja \vec{v}'_\perp o resultado da rotação de \vec{v}_\perp . Essa rotação ocorrerá em um plano paralelo a \vec{v}_\perp e perpendicular a \hat{n} (BIASI; GATTASS, 2002).

$$\vec{w} = \hat{n} \times \vec{v}_\perp = \hat{n} \times (\vec{v} - \vec{v}_\parallel) = \hat{n} \times \vec{v} - \hat{n} \times \vec{v}_\parallel = \hat{n} \times \vec{v} \quad (4.22)$$

Decompondo a rotação do vetor perpendicular no plano formado por \vec{v}_\perp e \vec{w}

$$\vec{v}'_\perp = \cos(\phi)\vec{v}_\perp + \sin(\phi)\vec{w} \quad (4.23)$$

⁴Euler possui diversos teoremas espalhados pela matemática. A expressão “Teorema de Euler” é surpreendentemente ambígua (WIKIPÉDIA, 2019c).

O resultado das rotações de \vec{v}_{\parallel} e de \vec{v}_{\perp} pode ser reescrito a partir Equações (4.18), (4.21), (4.22) e (4.23) em função de \vec{v} , \hat{n} e ϕ

$$\vec{v}'_{\parallel} = \langle \vec{v}, \hat{n} \rangle \cdot \hat{n} \quad (4.24)$$

$$\vec{v}'_{\perp} = \cos(\phi) \vec{v}_{\perp} + \sin(\phi) (\hat{n} \times \vec{v}) \quad (4.25)$$

Portanto, o resultado da rotação de \vec{v} é

$$\vec{v}' = \vec{v}'_{\parallel} + \vec{v}'_{\perp} \quad (4.26)$$

$$= \langle \vec{v}, \hat{n} \rangle \cdot \hat{n} + \cos(\phi) \vec{v}_{\perp} + \sin(\phi) (\hat{n} \times \vec{v}) \quad (4.27)$$

$$= \langle \vec{v}, \hat{n} \rangle \cdot \hat{n} + \cos(\phi) (\vec{v} - \langle \vec{v}, \hat{n} \rangle \cdot \hat{n}) + \sin(\phi) (\hat{n} \times \vec{v}) \quad (4.28)$$

$$= \langle \vec{v}, \hat{n} \rangle \cdot \hat{n} + \cos(\phi) \vec{v} - \cos(\phi) \langle \vec{v}, \hat{n} \rangle \cdot \hat{n} + \sin(\phi) \hat{n} \times \vec{v} \quad (4.29)$$

A Figura 4.12 apresenta a rotação do vetor \vec{v} ao redor do vetor normal \hat{n} visualizando a componente paralela a \hat{n} , \vec{v}_{\parallel} , e a componente perpendicular a \hat{n} , \vec{v}_{\perp} (BIASI; GATTASS, 2002).⁵

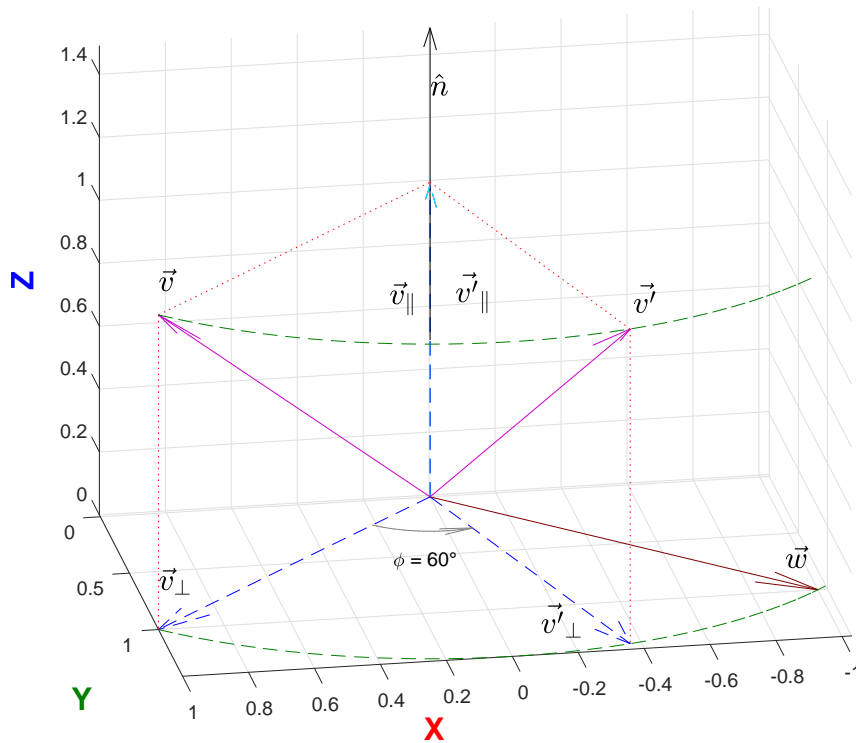


Figura 4.12: Rotação do vetor \vec{v} , decomposto em suas projeções ortogonal e perpendicular ao vetor \hat{n} , cujo resultado é o vetor \vec{v}' . Repare que as projeções paralelas são iguais antes e após a rotação ao redor de \hat{n} .

Definição 1 (Matriz do produto vetorial): $\overset{\times}{u} := (\vec{u} \times)$, é a matriz⁶ que produz o mesmo resultado de se aplicar o produto vetorial $(\vec{u} \times)$ ao vetor \vec{v} .

$$\overset{\times}{u} = \begin{bmatrix} 0 & -u_3 & +u_2 \\ +u_3 & 0 & -u_1 \\ -u_2 & +u_1 & 0 \end{bmatrix} \quad (4.30)$$

Antes de prosseguirmos, verifique o formato da matriz M_1 da Eq. (3.9). Existe alguma semelhança entre M_1 e a matriz $\overset{\times}{u}$ recém definida?

⁵O código gerador da figura encontra-se no Apêndice D.5, Algoritmo D.38.

⁶Alguns livros denotam essa matriz por \tilde{u} e utilizam a nomenclatura de *tensor (antissimétrico do produto vetorial)*, e.g., (BÖGE et al., 2017).

Exercício 42. Para $\vec{u} = [u_1, u_2, u_3]^\top$ e $\vec{v} = [v_1, v_2, v_3]^\top$. Mostre que

$$\vec{u} \times \vec{v} = \begin{bmatrix} 0 & -u_3 & +u_2 \\ +u_3 & 0 & -u_1 \\ -u_2 & +u_1 & 0 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \overset{\times}{u} \cdot \vec{v}. \quad (4.31)$$

Exercício 43. Mostre que se o vetor \vec{u} de $\overset{\times}{u}$ for unitário, i.e., $\vec{u} = \hat{u}$, então $\|\overset{\times}{u}\|_2 = 1$. **Dica:** Seja $A \in \mathbb{M}(n, \mathbb{R})$ então $\|A\|_2 = \sigma_{\max}(A)$, onde σ é o conjunto dos valores singulares de A . $\|A\|_2 = \sqrt{\lambda_{\max}(A^* A)}$.

Exercício 44. Mostre que $\langle \vec{v}, \vec{u} \rangle \vec{u} = \vec{u} \vec{u}^\top \vec{v}$, $\forall \vec{u}, \vec{v} \in \mathbb{R}^3$.

Retomando \vec{v}' com uso do resultado de (4.30)

$$\vec{v}' = \cos(\phi) \vec{v} + (1 - \cos(\phi)) \langle \vec{v}, \hat{n} \rangle \cdot \hat{n} + \sin(\phi) \hat{n} \times \vec{v} \quad (4.32)$$

$$= \cos(\phi) \vec{v} + \hat{n} \hat{n}^\top (1 - \cos(\phi)) \vec{v} + \sin(\phi) \overset{\times}{n} \vec{v} \quad (4.33)$$

Como \vec{v}' foi obtido a partir de uma rotação a partir de \vec{v} , pode-se isolar a matriz de rotação R

$$\vec{v}' = \left(\cos(\phi) I + \hat{n} \hat{n}^\top (1 - \cos(\phi)) + \sin(\phi) \overset{\times}{n} \right) \vec{v} = R \vec{v} \quad (4.34)$$

$$\Rightarrow R = \text{Rot}(\hat{n}, \phi) = \cos(\phi) I + \hat{n} \hat{n}^\top (1 - \cos(\phi)) + \sin(\phi) \overset{\times}{n} \quad (4.35)$$

A matriz $\text{Rot}(\hat{n}, \phi)$ apresentada em (4.35) é conhecida como *Fórmula (de rotação) de Rodrigues*⁷.

Exemplo 9. Exemplo da Fórmula de Rodrigues. Para \hat{n} , ϕ e \vec{v} dados por

$$\hat{n} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \quad \phi = \frac{\pi}{2}; \quad \vec{v} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.36)$$

(a) Encontrar a matriz de rotação $R = \text{Rot}(\hat{n}, \phi)$ ao redor de \hat{n} para ângulo arbitrário ϕ . (b) Aplicar a matriz de rotação R para o ângulo ϕ dado. (c) Calcular o resultado da rotação de \vec{v} .

A Fórmula de Rodrigues apresentada em (4.35) é $\text{Rot}(\hat{n}, \phi) = \cos(\phi) I + \hat{n} \hat{n}^\top (1 - \cos(\phi)) + \sin(\phi) \overset{\times}{n}$ de onde tem-se

$$\begin{aligned} \hat{n} \cdot \hat{n}^\top &= \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} \cdot \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad \overset{\times}{n} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \\ R &= \begin{bmatrix} \cos(\phi) & 0 & 0 \\ 0 & \cos(\phi) & 0 \\ 0 & 0 & \cos(\phi) \end{bmatrix} + \begin{bmatrix} 1 - \cos(\phi) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\sin(\phi) \\ 0 & \sin(\phi) & 0 \end{bmatrix} \\ R &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} = R_x(\phi) \end{aligned} \quad (4.37)$$

Para o caso particular de $\phi = 90^\circ$, a matriz de rotação R obtida pela Fórmula de Rodrigues é

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.38)$$

O resultado da rotação de \vec{v} ao redor de \hat{n}

$$\vec{v}' = R \vec{v} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \quad (4.39)$$

⁷Existe uma *Fórmula de Rodrigues* para os polinômios de Legendre que é formalmente denominada *Fórmula de Ivory-Jacobi* (WIKIPÉDIA, 2019d).

Código 22. Faça uma função que retorne $\text{Rot}(\hat{n}, \phi)$ a partir da Fórmula de Rodrigues.

$$R = \text{fc_Rot_Rodrigues}(n, \phi)$$

Exercício 45. Aplique a Fórmula de Rodrigues para $\hat{n} = \hat{j} = \hat{e}_2$.

Exercício 46. Aplique a Fórmula de Rodrigues para $\hat{n} = \hat{k} = \hat{e}_3$.

Exercício 47. Mostre que $\text{Rot}(\hat{n}, \phi) = \text{Rot}(-\hat{n}, 2\pi - \phi)$. Esse mapa é do tipo “um-para-um” (1-1)?

Para uma dada matriz de rotação $R \in \text{SO}(3)$ é possível recuperar o vetor \hat{n} e o ângulo ϕ .

$$R = \text{Rot}(\hat{n}, \phi) = \begin{bmatrix} c - n_1^2 (c - 1) & -n_3 s - n_1 n_2 (c - 1) & n_2 s - n_1 n_3 (c - 1) \\ n_3 s - n_1 n_2 (c - 1) & c - n_2^2 (c - 1) & -n_1 s - n_2 n_3 (c - 1) \\ -n_2 s - n_1 n_3 (c - 1) & n_1 s - n_2 n_3 (c - 1) & c - n_3^2 (c - 1) \end{bmatrix} \quad (4.40)$$

onde $c = \cos(\phi)$ e $s = \sin(\phi)$.

O traço Tr de uma matriz é a soma dos elementos de sua diagonal principal. Para a matriz de rotação R , o traço é dado por

$$\begin{aligned} \text{Tr} = \text{Trace}(R) &= c + n_1^2(1 - c) + c - n_2^2(c - 1) + c - n_3^2(c - 1) \\ &= 3c + (1 - c) \underbrace{(n_1^2 + n_2^2 + n_3^2)}_{|\hat{n}|_2^2=1} = 3c + 1 - c = 2c + 1 \end{aligned} \quad (4.41)$$

$$\Rightarrow c = \cos(\phi) = \frac{\text{Tr} - 1}{2}$$

Seja $A = -A^\top$, uma matriz antissimétrica

$$\begin{aligned} A = R - R^\top &= \begin{bmatrix} 0 & -2n_3 s & 2n_2 s \\ 2n_3 s & 0 & -2n_1 s \\ -2n_2 s & 2n_1 s & 0 \end{bmatrix} = 2s \cdot \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix} \\ R - R^\top &= 2 \sin(\phi) \hat{n}^\times \\ \Rightarrow \hat{n}^\times &= \frac{1}{2 \sin(\phi)} \cdot (R - R^\top) \end{aligned} \quad (4.42)$$

Note que \hat{n}^\times contém toda a informação para \hat{n} , sem resolver o problema de autovalor.

O mapa (\hat{n}, ϕ) para a matriz de rotação é “muitos para 1”, porém ao se restringir $\phi \in [0, \pi]$ faz o mapa ser “1 para 1” exceto para:

$$\begin{aligned} \text{Tr} = 3 &\Rightarrow \phi = 0 \Rightarrow \nexists \text{ único eixo} \\ \text{Tr} = 1 &\Rightarrow \phi = \pi \Rightarrow \hat{n} \text{ ou } -\hat{n} \end{aligned} \quad (4.43)$$

Exercício 48. Obtenha o vetor \hat{n} e o ângulo ϕ correspondentes à matriz de rotação

$$R = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{2}}{4} + \frac{1}{2} & \frac{1}{2} - \frac{\sqrt{2}}{4} \\ -\frac{1}{2} & \frac{1}{2} - \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} + \frac{1}{2} \end{bmatrix} \quad (4.44)$$

Exercício 49. Considere a matriz de rotação R a seguir

$$R = \begin{bmatrix} 0,9512830 & 0,0365378 & y \\ 0,0365378 & x & -0,22961 \\ -y & 0,2296100 & 0,923880 \end{bmatrix} \quad (4.45)$$

Calcule o valor de x e y sabendo que o vetor normal de rotação foi $\hat{n} = [3, 4, 0]^\top / 5$ e o ângulo utilizado foi $\phi = \pi/8$.

Exercício 50. Esse exercício é um pouco mais desafiador que o anterior, e revela uma interessante relação entre o vetor normal \hat{n} e o ângulo de rotação ϕ . Sabendo que a normal do vetor normal é 1 e $x > 0$, obtenha o vetor normal \hat{n} e o ângulo de rotação ϕ a partir da matriz de rotação R a seguir.

$$R = \begin{bmatrix} \sqrt{3}/4 + 1/2 & -\sqrt{2}/4 & 1/2 - \sqrt{3}/4 \\ \sqrt{2}/4 & x & -\sqrt{2}/4 \\ 1/2 - \sqrt{3}/4 & \sqrt{2}/4 & \sqrt{3}/4 + 1/2 \end{bmatrix} \quad (4.46)$$

4.4.1 Entendendo o grupo de rotação $SO(3)$

Imagine uma esfera sólida de raio π , conforme exibido na [Figura 4.13](#). Todo o ponto na superfície ou dentro da esfera representa uma rotação, cujo ângulo é dado pelo raio. E o vetor que descreve o ponto cuja origem é o centro da esfera é o eixo da rotação ([ROBOTICS, 2017b](#)).

Aparentemente o grupo $SO(3)$ é uma mapa “1 para 1” entre os pontos na superfície ou dentro da esfera e o conjunto de rotações. Isto é verdadeiro para o ponto na superfície da esfera. Dois pontos diametralmente opostos correspondem à mesma rotação, ou seja

$$\text{Rot}(\hat{n}, \phi) = \text{Rot}(-\hat{n}, 2\pi - \phi) \quad (4.47)$$

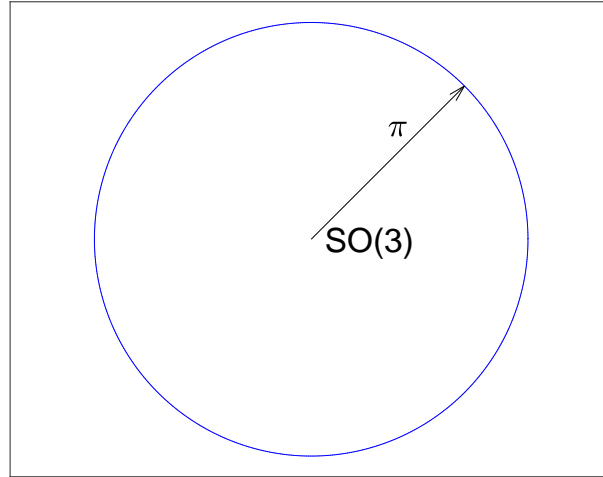


Figura 4.13: Representação do grupo $SO(3)$ como esfera de raio π .

4.5 Rotações 2D - Fórmula de Euler

Retomemos as rotações em 2 dimensões porém agora com uso da fórmula de Euler⁸ ([BIASI; GATTASS, 2002](#)).

Seja $z \in \mathbb{C}$, tem-se que a forma polar pode ser escrita como

$$z = r e^{i\theta}, \text{ com } \begin{cases} r^2 = x^2 + y^2 \\ \tan \theta = \frac{y}{x} \Rightarrow \theta = \text{atan2}(y, x) \end{cases} \quad (4.48)$$

Para um número complexo \hat{z} tal que $|\hat{z}| = r = 1$, tem-se o caso de rotação pura.

Exercício 51. Mostre que $e^{i\theta} = \cos(\theta) + i \sin(\theta)$. *Dica: Use a série de Taylor.*

Exercício 52. Mostre que a forma matricial de $\hat{z} \in \mathbb{C}$, com $|\hat{z}| = 1$, é a matriz de rotação sobre o eixo Z .

Exercício 53. Calcule:

- (a) $e^{\frac{\pi}{2}i}$,
- (b) i^i ,
- (c) z^n , tal que $z = a + bi$ com $a, b \in \mathbb{R}$,
- (d) $\ln(i)$,
- (e) $\ln(-5)$,
- (f) \sqrt{i} .

Exercício 54. Seja $z_1 = 1 + 5i$, calcule:

- (a) o ângulo θ_1 formado por z_1 e o eixo das abscissas,
- (b) o número complexo na forma cartesiana $z_2 = a + bi$, tal que z_2 realize uma rotação de 60° ,
- (c) $z_3 \in \mathbb{C}$ na forma cartesiana tal que z_3 seja o resultado da rotação de z_1 por um ângulo de 60° .

⁸O número $e = 2,718\dots$ representa a constante de Euler, esse número jamais será escrito com subíndice. O uso de subíndices fica restrito aos elementos da base $e_{(\cdot)}$.

4.6 Rotações 3D - Fórmula de Euler para Quatérnios

Vamos aplicar a fórmula de Euler para o quatérnio q que contém o escalar 0 e o vetor $\vec{v} = xi + yj + zk$, ou seja, $q = (0, \vec{v}) \in \mathbb{H}$. Essa seção é baseada principalmente em (MATHOMA, 2015; ACADEMICS, 2014).

Relembrando o produto de quatérnios

$$\begin{aligned} q_1 q_2 &= (a_1, \vec{v}_1) (a_2, \vec{v}_2) \\ &= (a_1 a_2 - \langle \vec{v}_1, \vec{v}_2 \rangle, a_1 \vec{v}_2 + a_2 \vec{v}_1 + \vec{v}_1 \times \vec{v}_2) \end{aligned} \quad (4.49)$$

quando $a_1 = 0$ e $a_2 = 0$ tem-se

$$q_1 q_2 = (-\langle \vec{v}_1, \vec{v}_2 \rangle, \vec{v}_1 \times \vec{v}_2) \quad (4.50)$$

com base nesse resultado, calculemos agora q^2

$$q^2 = q q = (-\langle \vec{v}, \vec{v} \rangle, \vec{v} \times \vec{v}) = -\|\vec{v}\|^2 \quad (4.51)$$

as demais potenciais podem ser calculadas com base nas anteriores

$$q^3 = q^2 q = -\|\vec{v}\|^2 q = -\|\vec{v}\|^2 (0, \vec{v}) = -\|\vec{v}\|^2 \cdot \vec{v} \quad (4.52)$$

$$q^4 = q^2 q^2 = (-\|\vec{v}\|^2) \cdot (-\|\vec{v}\|^2) = \|\vec{v}\|^4 \quad (4.53)$$

Exemplo 10. Seja $q = (0, [1, 1, 1]^\top) \in \mathbb{H}$

$$(a) q^2 = -(1^2 + 1^2 + 1^2) = -3,$$

$$(b) q^3 = (-3) \cdot [1, 1, 1]^\top = -3i - 3j - 3k$$

$$(c) q^4 = (-3) \cdot (-3) = 9$$

Assim concluí-se que, com $\alpha, \beta \in \mathbb{N}$, $q^\alpha = q^{2\beta}$ é sempre um escalar e $q^\alpha = q^{2\beta+1}$ é sempre um escalar multiplicado pelo vetor \vec{v} que compõe q .

Então podemos agora calcular e^q para $q = (0, \vec{v}) \in \mathbb{H}$ e para isso utilizamos a Série de Taylor

$$\begin{aligned} e^q &= 1 + q + \frac{q^2}{2} + \frac{q^3}{3!} + \frac{q^4}{4!} + \dots \\ &= 1 + \vec{v} - \frac{\|\vec{v}\|^2}{2} + \frac{\|\vec{v}\|^2}{3!} \vec{v} + \frac{\|\vec{v}\|^4}{4!} - \frac{\|\vec{v}\|^4}{5!} \vec{v} - \frac{\|\vec{v}\|^6}{6!} + \frac{\|\vec{v}\|^6}{7!} \vec{v} + \dots \\ &= \left(1 - \frac{\|\vec{v}\|^2}{2} + \frac{\|\vec{v}\|^4}{4!} - \frac{\|\vec{v}\|^6}{6!} + \dots\right) + \left(1 - \frac{\|\vec{v}\|^2}{3!} + \frac{\|\vec{v}\|^4}{5!} - \frac{\|\vec{v}\|^6}{7!} + \dots\right) \vec{v} \\ &= \cos(\|\vec{v}\|) + \frac{\|\vec{v}\|}{\|\vec{v}\|} \left(1 - \frac{\|\vec{v}\|^2}{3!} + \frac{\|\vec{v}\|^4}{5!} - \frac{\|\vec{v}\|^6}{7!} + \dots\right) \vec{v} \\ &= \cos(\|\vec{v}\|) + \left(\|\vec{v}\| - \frac{\|\vec{v}\|^3}{3!} + \frac{\|\vec{v}\|^5}{5!} - \frac{\|\vec{v}\|^7}{7!} + \dots\right) \frac{\vec{v}}{\|\vec{v}\|} \\ &= \cos(\|\vec{v}\|) + \frac{\sin(\|\vec{v}\|)}{\|\vec{v}\|} \vec{v} \\ &= \cos(\|\vec{v}\|) + \text{sen}(\|\vec{v}\|) \hat{v} \end{aligned} \quad (4.54)$$

retomando $q = (0, \vec{v}) = xi + yj + zk$, com $\|\vec{v}\| = \sqrt{x^2 + y^2 + z^2}$ tem-se

$$e^q = e^{xi+yj+zk} = \cos(\|\vec{v}\|) + \frac{\sin(\|\vec{v}\|)}{\|\vec{v}\|} \cdot (xi + yj + zk) \in \mathbb{H} \quad (4.55)$$

Adicionemos a parte escalar $S(q) = a \in \mathbb{R}$. O resultado da exponencial convencional com expoente $a + bi \in \mathbb{C}$ é simplesmente a separação da soma no expoente

$$e^{a+bi} = e^a \cdot e^{bi} = e^a \cdot (\cos(b) + i \sin(b)) \quad (4.56)$$

Para $q = a + xi + yj + zk \in \mathbb{H}$ tem-se a separação entre a parte escalar a e a vetorial $\vec{v} = xi + yj + zk$ cuja norma é $\|\vec{v}\| = \sqrt{x^2 + y^2 + z^2}$

$$e^q = e^{a+xi+yj+zk} = e^a \cdot e^{xi+yj+zk} = e^a \cdot \left(\cos(\|\vec{v}\|) + \frac{\sin(\|\vec{v}\|)}{\|\vec{v}\|} \vec{v}\right) \quad (4.57)$$

Exemplo 11. Aplicações da exponencial

(a)

$$e^{i+j} = \cos(\sqrt{2}) + \frac{\sin(\sqrt{2})}{\sqrt{2}}(i+j) \approx 0,156 + 0,698(i+j) \quad (4.58)$$

(b)

$$e^{3j+4k} = \cos(5) + \frac{\sin(5)}{5}(3j+4k) \approx 0,284 - 0,575j - 0,767k \quad (4.59)$$

(c)

$$e^{ix} = \cos(x) + \frac{\sin(x)}{x}(ix) = \cos(x) + i\sin(x) \quad (4.60)$$

Repare que a exponencial de quatérnio recupera a fórmula de Euler.

Código 23. Crie uma função que retorne a exponencial de um quatérnio.

`r = fc_quat_exp(q)`

Exercício 55. Demonstre que $e^{a+xi+yj+zk} = e^a \cdot e^{xi+yj+zk}$ através da série de potências.

Exercício 56. Prove que a igualdade $e^{i+j} = e^i \cdot e^j$ é falsa.

Qual o significado de e^{i+j} ? E qual o significado de $e^i e^j$? Faz sentido as duas operações terem resultados diferentes?

Exercício 57. (a) Mostre que $e^{i\pi} = e^{j\pi} = e^{k\pi} = -1$,

(b) Mostre que

$$e^{\frac{\pi}{\sqrt{2}}i + \frac{\pi}{\sqrt{2}}j} = e^{\frac{\pi}{\sqrt{3}}(i+j+k)} = -1,$$

(c) Para quantos quatérnios vale a relação $e^q = -1$?

(d) Mostre que $\forall q \in \mathbb{H}$ tal que $q = |q| \cdot e^{\theta \hat{n}}$ tem-se a relação da parte escalar e da parte vetorial com o módulo do quatérnio dados por $a = |q| \cos(\theta)$ e $\vec{v} = \|\vec{v}\| \hat{n} = |q| \sin(\theta) \hat{n}$,

(e) Mostre que $\forall k \in \mathbb{R}, \forall q \in \mathbb{H}$

$$q^k = |q|^k \cdot e^{k\theta \hat{n}} = |q|^k \cdot (\cos(k\theta) + \sin(k\theta) \cdot \hat{n}), \quad (4.61)$$

(f) Calcule e^q para $q = 2 + 1i + 2j + 4k$.

4.7 Rotações 3D - Fórmula de Rodrigues para Quatérnios

Vamos então rever o problema de rotação um vetor \vec{v} sobre um eixo de rotação \hat{n} e ângulo ϕ . Essa seção é baseada principalmente em (MATHOMA, 2016).

Como consequência da Equação (4.25) tem-se que a decomposição do vetor rotacionado \vec{v}' em função de \vec{v} e da projeção ortogonal $\hat{n} \times \vec{v}$

$$\vec{v}' = \cos(\phi) \vec{v} + \sin(\phi) (\hat{n} \times \vec{v}) \quad (4.62)$$

Vamos aplicar quatérnios ao problema.

Seja $q = (0, \vec{v}) \in \mathbb{H}$ o quatérnio que contém o vetor \vec{v} a ser rotacionado. Seja $p = (0, \hat{n}) \in \mathbb{H}$ o quatérnio que contém o vetor \hat{n} do eixo de rotação. Repare que $|p| = \|\hat{n}\| = 1$, logo p é um quatérnio unitário.

A multiplicação de p e q é

$$pq = (0, \hat{n}) \cdot (0, \vec{v}) = (0 - \langle \hat{n}, \vec{v} \rangle, 0\vec{v} + 0\hat{n} + \hat{n} \times \vec{v}) = (-\langle \hat{n}, \vec{v} \rangle, \hat{n} \times \vec{v}) \quad (4.63)$$

mas pela construção de \hat{n} e \vec{v} tem-se

$$\hat{n} \perp \vec{v} \Rightarrow \langle \hat{n}, \vec{v} \rangle = 0 \Rightarrow pq = (0, \hat{n} \times \vec{v}) \quad (4.64)$$

abusando da notação tem-se $pq = \hat{n} \times \vec{v}$, e definindo $r = (0, \vec{v}')$, então

$$\begin{aligned}
 \vec{v}' &= \cos(\phi)\vec{v} + \sin(\phi)(\hat{n} \times \vec{v}) \\
 &= \cos(\phi)q + \sin(\phi)pq \\
 &= (\cos(\phi) + \sin(\phi)p)q \\
 &= \left(\cos(\phi) + \sin(\phi)\frac{\vec{n}}{\|\vec{n}\|} \right) q; \quad \phi = \|\vec{n}\| \\
 r &= (e^{\phi\hat{n}})q
 \end{aligned} \tag{4.65}$$

Da definição de vetor unitário e da relação $\phi = \|\vec{n}\|$

$$\hat{n} = \frac{\vec{n}}{\|\vec{n}\|} \Rightarrow \vec{n} = \|\vec{n}\|\hat{n} = \phi\hat{n} \tag{4.66}$$

então é possível determinar que

$$e^{\phi\hat{n}} = e^{\vec{n}} = e^{(0, \vec{n})} = e^s = p \tag{4.67}$$

e com base no resultado de (4.55), define-se $s = (0, \vec{n}) \in \mathbb{H}$.

Exemplo 12. Rotação de \hat{e}_3 ao redor de \hat{e}_2 por um ângulo $\phi = \frac{\pi}{2}$ rad.

Sejam $p, q \in \mathbb{H}$ tais que:

$$q = (0, \hat{e}_3) = (0, [0, 0, 1]^\top) = k, \quad \vec{v} = k = \hat{e}_3 \tag{4.68}$$

$$p = e^{\phi\hat{n}}, \quad \hat{n} = j = \hat{e}_2, \quad \phi = \frac{\pi}{2} \tag{4.69}$$

O resultado da rotação utilizando a exponencial de quatérnio é

$$\begin{aligned}
 r &= pq = e^{\phi\hat{n}}q \\
 &= (\cos(\phi) + \sin(\phi)\hat{n})q \\
 &= \left(\cos\left(\frac{\pi}{2}\right) + \sin\left(\frac{\pi}{2}\right)\hat{n} \right) q \\
 &= (0 + 1j) \cdot k = j \cdot k \\
 &= \hat{e}_2 \cdot \hat{e}_3 = \hat{e}_1 = i \\
 \therefore r &= \hat{e}_3' = \hat{e}_1
 \end{aligned} \tag{4.70}$$

O resultado é apresentado na Figura 4.14.

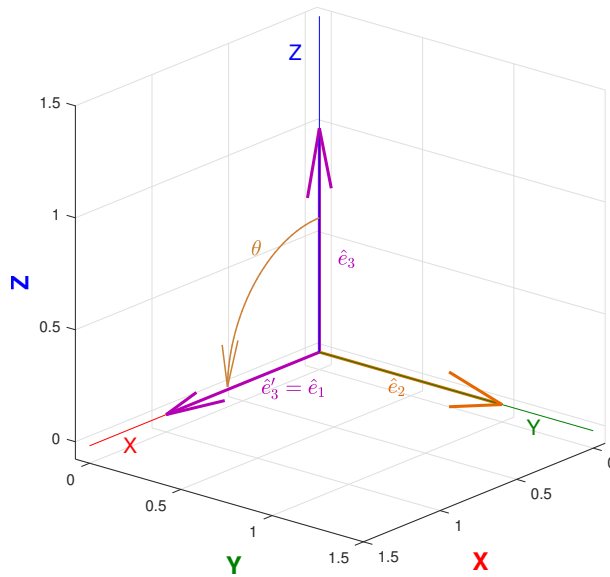


Figura 4.14: Resultado da rotação de \hat{e}_3 ao redor de \hat{e}_2 por um ângulo de $\frac{\pi}{2}$ rad.

Exercício 58. Mostre que $\forall q, p \in \mathbb{H}$ tais que $q = (0, \vec{v})$ e $p = (0, \hat{n})$ com $\phi \in \mathbb{R}$, $\vec{v}, \hat{n} \in \mathbb{R}^3$ e $\hat{n} \perp \vec{v}$ tem-se:

$$e^{\phi p} q = q e^{-\phi p}. \quad (4.71)$$

Exercício 59. Mostre que $\forall q, p \in \mathbb{H}$ tais que $q = (0, \vec{u})$ e $p = (0, \hat{n})$ com $\theta \in \mathbb{R}$, $\vec{u}, \hat{n} \in \mathbb{R}^3$ e $\hat{n} \parallel \vec{u}$ tem-se:

$$e^{\theta p} q = q e^{\theta p}. \quad (4.72)$$

Retomemos a explicação de rotação sobre um eixo para a *Fórmula de Rodrigues*.

Seja \vec{v} o vetor a ser rotacionado ao redor de \hat{n} , e ele seja decomposto nas componentes paralela e perpendicular a \hat{n} . O resultado da rotação \vec{v}' será a soma de cada componente rotacionada

$$\begin{aligned} \vec{v} &= \vec{v}_{\parallel} + \vec{v}_{\perp} \\ \vec{v}' &= \vec{v}'_{\parallel} + \vec{v}'_{\perp} \end{aligned} \quad (4.73)$$

relembrando o fato de que a componente paralela ao eixo de rotação permanece inalterada após a rotação, ou seja

$$\vec{v}'_{\parallel} = \vec{v}_{\parallel} \quad (4.74)$$

Sejam $p, q, r \in \mathbb{H}$, $q = (0, \vec{v})$, $p = (0, \hat{n})$, $r = (0, \vec{v}')$, tem-se

$$r = pq = e^{\phi \hat{n}} \cdot q = e^{\phi \hat{n}} \cdot (0, \vec{v}) = e^{\phi \hat{n}} \cdot (0, \vec{v}_{\parallel} + \vec{v}_{\perp}) \quad (4.75)$$

Então, por extensão, vamos definir os respectivos quatérnios das componentes paralela e perpendicular de \vec{v} e \vec{v}'

$$q = (0, \vec{v}), \quad q_{\parallel} = (0, \vec{v}_{\parallel}), \quad q_{\perp} = (0, \vec{v}_{\perp}) \quad \Rightarrow q = q_{\parallel} + q_{\perp} \quad (4.76)$$

$$r = (0, \vec{v}'), \quad r_{\parallel} = (0, \vec{v}'_{\parallel}), \quad r_{\perp} = (0, \vec{v}'_{\perp}) \quad \Rightarrow r = r_{\parallel} + r_{\perp} \quad (4.77)$$

com base nos resultados anteriores, incluindo (4.65), tem-se que

$$r = r_{\parallel} + r_{\perp} = q_{\parallel} + e^{\phi \hat{n}} q_{\perp} = q_{\parallel} + \cos(\phi) q_{\perp} + \sin(\phi) pq_{\perp} \quad (4.78)$$

que é a *Fórmula de Rodrigues*.

Por notação, sejam $c_{\phi} = \cos(\phi)$ e $s_{\phi} = \sin(\phi)$. O resultado de (4.78) é compactado como

$$r = q_{\parallel} + c_{\phi} q_{\perp} + s_{\phi} pq_{\perp} \quad (4.79)$$

O primeiro passo para obtermos a *Fórmula de Rodrigues* da (4.35) é analisar o termo pq_{\perp}

$$\begin{aligned} pq_{\perp} &= p(q - q_{\parallel}) = pq - pq_{\parallel} \\ pq &= (0, \hat{n}) \cdot (0, \vec{v}) = \hat{n} \times \vec{v} = \hat{n} \times \vec{v} \\ pq_{\parallel} &= (0, \hat{n}) \cdot (0, \vec{v}_{\parallel}) = \hat{n} \times \vec{v}_{\parallel} = \mathbf{0}_{\mathbb{R}^3} \\ \Rightarrow s_{\phi} pq_{\perp} &= s_{\phi} \cdot pq = s_{\phi} \cdot \hat{n} \times \vec{v} \end{aligned} \quad (4.80)$$

Em seguida, apresentamos o termo q_{\parallel} , que está relacionado com o produto interno

$$q_{\parallel} = (0, \vec{v}_{\parallel}) = (0, \langle \hat{n}, \vec{v} \rangle \hat{n}) = (0, \hat{n} \langle \hat{n}, \vec{v} \rangle) = (0, \hat{n} \hat{n}^{\top} \vec{v}) \quad (4.81)$$

Então com os resultados parciais, recupera-se a *Fórmula de Rodrigues* na forma matricial-vetorial

$$\begin{aligned} r &= pq = e^{\phi \hat{n}} q = q_{\parallel} + (c_{\phi} + s_{\phi} p) q_{\perp} \\ &= (0, \hat{n} \hat{n}^{\top} \vec{v}) + c_{\phi} q_{\perp} + s_{\phi} pq_{\perp} \\ &= (0, \hat{n} \hat{n}^{\top} \vec{v}) + c_{\phi} (q - q_{\parallel}) + s_{\phi} \cdot \hat{n} \times \vec{v} \\ &= (0, \hat{n} \hat{n}^{\top} \vec{v}) + c_{\phi} (0, \vec{v}) - c_{\phi} (0, \hat{n} \hat{n}^{\top} \vec{v}) + (0, s_{\phi} \cdot \hat{n} \times \vec{v}) \\ &= \left(0, \left(c_{\phi} I + (1 - c_{\phi}) \hat{n} \hat{n}^{\top} + s_{\phi} \hat{n} \times \right) \vec{v} \right) \\ &= (0, \text{Rot}(\hat{n}, \phi) \cdot \vec{v}) \end{aligned} \quad (4.82)$$

A *Fórmula de Rodrigues* ainda pode ser escrita em uma **forma compacta** com quatérnios. Retomando a partir de (4.78) tem-se

$$\begin{aligned} r &= q_{\parallel} + e^{\phi \hat{n}} q_{\perp} \\ &= e^{\frac{\phi}{2} \hat{n}} \cdot \underbrace{e^{-\frac{\phi}{2} \hat{n}} q_{\parallel}}_{\text{comutativo}} + e^{\frac{\phi}{2} \hat{n}} \cdot \underbrace{e^{-\frac{\phi}{2} \hat{n}} q_{\perp}}_{\text{anticomutativo}} \end{aligned} \quad (4.83)$$

com o resultado do Exercício 58 para o caso anticomutativo e do Exercício 59 para o caso comutativo tem-se

$$\begin{aligned} r &= e^{\frac{\phi}{2} \hat{n}} q_{\parallel} e^{-\frac{\phi}{2} \hat{n}} + e^{\frac{\phi}{2} \hat{n}} q_{\perp} e^{-\frac{\phi}{2} \hat{n}} \\ &= e^{\frac{\phi}{2} \hat{n}} (q_{\parallel} + q_{\perp}) e^{-\frac{\phi}{2} \hat{n}} \\ &= e^{\frac{\phi}{2} \hat{n}} q e^{-\frac{\phi}{2} \hat{n}} \end{aligned} \quad (4.84)$$

o que indica que não é necessário decompor nas componentes ortogonal e paralela para se realizar a rotação com uso de quatérnios.

Seja $s \in \mathbb{H}$ e s^* seu conjugado tais que

$$\begin{aligned} s &= e^{\frac{\phi}{2} \hat{n}} = \cos\left(\frac{\phi}{2}\right) + \sin\left(\frac{\phi}{2}\right) \hat{n} \\ s^* &= e^{-\frac{\phi}{2} \hat{n}} = \cos\left(\frac{\phi}{2}\right) - \sin\left(\frac{\phi}{2}\right) \hat{n} \\ \Rightarrow r &= sqs^* \end{aligned} \quad (4.85)$$

que é a fórmula de rotação escrita em termos de quatérnios e diretamente a partir do vetor \hat{n} do eixo de rotação e do ângulo ϕ .

Código 24. Crie uma função que implementa a rotação entre quatérnios dada por $r = sqs^*$.

`r = fc_quat_rotacao(q,s)`

Código 25. Crie uma função que implementa construa a rotação de um vetor \vec{v} ao redor de um vetor unitário \hat{n} por um ângulo ϕ .

`r = fc_quat_angulo_vetor_rotacao(v,phi,n)`

Exemplo 13. Rotacione $\vec{v} = [1, -1, 0]^T$ por $\phi = \frac{\pi}{3}$ rad ao redor de $\vec{u} = [1, 1, 1]^T$.

Defina $\hat{n} = \frac{\vec{u}}{\|\vec{u}\|} = \frac{\vec{u}}{\sqrt{3}} = \frac{[1, 1, 1]^T}{\sqrt{3}}$ e o resultado da rotação no quatérnio $r = (0, \vec{v})$.

Então $r = pq = e^{\phi \hat{n}} q = (c_{\phi} + s_{\phi} \hat{n}) q$, e o resultado da rotação será

$$\begin{aligned} \vec{v}' &= c_{\phi} \vec{v} + s_{\phi} (\hat{n} \times \vec{v}) \\ &= \cos\left(\frac{\pi}{3}\right) \cdot \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} + \frac{\sin(\pi/3)}{\sqrt{3}} \cdot \begin{bmatrix} n_2 v_3^0 - n_3 v_2 \\ n_3 v_1 - n_1 v_3^0 \\ n_1 v_2 - n_2 v_1 \end{bmatrix} \\ &= \left(\frac{1}{2}\right) \cdot \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} + \frac{\sqrt{3}}{2} \cdot \frac{1}{\sqrt{3}} \cdot \begin{bmatrix} -v_2 \\ v_1 \\ v_2 - v_1 \end{bmatrix} \\ &= \left(\frac{1}{2}\right) \cdot \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} + \left(\frac{1}{2}\right) \cdot \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = \left(\frac{1}{2}\right) \cdot \begin{bmatrix} 2 \\ 0 \\ -2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \end{aligned} \quad (4.86)$$

Exemplo 14. Rotacione $\vec{v} = [1, -1, 0]^T$ por $\phi = \frac{\pi}{3}$ rad ao redor de $\vec{u} = [1, 1, 1]^T$. Utilizando a expressão $r = sqs^*$ resolvendo em termo dos quatérnios.

$$q = \left(0, [1, -1, 0]^\top\right) \quad (4.87)$$

$$\hat{n} = \frac{\vec{u}}{\|\vec{u}\|} = \frac{\vec{u}}{\sqrt{3}} = \frac{[1, 1, 1]^\top}{\sqrt{3}} \quad (4.88)$$

$$s = \cos(\phi) + \sin(\phi)\hat{n} = \left(\sqrt{\frac{3}{2}}, \frac{1}{2} \cdot \frac{[1, 1, 1]^\top}{\sqrt{3}}\right) \quad (4.89)$$

Aplicando as funções para obter o conjugado de um quatérnio e o produto entre quatérnios tem-se

$$r = sqs^* = (0, [1, 1547; -0, 5774; -0, 5774]^\top) \cdot s^* = (0, [1, 0, -1]^\top) \Rightarrow \vec{v'} = [1, 0, -1]^\top \quad (4.90)$$

O resultado concorda com o obtido no Exemplo 13.

Exercício 60. Calcule o resultado da rotação de $[1, 2, 5]^\top$ ao redor de $[1, 1, 0]^\top$ para os seguintes ângulos
(a) $\phi = \frac{\pi}{3}$ rad, (b) $\phi = \frac{2\pi}{3}$ rad, (c) $\phi = \frac{4\pi}{3}$ rad, (d) $\phi = \frac{\pi}{2}$ rad, (e) $\phi = \frac{3\pi}{2}$ rad.

Exercício 61. Ao redor de qual vetor \hat{n} deve-se rotacionar o vetor $\vec{v} = [1, 2, 4]^\top$ por um ângulo de $\phi = \frac{\pi}{2}$ rad tal que o resultado seja $\vec{v'} = [-2, 1, 4]^\top$?

Apêndice A

Resoluções de Exercícios Propostos

A.1 Exercício 1

Solução:

Sabendo que $ij = k$, temos:

$$ijk = kk = k^2 = -1 \quad (\text{A.1})$$

Sabendo que $jk = i$, temos:

$$ijk = ii = i^2 = -1 \quad (\text{A.2})$$

Sabendo que $ki = j$, temos:

$$kij = ii = i^2 = -1 \quad (\text{A.3})$$

A.2 Exercício 2

Solução:

(a) Sabendo que $jk = i$, temos:

$$jki = ii = i^2 = -1 \quad (\text{A.4})$$

Sabendo que $ki = j$, temos:

$$jki = jj = j^2 = -1 \quad (\text{A.5})$$

(b) Sabendo que $ki = j$, temos:

$$kij = jj = j^2 = -1 \quad (\text{A.6})$$

Sabendo que $ij = k$, temos:

$$kij = kk = k^2 = -1 \quad (\text{A.7})$$

Analisando os Exercício 1 e 2, mostramos que as três expressões são equivalentes, ou seja, $ijk = jki = kij$.

A.3 Exercício 3

Solução:

Seja $q \in \mathbb{H} \mid q = q_0 + q_1e_1 + q_2e_2 + q_3e_3$. Sabendo que para multiplicação por escalar, vale a propriedade distributiva, então:

$$\lambda q = \lambda q_0 + \lambda q_1e_1 + \lambda q_2e_2 + \lambda q_3e_3 \quad (\text{A.8})$$

Vemos que $\lambda q \in \mathbb{H}$. Ou seja,

$$|\lambda q| = \sqrt{(\lambda q_0)^2 + (\lambda q_1)^2 + (\lambda q_2)^2 + (\lambda q_3)^2} \quad (\text{A.9})$$

$$= \sqrt{\lambda^2 q_0^2 + \lambda^2 q_1^2 + \lambda^2 q_2^2 + \lambda^2 q_3^2} \quad (\text{A.10})$$

$$= \sqrt{\lambda^2} \cdot \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \quad (\text{A.11})$$

$$= |\lambda| |q| \quad (\text{A.12})$$

A.4 Exercício 6

Solução:

Seja $q \in \mathbb{H} \mid q = q_0 + q_1e_1 + q_2e_2 + q_3e_3$:

$$q + \bar{q} = q_0 + q_1e_1 + q_2e_2 + q_3e_3 + q_0 - q_1e_1 - q_2e_2 - q_3e_3 \quad (\text{A.13})$$

$$= 2q_0 \quad (\text{A.14})$$

$$= 2S(q) \quad (\text{A.15})$$

A.5 Exercício 7

Solução:

Seja $q \in \mathbb{H} \mid q = q_0 + q_1e_1 + q_2e_2 + q_3e_3$:

$$q + \bar{q} = q_0 + q_1e_1 + q_2e_2 + q_3e_3 - (q_0 - q_1e_1 - q_2e_2 - q_3e_3) \quad (\text{A.16})$$

$$q + \bar{q} = q_0 + q_1e_1 + q_2e_2 + q_3e_3 - q_0 + q_1e_1 + q_2e_2 + q_3e_3 \quad (\text{A.17})$$

$$= 2(q_1e_1 + q_2e_2 + q_3e_3) \quad (\text{A.18})$$

$$= 2V(q) \quad (\text{A.19})$$

A.6 Exercício 8

Solução:

(a)

$$r = p + q - \bar{q} \quad (\text{A.20})$$

$$= p + 2V(q) \quad (\text{A.21})$$

$$= (4 + 5i - 5j + 5k) + 2(2j - 10k) \quad (\text{A.22})$$

$$= 4 + 5i - j - 15k \quad (\text{A.23})$$

(b)

$$r = 3 \cdot p + 5\bar{q} - |p| \quad (\text{A.24})$$

$$= 3(4 + 5i - 5j + 5k) + 5(2 + 2j - 10k) + \sqrt{4^2 + 5^2 + (-5)^2 + 5^2} \quad (\text{A.25})$$

$$= 12 + 15i - 15j + 15k + 10 + 10j - 50k + \sqrt{91} \quad (\text{A.26})$$

$$= 31.5394 + 15i - 5j - 35k \quad (\text{A.27})$$

(c)

$$r = S(p + q) - S(\bar{q}) \quad (\text{A.28})$$

$$= p_0 + q_0 - p_0 \quad (\text{A.29})$$

$$= 2 \quad (\text{A.30})$$

(d)

$$r = |p|^2 S(q)p + |q|^2 S(p)\bar{q} \quad (\text{A.31})$$

$$= 91 \cdot (-2)(1 + 2i + 3j - 5k) + (4 + 4 + 4 + 4)(1)(-2 - 2i + 2j + 2k) \quad (\text{A.32})$$

$$= (-182)(1 + 2i + 3j - 5k) + 16(-2 - 2i + 2j + 2k) \quad (\text{A.33})$$

$$= (-182 - 364i - 546j + 910k) + (-32 - 32i + 32j + 32k) \quad (\text{A.34})$$

$$= -214 - 396i - 514j + 878k \quad (\text{A.35})$$

A.7 Exercício 10

Solução:

Considerando $q = (a_1, \vec{v}_1)$ e $p = (a_2, \vec{v}_2)$, sabemos que:

$$qp = (a_1 a_2 - \langle \vec{v}_1, \vec{v}_2 \rangle, a_1 \vec{v}_2 + a_2 \vec{v}_1 + \vec{v}_1 \times \vec{v}_2) \quad (\text{A.36})$$

$$pq = (a_2 a_1 - \langle \vec{v}_2, \vec{v}_1 \rangle, a_2 \vec{v}_1 + a_1 \vec{v}_2 + \vec{v}_2 \times \vec{v}_1) \quad (\text{A.37})$$

Observamos que a parte escalar do resultado é igual, pois: $a_1 a_2 = a_1 a_2$ e $\langle \vec{v}_1, \vec{v}_2 \rangle = \langle \vec{v}_2, \vec{v}_1 \rangle$. Enquanto a parte vetorial só será igual em casos particulares, uma vez que, apesar de $a_1 \vec{v}_2 + a_2 \vec{v}_1 = a_2 \vec{v}_1 + a_1 \vec{v}_2$ temos que $\vec{v}_1 \times \vec{v}_2 \neq \vec{v}_2 \times \vec{v}_1$.

A.8 Exercício 11

Solução:

Para $qp \neq pq$, com $q = (a_1, \vec{v}_1) \in \mathbb{H}$ e $p = (a_2, \vec{v}_2) \in \mathbb{H}$, é necessário que $\vec{v}_1 \times \vec{v}_2 \neq \vec{v}_2 \times \vec{v}_1$, por exemplo: $\vec{v}_1 = [1 \ 0 \ 0]^\top$, $\vec{v}_2 = [1 \ 1 \ 1]^\top$, resultaria em:

$$\vec{v}_1 \times \vec{v}_2 = [0, -1, 1]^\top \quad (\text{A.38})$$

$$\vec{v}_2 \times \vec{v}_1 = [0, 1, -1]^\top \quad (\text{A.39})$$

$$\vec{v}_1 \times \vec{v}_2 \neq \vec{v}_2 \times \vec{v}_1, \forall a_1, a_2 \in \mathbb{R} \quad (\text{A.40})$$

A.9 Exercício 12

Solução:

Dado $\vec{v} = v_1 \hat{i} + v_2 \hat{j} + v_3 \hat{k}$ e $\vec{u} = u_1 \hat{i} + u_2 \hat{j} + u_3 \hat{k}$, temos que:

$$\langle \vec{v}, \vec{u} \rangle = v_1 u_1 + v_2 u_2 + v_3 u_3 \quad (\text{A.41})$$

$$= u_1 v_1 + u_2 v_2 + u_3 v_3 = \langle \vec{u}, \vec{v} \rangle \quad (\text{A.42})$$

A.10 Exercício 13

Solução:

(a)

$$\begin{aligned} \vec{v} \times \vec{u} &= (v_2 u_3 - v_3 u_2) e_1 + (v_3 u_1 - v_1 u_3) e_2 + (v_1 u_2 - v_2 u_1) e_3 \\ \vec{u} \times \vec{v} &= (u_2 v_3 - u_3 v_2) e_1 + (u_3 v_1 - u_1 v_3) e_2 + (u_1 v_2 - u_2 v_1) e_3 \\ \vec{u} \times \vec{v} &= -(v_2 u_3 - v_3 u_2) e_1 - (v_3 u_1 - v_1 u_3) e_2 - (v_1 u_2 - v_2 u_1) e_3 \\ &= -\vec{v} \times \vec{u} \end{aligned} \quad (\text{A.43})$$

(b) Significa que as partes vetoriais de cada quatérnio são colineares.

$$q = (a, \vec{v}), p = (b, \vec{u}) \in \mathbb{H} \quad (\text{A.44})$$

Os produtos de quatérnios qp e pq são

$$\begin{aligned} qp &= (ab - \langle \vec{v}, \vec{u} \rangle, a\vec{u} + b\vec{v} + \vec{v} \times \vec{u}) \\ pq &= (ba - \langle \vec{u}, \vec{v} \rangle, b\vec{v} + a\vec{u} + \vec{u} \times \vec{v}) \\ &= (ab - \langle \vec{v}, \vec{u} \rangle, a\vec{u} + b\vec{v} - \vec{v} \times \vec{u}) \end{aligned} \quad (\text{A.45})$$

de onde a condição $qp = pq$ implica na igualdade entre o produto vetorial

$$\vec{v} \times \vec{u} = -\vec{v} \times \vec{u} \Rightarrow \vec{v} \times \vec{u} = \mathbf{0} \quad (\text{A.46})$$

Dois vetores $\vec{v}, \vec{u} \in \mathbb{R}^3$ tem o produto vetorial nulo quando eles são colineares

$$\begin{aligned}\vec{u} &= c\vec{v}, \quad \forall c \in \mathbb{R} \\ \vec{v} \times \vec{u} &= \vec{v} \times (c\vec{v}) = c\vec{v} \times \vec{v} = c\mathbf{0} = \mathbf{0}\end{aligned}\tag{A.47}$$

Então se dois quatérnios comutarem, o resultado do produto sempre será

$$qp = (ab - \langle \vec{v}, \vec{u} \rangle, a\vec{u} + b\vec{v}) = pq\tag{A.48}$$

e como consequência de $\vec{u} = c\vec{v}$ tem-se

$$qp = (ab - c\langle \vec{v}, \vec{v} \rangle, ac\vec{v} + b\vec{v}) = (ab - c\|\vec{v}\|^2, (ac + b)\vec{v}) = pq\tag{A.49}$$

(c) Baseado no resultados do item (b), o caso geral de comutação é

$$q = a + xi + yj + zk, \quad p = b + c(xi + yj + zk)\tag{A.50}$$

que em valores numéricos pode ser exemplificado

$$q = 1 + 2i + 5j + 7k, \quad p = 7 + 20i + 50j + 70k \Rightarrow qp = -773 + 34i + 85j + 119k = pq\tag{A.51}$$

(d)

$$q = 1 + 2i + 5j + 7k, \quad p = b + c(2i + 5j + 7k), \quad b, c \in \mathbb{R}\tag{A.52}$$

A.11 Exercício 16

Solução: Partindo da solução do caso em que dois quatérnios comutam apresentado em (A.49) tem-se que o resultado para $q = (a, \vec{v}), p = (b, \vec{u}) = (b, c\vec{v}) \in \mathbb{H}$, como $p = \bar{q}$ o conjugado do quatérnio q , então tem-se

$$p = \bar{q} = (a, -\vec{v}) \Rightarrow b = a, \quad c = -1\tag{A.53}$$

que recai nas condições para que o produto de quatérnios comute. Portanto $\bar{q}q = q\bar{q}, \quad \forall q \in \mathbb{H}$.

A.12 Exercício 17

Solução: Sejam $q = (a, \vec{v}), \bar{q} = (a, -\vec{v}) \in \mathbb{H}$. A partir do resultado caso em que o produto comuta tem-se

$$q\bar{q} = (aa - c\|\vec{v}\|^2, (ac + b)\vec{v}) = (aa - (-1)\|\vec{v}\|^2, (a(-1) + b)\vec{v}) = (a^2 + \|\vec{v}\|^2, \mathbf{0}) = |q|^2 \in \mathbb{R}^+\tag{A.54}$$

A.13 Exercícios 32, 33, 34, 35

Solução:

Essa série de exercícios aborda o uso do comando `latex(sym)`¹ do Matlab®.

O Matlab® é capaz de gerar o código L^AT_EX diretamente de funções de variáveis simbólicas.

Os autores fornecem junto com a apostila a função `save_file.tex` e sua função de encapsulamento `fc.lib_save_file_extensao`. A função `fc.lib_save_file_extensao` prepara as entradas para `save_file.tex`. O resultado da operação é a escrita da *string* fornecida no arquivo cujo nome e formatos foram fornecidos à função.

Essa técnica visa agilizar deduções mais elaboradas que sejam muito grandes para se manipular manualmente, ou transcrever resultados obtidos no Matlab® diretamente para o L^AT_EX (MEZA; ZUCATELLI, 2016; ZUCATELLI; MEZA; FENILI, 2017).

O programa fornecido contém um trecho para substituição de caracteres para permitir rápida adaptação entre o nome da variável usada no código do Matlab® e o nome da variável que ela representará na notação que for usada em L^AT_EX. Exemplo é usar “th” para a variável θ no Matlab® porém precisar substituir pelos caracteres `\theta`, que é o comando usado em L^AT_EX.

¹help latex

$$R_{xzy}(\phi, \psi, \theta) = \begin{bmatrix} \cos(\psi) \cos(\theta) & -\sin(\psi) & \cos(\psi) \sin(\theta) \\ \sin(\phi) \sin(\theta) + \cos(\phi) \cos(\theta) \sin(\psi) & \cos(\phi) \cos(\psi) & \cos(\phi) \sin(\psi) \sin(\theta) - \cos(\theta) \sin(\phi) \\ \cos(\theta) \sin(\phi) \sin(\psi) - \cos(\phi) \sin(\theta) & \cos(\psi) \sin(\phi) & \cos(\phi) \cos(\theta) + \sin(\phi) \sin(\psi) \sin(\theta) \end{bmatrix} \quad (\text{A.55})$$

$$R_{yxz}(\psi, \phi, \theta) = \begin{bmatrix} \cos(\psi) \cos(\theta) + \sin(\phi) \sin(\psi) \sin(\theta) & \cos(\psi) \sin(\phi) \sin(\theta) - \cos(\theta) \sin(\psi) & \cos(\phi) \sin(\theta) \\ \cos(\phi) \sin(\psi) & \cos(\phi) \cos(\psi) & -\sin(\phi) \\ \cos(\theta) \sin(\phi) \sin(\psi) - \cos(\psi) \sin(\theta) & \sin(\psi) \sin(\theta) + \cos(\psi) \cos(\theta) \sin(\phi) & \cos(\phi) \cos(\theta) \end{bmatrix} \quad (\text{A.56})$$

$$R_{zyx}(\psi, \phi, \theta) = \begin{bmatrix} \cos(\psi) \cos(\theta) & \cos(\psi) \sin(\phi) \sin(\theta) - \cos(\phi) \sin(\psi) & \sin(\phi) \sin(\psi) + \cos(\phi) \cos(\psi) \sin(\theta) \\ \cos(\theta) \sin(\psi) & \cos(\phi) \cos(\psi) + \sin(\phi) \sin(\psi) \sin(\theta) & \cos(\phi) \sin(\psi) \sin(\theta) - \cos(\psi) \sin(\phi) \\ -\sin(\theta) & \cos(\theta) \sin(\phi) & \cos(\phi) \cos(\theta) \end{bmatrix} \quad (\text{A.57})$$

$$R_{zyz}(\psi, \phi, \theta) = \begin{bmatrix} \cos(\phi) \cos(\psi) \cos(\theta) - \sin(\phi) \sin(\psi) & -\cos(\phi) \sin(\psi) - \cos(\psi) \cos(\theta) \sin(\phi) & \cos(\psi) \sin(\theta) \\ \cos(\psi) \sin(\phi) + \cos(\phi) \cos(\theta) \sin(\psi) & \cos(\phi) \cos(\psi) - \cos(\theta) \sin(\phi) \sin(\psi) & \sin(\psi) \sin(\theta) \\ -\cos(\phi) \sin(\theta) & \sin(\phi) \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (\text{A.58})$$

A.13.1 Código de resolução

Esse código utiliza o código que substitui expressões e o código que gera arquivos a partir de uma *string*. Juntos, eles facilitam consideravelmente o trabalho manual de transcrever expressões do Matlab® para o relatório em L^AT_EX.

Algoritmo A.1: Resolução dos Exercícios 34, 33, 34 e 35 com uso de simbólico no Matlab® e integração com L^AT_EX

```

1 syms th psi phi real;
2 %% Exemplo 03 - 04 - XYZ
3 R = fc_lib_Rx(phi)*fc_lib_Ry(th)*fc_lib_Rz(psi);
4 R_str = fc_int_old_to_new(R);
5 fc_lib_save_file_extensao('input_Ex_matriz_R_XYZ', '.tex', R_str);
6 %% Ex26 - Exemplo 05 - XZY
7 R = fc_lib_Rx(phi)*fc_lib_Rz(psi)*fc_lib_Ry(th);
8 R_str = fc_int_old_to_new(R);
9 fc_lib_save_file_extensao('input_Ex_matriz_R_XZY', '.tex', R_str);
10 %% Ex27 - Exemplo 06 - YXZ
11 R = fc_lib_Ry(th)*fc_lib_Rx(phi)*fc_lib_Rz(psi);
12 R_str = fc_int_old_to_new(R);
13 fc_lib_save_file_extensao('input_Ex_matriz_R_YXZ', '.tex', R_str);
14 %% Ex28 - Exemplo 07 - ZYX
15 R = fc_lib_Rz(th)*fc_lib_Ry(phi)*fc_lib_Rx(psi);
16 R = fc_lib_Rz(psi)*fc_lib_Ry(th)*fc_lib_Rx(phi);
17 R_str = fc_int_old_to_new(R);
18 fc_lib_save_file_extensao('input_Ex_matriz_R_ZYX', '.tex', R_str);
19 %% Ex29 - Exemplo 08 - ZYZ
20 R = fc_lib_Rz(psi)*fc_lib_Ry(th)*fc_lib_Rz(phi);
21 R_str = fc_int_old_to_new(R);
22 fc_lib_save_file_extensao('input_Ex_matriz_R_ZYZ', '.tex', R_str);
23 %% função interna para correção de nomenclatura para LaTeX
24 function R_str = fc_int_old_to_new(R)
25 R_str = latex(R);
26 old = {'\mathrm{th}','\mathrm{psi}','\varphi','\left(\begin{array}', '\\end{array}\right)'};
27 new = {'theta','psi','phi','\left[\begin{array}', '\\end{array}\right]'};
28 R_str = regexprep(R_str, old, new);
29 end

```

A.14 Exercício 40

Solução:

$\exists \vec{p}$, tal que $\vec{p} = R\vec{p}$?

Problema de autovalor:

$$R\vec{p} = \lambda\vec{p}, \quad R \in M(3, \mathbb{R}), \lambda \in \mathbb{R} \quad (\text{A.59})$$

$$\Rightarrow \vec{p}(1 - \lambda) = \mathbf{0} \quad (\text{A.60})$$

$$\vec{p} \neq \mathbf{0} \Rightarrow \lambda = 1 \quad (\text{A.61})$$

Sim, existe, são todos cujas matrizes R tem autovalor $\lambda = 1$.

A.15 Exercício 41

Solução: Seja o vetor \vec{w} o resultado da rotação de \vec{v} pela matriz $R \in \mathbb{R}^{n \times n}$ dada por

$$\vec{w} = R\vec{v} \quad (\text{A.62})$$

Convencionalmente, após a rotação tem-se $\|\vec{w}\|_2 = \|\vec{v}\|_2$, então a matriz R tem a relação

$$\|R\vec{v}\|_2 = 1, \quad \forall \vec{v} \quad (\text{A.63})$$

Por simplicidade, analisemos o autovetor \vec{x} de autovalor λ de R

$$\|R\vec{x}\|_2 = \|\lambda\vec{x}\|_2 = |\lambda|\|\vec{x}\|_2 = \|\vec{x}\|_2 \Rightarrow |\lambda| = 1, \forall \lambda \quad (\text{A.64})$$

Consequentemente, o $\det(R)$ tem seu módulo fixado em 1

$$|\det(R)| = |\lambda_1||\lambda_2| \dots |\lambda_n| = 1 \quad (\text{A.65})$$

De onde se nota que se $|\det R| > 1$ então $\|\vec{w}\|_2 > 1$. E se $|\det R| < 1$ então $\|\vec{w}\|_2 < 1$.

A.16 Exercício 44

Solução:

Mostrar $\langle \vec{v}, \vec{u} \rangle \vec{u} = \vec{u} \vec{u}^\top \vec{v}$.

$$\langle \vec{v}, \vec{u} \rangle \cdot \vec{u} = \underbrace{(v_1 u_1 + v_2 u_2 + v_3 u_3)}_C \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} C, \quad C \in \mathbb{R} \quad (\text{A.66})$$

Por outro lado

$$\vec{u} \cdot \vec{u}^\top \cdot \vec{v} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \underbrace{(v_1 u_1 + v_2 u_2 + v_3 u_3)}_C = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} C \quad (\text{A.67})$$

de onde se concluir que

$$\langle \vec{v}, \vec{u} \rangle \vec{u} = \vec{u} \vec{u}^\top \vec{v} \quad (\text{A.68})$$

Vale destacar que o produto interno é comutativo

$$\langle \vec{v}, \vec{u} \rangle = \langle \vec{u}, \vec{v} \rangle = \vec{u}^\top \cdot \vec{v} = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (\text{A.69})$$

E que a ordem das operações matriciais influencia no resultado. O produto interno retorna **sempre** um escalar porém, a inversão da ordem da operação retorna uma matriz.

$$\langle \vec{u}, \vec{u} \rangle = \vec{u}^\top \cdot \vec{u} = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = u_1^2 + u_2^2 + u_3^2 \in \mathbb{R} \quad (\text{A.70})$$

$$\vec{u} \cdot \vec{u}^\top = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} = \begin{bmatrix} u_1 u_1 & u_1 u_2 & u_1 u_3 \\ u_2 u_1 & u_2 u_2 & u_2 u_3 \\ u_3 u_1 & u_3 u_2 & u_3 u_3 \end{bmatrix} \in \mathbb{R}^{3 \times 3} \quad (\text{A.71})$$

A.17 Exercício 50

Solução:

Para este exercício, optamos por uma resolução que exemplifica o uso do pacote simbólico do Matlab[®] conforme apresenta o Algoritmo A.2.

Algoritmo A.2: Código de resolução com Matlab[®] simbólico do Exercício 50.

```

1 syms x real;
2 R = [0.9330 -0.3536 0.0670;
3      0.3536 x -0.3536;
4      0.0670 0.3536 0.9330];
5 A = R - R.'
6 c = (trace(R)-1)/2
7 phi = acos(c)
8 %2*sin(phi)*nX = A
9 nX = A/(2*sin(phi))
10 y = nX(1,2)^2 + nX(1,3)^2 + nX(2,3)^2
11 S = solve(y == 1, x)
12 S_double = double(S)
13 x_double = S_double(S_double > 0) %S_double(2)
14 phi_double = double(acos((R(1,1) + x_double + R(3,3) - 1)/2))
15 % alternativamente
16 % phi_double = double(subs(phi,x,x_double))
17 %% vetor normal
18 n = zeros(3,1);
19 n(3) = double(subs(-nX(1,2),x,x_double));
20 n(2) = double(subs(+nX(1,3),x,x_double));
21 n(1) = double(subs(-nX(2,3),x,x_double));
22 disp(n);

```

A.18 Exercício 56

Solução:

Aplicando a relação de Euler para quaternions da Eq. (4.57) tem-se que

$$\begin{aligned}
 e^{i+j} &= \cos(\|i+j\|) + \frac{\text{sen}(\|i+j\|)}{\|i+j\|}(i+j) \\
 &= \cos(\sqrt{2}) + \frac{\text{sen}(\sqrt{2})}{\sqrt{2}}(i+j) \approx 0,156 + 0,698(i+j) \\
 e^i \cdot e^j &= \left(\cos(1) + \text{sen}(1)i \right) \left(\cos(1) + \text{sen}(1)j \right) \\
 &= \cos^2(1) + \cos(1)\text{sen}(1)j + \cos(1)\text{sen}(1)i + \text{sen}^2(1)ij \\
 &= \cos^2(1) + \cos(1)\text{sen}(1)(i+j) + \text{sen}^2(1)k
 \end{aligned} \tag{A.72}$$

e claramente as duas expressões são diferentes.

A expressão e^{i+j} significa **uma** rotação ao redor do vetor $i+j$ enquanto que $e^i \cdot e^j$ significa **duas** rotações sucessivas, a **primeira** ao redor de i e a **segunda** ao redor de j . Então faz todo o sentido que os resultados de cada uma das expressões sejam diferentes entre si.

A.19 Exercício 58

Solução:

$$e^{\phi P} q = q e^{-\phi P} \tag{A.73}$$

$$(\cos(\phi), \text{sen}(\phi)\hat{n})(0, \vec{v}) = (0, \vec{v})(\cos(\phi), \text{sen}(\phi)\hat{n}) \tag{A.74}$$

$$(0 - \text{sen}(\phi) \langle \hat{n}, \vec{v} \rangle, \cos(\phi)\vec{v} + \text{sen}(\phi)\hat{n} \times \vec{v}) = (0 - \text{sen}(\phi) \langle \vec{v}, \hat{n} \rangle, \cos(\phi)\vec{v} - \text{sen}(\phi)\vec{v} \times \hat{n}) \tag{A.75}$$

$$\hat{n} \perp \vec{v} \Rightarrow \langle \hat{n}, \vec{v} \rangle = \langle \vec{v}, \hat{n} \rangle = 0; \quad \vec{v} \times \hat{n} = -\hat{n} \times \vec{v} \text{ antissimétrico} \tag{A.76}$$

$$\Rightarrow (0, \cos(\phi)\vec{v} + \text{sen}(\phi)\hat{n} \times \vec{v}) = (0, \cos(\phi)\vec{v} + \text{sen}(\phi)\hat{n} \times \vec{v}) \tag{A.77}$$

A.20 Exercício 59

Solução:

$$e^{\theta p} q = q e^{\theta p} \quad (\text{A.78})$$

$$(\cos(\theta), \sin(\theta)\hat{n})(0, \vec{u}) = (0, \vec{u})(\cos(\theta), \sin(\theta)\hat{n}) \quad (\text{A.79})$$

$$(-\sin(\theta) \langle \hat{n}, \vec{u} \rangle, \cos(\theta)\vec{u} + \sin(\theta)\hat{n} \times \vec{u}) = (-\sin(\theta) \langle \hat{n}, \vec{u} \rangle, \cos(\theta)\vec{u} - \sin(\theta)\vec{u} \times \hat{n}) \quad (\text{A.80})$$

$$\langle \hat{n}, \vec{v} \rangle = \langle \vec{v}, \hat{n} \rangle; \quad \vec{u} \parallel \hat{n} \Rightarrow \vec{u} \times \hat{n} = \hat{n} \times \vec{u} = \mathbf{0} \quad (\text{A.81})$$

$$\Rightarrow (-\sin(\theta) \langle \hat{n}, \vec{u} \rangle, \cos(\theta)\vec{u}) = (-\sin(\theta) \langle \hat{n}, \vec{u} \rangle, \cos(\theta)\vec{u}) \quad (\text{A.82})$$

Apêndice B

Normas

Este material ainda é um rascunho e apanhado de ideias gerais.

Auxílio sobre normas¹ e módulos.

Uma norma $\|\cdot\|$ é uma operação que leva um elemento do espaço vetorial \mathbb{V} no corpo dos reais não-negativos \mathbb{R}^+

$$\|\cdot\| : \mathbb{V} \rightarrow \mathbb{R}^+ \quad (\text{B.1})$$

com as seguintes propriedades. Seja $u, v \in \mathbb{V}$ e $a \in \mathbb{R}$

$$\begin{aligned} \|v\| &= 0 \Leftrightarrow v = \mathbf{0} \\ \|av\| &= |a| \cdot \|v\| \\ \|u + v\| &\leq \|u\| + \|v\|, \quad (\text{Desigualdade Triangular}) \end{aligned} \quad (\text{B.2})$$

a desigualdade triangular é uma consequência das dimensões de qualquer triângulo na geometria euclidiana. Como forma de partida para se lembrar da propriedade, tem que em qualquer triângulo retângulo cujos lados são dados pelos vetores u e v , tal que a hipotenusa é a soma desses vetores $h = u + v$. Assim tem-se que a hipotenusa h é sempre menor ou igual que a soma dos catetos u e v .

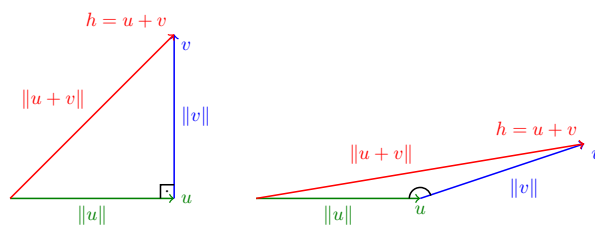


Figura B.1: Desigualdade triangular.

$$\|v + u\| \leq \|v\| + \|u\| \quad (\text{B.3})$$

As principais normas utilizadas para $\mathbb{V} = \mathbb{R}^n$ são as normas 1, 2, e ∞ , norma p a seguir.

$$\|v\|_1 = \sum_{i=1}^n |v_i| \quad (\text{B.4})$$

$$\|v\|_2 = \left(\sum_{i=1}^n |v_i|^2 \right)^{\frac{1}{2}} \quad (\text{B.5})$$

$$\|v\|_p = \left(\sum_{i=1}^n |v_i|^p \right)^{\frac{1}{p}}, \quad 1 \leq p < \infty \quad (\text{B.6})$$

$$\|v\|_\infty = \max_{i=1}^n (|v_i|) \quad (\text{B.7})$$

¹Normas em espaços de dimensão finita e infinita.

Relação entre as normas

$$\|v\|_\infty \leq \|v\|_2 \leq \|v\|_1 \leq \sqrt{n}\|v\|_2 \leq n\|v\|_\infty \quad (\text{B.8})$$

A Figura B.2 ilustra os casos com $\|\cdot\| = 1$ para as normas 1, 2 e ∞ .

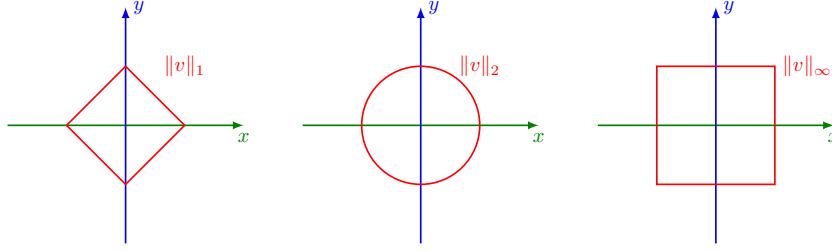


Figura B.2: Ilustração dos casos com $\|\cdot\|_p = 1$ para as normas $p = \{1, 2, \infty\}$.

Fonte: (WIKIPÉDIA, 2018e).

Para duas dimensões é possível desenhar figuras geométricas que representam as normas. Esse conceito é expansível para dimensões maiores.

A norma 1 é composta por quatro retas formando um losango:

$$\begin{aligned} \|x + y\|_1 &= |x| + |y| = 1 \\ x > 0, y > 0 : x + y &= 1 \Rightarrow y = 1 - x \\ x > 0, y < 0 : x - y &= 1 \Rightarrow y = x - 1 \\ x < 0, y > 0 : -x + y &= 1 \Rightarrow y = 1 + x \\ x < 0, y < 0 : -x - y &= 1 \Rightarrow y = -1 - x \end{aligned} \quad (\text{B.9})$$

A norma 2 descreve a equação de uma circunferência de raio $R = 1$:

$$\|x + y\|_2 = \sqrt{x^2 + y^2} = 1 \Rightarrow x^2 + y^2 = 1 \quad (\text{B.10})$$

A norma infinita descreve um quadrado de lado $L = 2$:

$$\begin{aligned} \|x + y\|_\infty &= \max\{|x|, |y|\} = 1 \\ x > 0, y > 0 : \max\{|x|, |y|\} &= 1 \Rightarrow x = 1 \text{ ou } y = 1 \\ x > 0, y < 0 : \max\{|x|, |y|\} &= 1 \Rightarrow x = 1 \text{ ou } y = -1 \\ x < 0, y > 0 : \max\{|x|, |y|\} &= 1 \Rightarrow x = -1 \text{ ou } y = 1 \\ x < 0, y < 0 : \max\{|x|, |y|\} &= 1 \Rightarrow x = -1 \text{ ou } y = -1 \end{aligned} \quad (\text{B.11})$$

A Figura B.3 apresenta 3 vetores \vec{v} , \vec{u} , \vec{w} e serve de base para de exemplo para se entender as aplicações das principais normas.

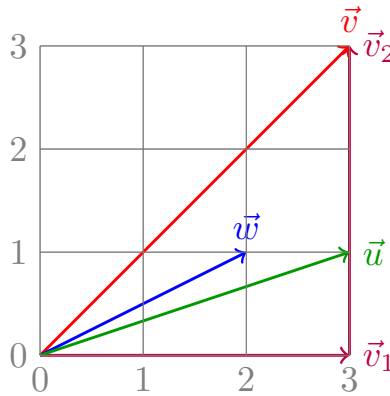


Figura B.3: Exemplo de normas.

Exemplo do uso das principais normas:

$$\begin{aligned}\|\vec{v}\|_1 &= |\vec{v}_1| + |\vec{v}_2| = 3 + 3 = 6 \\ \|\vec{v}\|_2 &= \sqrt{|\vec{v}_1|^2 + |\vec{v}_2|^2} = \sqrt{9 + 9} = \sqrt{18} \approx 4,2426\end{aligned}\tag{B.12}$$

$$\begin{aligned}\|\vec{v}\|_\infty &= \max\{|\vec{v}_1|, |\vec{v}_2|\} = \max\{3, 3\} = 3 \\ \|\vec{u}\|_1 &= |3| + |1| = 3 + 1 = 4 \\ \|\vec{u}\|_2 &= \sqrt{3^2 + 1^2} = \sqrt{9 + 1} = \sqrt{10} \approx 3,1622\end{aligned}\tag{B.13}$$

$$\begin{aligned}\|\vec{u}\|_\infty &= \max\{3, 1\} = 3 \\ \|\vec{w}\|_1 &= |2| + |1| = 2 + 1 = 3 \\ \|\vec{w}\|_2 &= \sqrt{2^2 + 1^2} = \sqrt{4 + 1} = \sqrt{5} \approx 2,2361 \\ \|\vec{w}\|_\infty &= \max\{2, 1\} = 2\end{aligned}\tag{B.14}$$

Uma generalização de terminologia para escalares, vetores e matrizes é usar a palavra *tensor*. A Tabela B.1 apresenta a nomenclatura normalmente empregada na classificação de tensores.

Tabela B.1: Nomes dos elementos tensoriais

rank	tipo de elemento	gráfico
0	escalar	\cdot
1	(co)vetor	$(-)$
2	matriz	\square
$n \geq 3$	tensor rank n	\square

Para definirmos a noção de norma de uma função, primeiro, entenderemos o conceito de produto interno para vetores, expandiremos esse conceito para funções e finalmente aplicaremos esse conceito para definir norma de funções contínuas.

Definição 2 (Produto Interno): O **produto interno** é uma operação que leva dois vetores de um espaço vetorial \mathbb{V} em um número real não-negativo, i.e., \mathbb{R}^+ , com a propriedade de resultar no escalar zero se e somente ao menos um dos vetores envolvidos for o vetor nulo.

$$\begin{aligned}\langle \cdot, \cdot \rangle : \mathbb{V} \times \mathbb{V} &\rightarrow \mathbb{R}^+ \\ \langle v, u \rangle &= 0 \Leftrightarrow v = \mathbf{0} \text{ ou } u = \mathbf{0}\end{aligned}\tag{B.15}$$

A matriz simétrica $M = M^\top$ representa uma forma bilinear do produto interno em termos vetoriais.

$$\langle v, u \rangle = v^\top M u = v^\top M u^\top = u^\top v^\top M^\top = u^\top M^\top (v^\top)^\top = u^\top M v = \langle u, v \rangle\tag{B.16}$$

Para o caso que $\mathbb{V} = \mathbb{R}^n$ e $M = I_n$.

$$\langle v, u \rangle = v^\top u = \sum_{i=1}^n v_i u_i = \sum_{i=1}^n u_i v_i = u^\top v = \langle u, v \rangle\tag{B.17}$$

Para o caso que $\mathbb{V} = \mathbb{C}^n$, altera-se o transposto pelo transposto conjugado.

$$\langle v, u \rangle = v^H u = \sum_{i=1}^n \bar{v}_i u_i = \sum_{i=1}^n \bar{u}_i v_i = u^H v = \langle u, v \rangle\tag{B.18}$$

Uma matriz A é dita *hermitiana* caso ela seja igual a sua transposta conjugada.

$$A = A^H = \overline{(A^\top)} = (\bar{A})^\top\tag{B.19}$$

É possível criar normas a partir de produtos interno. Essas normas são conhecidas como **norma induzida** por um produto interno em um espaço vetorial. Para o caso que cada entrada no operador produto interno ser o mesmo elemento, tem-se o seguinte resultado.

$$\|v\|^2 = \langle v, v \rangle = 0 \Leftrightarrow v = \mathbf{0}\tag{B.20}$$

A justificativa do uso do transporte conjugado $(\cdot)^\dagger$ ou $(\cdot)^H$ é que essa operação preserva a noção de norma para vetores baseados no corpo dos complexos.

$$z_1^\top \cdot z_1 = \begin{bmatrix} a, & -bi \end{bmatrix} \cdot \begin{bmatrix} a \\ bi \end{bmatrix} = a^2 - b^2 \neq |z_1|^2 \quad (\text{B.21})$$

$$z_1^H \cdot z_1 = \begin{bmatrix} a, & -bi \end{bmatrix} \cdot \begin{bmatrix} a \\ bi \end{bmatrix} = a^2 + b^2 = |z_1|^2 \in \mathbb{R}^* \quad (\text{B.22})$$

Para funções, usa-se normas de espaços de dimensão infinita ([WIKIPÉDIA, 2018b](#)).²

Para $f, g : \mathbb{R} \rightarrow \mathbb{R}$

$$\langle f, g \rangle = \int_{-\infty}^{\infty} f(x) \cdot g(x) dx \quad (\text{B.23})$$

Para $f, g : \mathbb{C} \rightarrow \mathbb{C}$

$$\langle f, g \rangle = \int_{-\infty}^{\infty} \overline{f(x)} \cdot g(x) dx \quad (\text{B.24})$$

onde $\overline{f(x)}$ é o complexo conjugado da função $f(x)$.

A partir da definição de produto interno entre f e g , pode-se calcular a norma de $f(x)$:

$$\|f(x)\|_2^2 = \langle f, f \rangle = \int_{-\infty}^{\infty} \overline{f(x)} \cdot f(x) dx = \int_{-\infty}^{\infty} |f(x)|^2 dx \quad (\text{B.25})$$

a norma-1 é obtida como a integral do valor absoluto da função f :

$$\|f(x)\|_1 = \int_{-\infty}^{\infty} |f(x)| dx \quad (\text{B.26})$$

e a norma infinita o supremo da função para todo x :

$$\|f(x)\|_\infty = \sup_x |f(x)| \quad (\text{B.27})$$

A definição apresentada na Equação (B.24) é a definição do espaço $L^2(D)$ ([WIKIPÉDIA, 2018d](#)), o qual é um espaço de Hilbert ([WIKIPÉDIA, 2018c](#)) dotado do produto interno apresentado.

Repare que para funções vetoriais é necessário acrescentar a transposição.

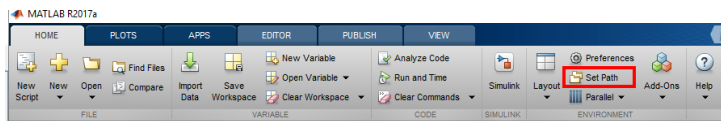
²As citações à Wikipédia devem ser entendidas como fontes primárias de consulta e introdução aos tópicos mencionados.

Apêndice C

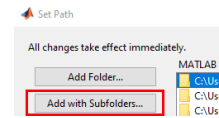
Criando sua própria biblioteca em Matlab®

A seguir apresentamos como criar sua própria biblioteca particular em Matlab® ao adicionar pastas aos caminhos reconhecidos pelo programa.

A opção padrão do Matlab® é executar os códigos que encontram-se na pasta atual do programa em uso, além daqueles que constam na própria biblioteca do programa. Para criar uma biblioteca pessoal, o usuário deve adicionar as pastas e **TODAS** as subpastas que contenham os códigos que deseja usar como biblioteca. A adição dos caminhos das pastas é feito pela opção **Add Path** (Figura C.1(a)) que se encontra no menu **HOME**. Atenta-se para o cuidado de se inserir as subpastas conforme apresenta a Figura C.1(b).



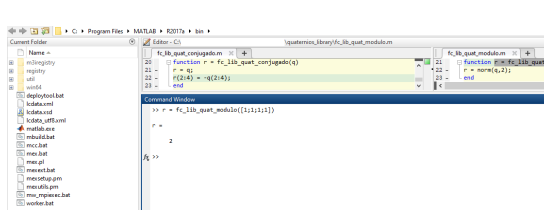
(a) Entrada para opção **Add Path**.



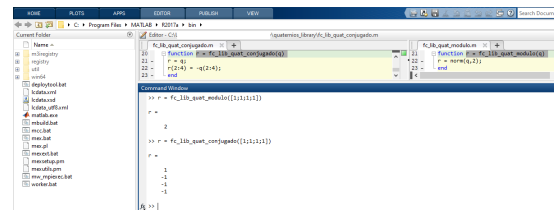
(b) Adição de pastas e subpastas.

Figura C.1: Adicionando pastas e subpastas para criar biblioteca do usuário.

Os exemplos de execução de códigos a partir da biblioteca, enquanto a pasta selecionada é outra, são apresentados na Figura C.2.



(a) Execução do módulo de um quaternário.



(b) Execução do conjugado de um quaternário.

Figura C.2: Chamando funções a partir da biblioteca.

Lembre-se que ao utilizar uma biblioteca é importante manter os códigos organizados, com nomes claros e bem documentados.

Apêndice D

Códigos em Matlab[®] para Álgebra de Quatérnios

Os códigos que serão inseridos aqui não terão os comentários iniciais que usamos para as informações do *help* e para exemplos de uso.

O prefixo `fc_lib` é referente às funções que pertencem à biblioteca (*library*) de funções dos autores.

Essa biblioteca faz uso do isomorfismo entre \mathbb{H} e \mathbb{R}^4 explicado no [Capítulo 3](#).

Lembrando: $q = q_0 + q_1i + q_2j + q_3k \in \mathbb{H}$ é descrito por $q = [q_0, q_1, q_2, q_3]^\top \in \mathbb{R}^4$.

D.1 Códigos com álgebra de quatérnios

Algoritmo D.1: Rotação de um vetor por um ângulo com quatérnios.

```
1 function r = fc_lib.quat_angulo_vetor_rotacao(v,th,n)
2 q = [0;v];
3 p = [cos(th/2);sin(th/2)*n];
4 r = fc_lib.quat_rotacao(q,p);
5 r = r(2:4); % somente a parte imaginária-vetorial.
6 end
```

Algoritmo D.2: Conjugado de quatérnio.

```
1 function r = fc_lib.quat_conjugado(q)
2 r = q;
3 r(2:4) = -q(2:4);
4 end
```

Algoritmo D.3: Divisão de quatérnio à direita.

```
1 function r = fc_lib.quat_divisao_dir(q,p)
2 r = fc_lib.quat_produto(q,fc_lib.quat_inverso(p));
3 end
```

Algoritmo D.4: Divisão de quatérnio à esquerda.

```
1 function r = fc_lib.quat_divisao_esq(p,q)
2 r = fc_lib.quat_produto(fc_lib.quat_inverso(p),q);
3 end
```

Algoritmo D.5: Inverso de quatérnio.

```
1 function r = fc_lib.quat_inverso(q)
2 r = fc_lib.quat_conjugado(q)/( dot(q,q) );
3 end
```


Algoritmo D.6: Módulo de quatérnio.

```

1 function r = fc_lib_quat_modulo(q)
2 r = norm(q,2);
3 end

```

Algoritmo D.7: Exponencial de quatérnio.

```

1 function r = fc_lib_quat_exp(q)
2 v = q(2:4); th = norm(v,2); n = v/th;
3 r = exp(q(1))*[cos(th); sin(th)*n];
4 end

```

Algoritmo D.8: Plot coordenadas 3D com quatérnios.

```

1 function fc_lib_quat_plot_coordenadas_3D(x,y,z,TH,N,flags,fh)
2 flag_plot_xyz = flags(1);
3 flag_drawnow = flags(2);
4 flag_sphere = flags(3);
5 flag_normal = flags(4);
6 figure(fh);
7 Qsi = eye(3); % base inicial
8
9 if flag_plot_xyz == 1
10     plot3(x,y,z,'--k');
11 end
12 xlabel('X','FontSize',16,'FontWeight','bold','Color','r');
13 ylabel('Y','FontSize',16,'FontWeight','bold','Color','g');
14 zlabel('Z','FontSize',16,'FontWeight','bold','Color','b');
15 hold all; grid on; AZ = 135; EL = 30; view([AZ,EL]);
16
17 if flag_sphere == 1
18     [XS,YS,ZS] = sphere(10);
19     XS = XS/2; YS = YS/2; ZS = ZS/2;
20     mesh(XS,YS,ZS); alpha(0.6);
21 end
22
23 for k = 1:length(x)
24     th = TH(k); n = N(k,:);
25     xyz = [x(k);y(k);z(k)];
26
27     % rotação com construção de quaternios
28     Qsi(:,1) = fc_lib_quat_angulo_vetor_rotacao(Qsi(:,1),th,n);
29     Qsi(:,2) = fc_lib_quat_angulo_vetor_rotacao(Qsi(:,2),th,n);
30     Qsi(:,3) = fc_lib_quat_angulo_vetor_rotacao(Qsi(:,3),th,n);
31
32     % setas e plot
33     seta1 = [xyz, xyz+Qsi(:,1)];
34     seta2 = [xyz, xyz+Qsi(:,2)];
35     seta3 = [xyz, xyz+Qsi(:,3)];
36     plot3(seta1(1,:),seta1(2,:),seta1(3,:), 'r');
37     plot3(seta2(1,:),seta2(2,:),seta2(3,:), 'g');
38     plot3(seta3(1,:),seta3(2,:),seta3(3,:), 'b');
39     if flag_normal == 1
40         setan = [xyz, xyz+n];
41         plot3(setan(1,:),setan(2,:),setan(3,:), 'k');
42     end
43     % formas de apresentação
44     if flag_drawnow == 1
45         drawnow;
46     elseif flag_drawnow == 2
47         % temporizador maroto
48         tic, while toc < 0.5, end
49         drawnow;
50     elseif flag_drawnow == 3
51         % temporizador manual
52         input('Tecle algo para continuar: \n');
53         drawnow;
54     end
55 end

```

```

56 axis equal; hold off;
57 end

```

Algoritmo D.9: Produto de quatérnios.

```

1 function r = fc_lib_quat_produto(q,p)
2 r = zeros(4,1);
3 r(1) = q(1)*p(1) - dot(q(2:4),p(2:4));
4 r(2:4) = q(1)*p(2:4) + p(1)*q(2:4) + cross(q(2:4),p(2:4));
5 end

```

Algoritmo D.10: Rotação de quatérnios.

```

1 function r = fc_lib_quat_rotacao(q,p)
2 r = fc_lib_quat_produto(p,q);
3 r = fc_lib_quat_produto(r,fc_lib_quat_conjugado(p));
4 end

```

D.2 Códigos com *plot* de coordenadas

Algoritmo D.11: Plot de coordenadas 2D.

```

1 function fc_lib_plot_coordenadas_2D(x,y,r,th,flags,fh,alfa)
2 flag_plot_xy = flags(1);
3 flag_drawnow = flags(2);
4 figure(fh);
5 %% Trajetória
6 if flag_plot_xy == 1
7     plot(x,y,'--k');
8 end
9 %% Cores dos eixos e ângulo de câmera
10 xlabel('X','FontSize',16,'FontWeight','bold','Color','r');
11 ylabel('Y','FontSize',16,'FontWeight','bold','Color','g');
12 hold all; grid on;
13 str1 = 'r'; str2 = 'g';
14 %% plot dos sistemas de coordenadas conforme dados de entrada
15 for k = 1:length(x)
16     L = r(k);
17     S = L/3;
18     % cálculo do corpo da seta e de sua pontal
19     setalx = [x(k), x(k)+L*cos(th(k))];
20     setaly = [y(k), y(k)+L*sin(th(k))];
21     pontalx = [setalx(2)+S*cos(th(k) - alfa), setalx(2), setalx(2)+S*cos(th(k) + alfa)];
22     pontaly = [setaly(2)+S*sin(th(k) - alfa), setaly(2), setaly(2)+S*sin(th(k) + alfa)];
23     % cálculo do corpo da seta e de sua pontal
24     seta2x = [x(k), x(k)+L*cos(th(k) + pi/2)];
25     seta2y = [y(k), y(k)+L*sin(th(k) + pi/2)];
26     ponta2x = [seta2x(2)+S*cos(th(k) + pi/2 - alfa), seta2x(2), ...
27         seta2x(2)+S*cos(th(k) + pi/2 + alfa)];
28     ponta2y = [seta2y(2)+S*sin(th(k) + pi/2 - alfa), seta2y(2), ...
29         seta2y(2)+S*sin(th(k) + pi/2 + alfa)];
30     % plot
31     plot(setalx,setaly, str1);
32     plot(pontalx,pontaly, str1);
33     % plot
34     plot(seta2x,seta2y, str2);
35     plot(ponta2x,ponta2y, str2);
36     if flag_drawnow == 1
37         drawnow;
38     end
39 end
40 axis equal; hold off;
41 end

```

Algoritmo D.12: Plot de coordenadas 3D.

```

1 function fc_lib_plot_coordenadas_3D(x,y,z,TH,PHI,PSI,flags,o,fh)

```

```

2  flag_plot_xyz = flags(1); flag_drawnow = flags(2); flag_sphere = flags(3);
3  figure(fh);
4  Base = eye(3); % base inicial
5  dkg = [0 0.5 0]; % Dark Green
6  %% Trajetória
7  if flag_plot_xyz == 1
8      plot3(x,y,z, '--k');
9  end
10 %% Cores dos eixos e ângulo de câmera
11 fh = fc_lib_XYZ_RGB_axis(fh,16);
12 hold all; grid on; AZ = 135; EL = 30; view([AZ,EL]);
13 %% Esfera de referência
14 if flag_sphere == 1
15     [XS,YS,ZS] = sphere(10);
16     XS = XS/2; YS = YS/2; ZS = ZS/2;
17     mesh(XS,YS,ZS); alpha(0.6);
18 end
19 %% plot dos sistemas de coordenadas conforme dados de entrada
20 for k = 1:length(x)
21     th = TH(k); phi = PHI(k); psi = PSI(k);
22     xyz = [x(k);y(k);z(k)];
23     switch o
24         case 'xyz'
25             R = fc_lib_R_xyz(th,phi,psi);
26         case 'xzy'
27             R = fc_lib_R_xzy(th,psi,phi);
28         case 'yxz'
29             R = fc_lib_R_yxz(phi,th,psi);
30         case 'yzx'
31             R = fc_lib_R_yzx(phi,psi,th);
32         case 'zyx'
33             R = fc_lib_R_zyx(phi,psi,th);
34         case 'zxy'
35             R = fc_lib_R_zxy(psi,th,phi);
36         case 'zxz'
37             R = fc_lib_R_zxz(psi,th,phi);
38         case 'zyz'
39             R = fc_lib_R_zyz(psi,phi,th);
40         otherwise
41             disp('Error');
42             return;
43     end
44     Body = R*Base;
45     % setas e plot
46     seta1 = [xyz, xyz+Body(:,1)];
47     seta2 = [xyz, xyz+Body(:,2)];
48     seta3 = [xyz, xyz+Body(:,3)];
49     plot3(seta1(1,:),seta1(2,:),seta1(3,:), 'Color', 'r');
50     plot3(seta2(1,:),seta2(2,:),seta2(3,:), 'Color', dkg);
51     plot3(seta3(1,:),seta3(2,:),seta3(3,:), 'Color', 'b');
52     % formas de apresentação
53     if flag_drawnow == 1
54         drawnow;
55     elseif flag_drawnow == 2 % temporizador automático
56         tic, while toc < 0.5, end
57         drawnow;
58     elseif flag_drawnow == 3 % temporizador manual
59         input('Tecle algo para continuar: \n');
60         drawnow;
61     end
62 end
63 axis equal; hold off;
64 end

```

Algoritmo D.13: Plot setas 2D.

```

1  function fh = fc_lib_plot_setas_2D(x,y,r,th,flags,fh,alfa)
2  % alfa = -5*pi/6; % alfa = -11*pi/12; % alfa = -23*pi/24;
3  flag_plot_xy = flags(1);
4  flag_drawnow = flags(2);

```

```

5 figure(fh);
6 %% Trajetória
7 if flag_plot_xy == 1
8     plot(x,y,'--k');
9 end
10 %% Cores dos eixos
11 xlabel('X','FontSize',16,'FontWeight','bold','Color','r');
12 ylabel('Y','FontSize',16,'FontWeight','bold','Color','g');
13 hold all; grid on; str = 'b';
14 %% plot dos sistemas de coordenadas conforme dados de entrada
15 for k = 1:length(x)
16     L = r(k);
17     S = L/3;
18     % cálculo do corpo da seta e de sua ponta
19     setax = [x(k), x(k)+L*cos(th(k))];
20     setay = [y(k), y(k)+L*sin(th(k))];
21     pontax = [setax(2)+S*cos(th(k) - alfa), setax(2), setax(2)+S*cos(th(k) + alfa)];
22     pontay = [setay(2)+S*sin(th(k) - alfa), setay(2), setay(2)+S*sin(th(k) + alfa)];
23     % plot
24     plot(setax,setay, str);
25     plot(pontax,pontay, str);
26     if flag_drawnow == 1
27         drawnow;
28     end
29 end
30 axis equal; hold off;
31 end

```

Algoritmo D.14: Plot setas 3D.

```

1 function fc_lib_plot_setas_3D(x,y,z,r,th,phi,flags,fh,alfa)
2 flag_plot_xyz = flags(1);
3 flag_drawnow = flags(2);
4 figure(fh);
5 %% Trajetória
6 if flag_plot_xyz == 1
7     plot3(x,y,z,'--k');
8 end
9 %% Cores dos eixos
10 fc_lib_XYZ_RGB.axis(fh,16);
11 hold all; grid on; str = 'b';
12 %% plot dos sistemas de coordenadas conforme dados de entrada
13 for k = 1:length(x)
14     L = r(k);
15     S = L/10;
16     % cálculo do corpo da seta e de sua ponta
17     setax = [x(k), x(k)+L*cos(th(k))*sin(phi(k))];
18     setay = [y(k), y(k)+L*sin(th(k))*sin(phi(k))];
19     setaz = [z(k), z(k)+L*cos(phi(k))];
20     pontax = [setax(2)+S*cos(th(k))*sin(phi(k) - alfa), setax(2), ...
21         setax(2)+S*cos(th(k))*sin(phi(k) + alfa)];
22     pontay = [setay(2)+S*sin(th(k))*sin(phi(k) - alfa), setay(2), ...
23         setay(2)+S*sin(th(k))*sin(phi(k) - alfa)];
24     pontaz = [setaz(2)+S*cos(phi(k) - alfa), setaz(2), ...
25         setaz(2)+S*cos(phi(k) + alfa)];
26     % plot
27     plot3(setax,setay,setaz, str);
28     plot3(pontax,pontay,pontaz, str);
29     if flag_drawnow == 1
30         drawnow;
31     end
32 end
33 axis equal; hold off;
34 end

```

Algoritmo D.15: Plot de vetor em 3D com identificação.

```

1 function fh = fc_lib_plot_vetor_3D(fh,r,str,T,rt,...
2     alfa,beta,th,phi,S,cor,linha,flag_tex)
3 plot3(r(1,:),r(2,:),r(3,:), 'Color',cor(1,:), 'LineStyle',linha);

```

```

4 fc_lib_plot_ponta_setas_3D(fh,r(1,end),r(2,end),r(3,end), ...
5     alfa,beta,th,phi,S,cor(1,:),linha);
6 if ~strcmp(str, '')
7     if flag_tex == 1
8         text(rt(1), rt(2), rt(3),str,'Interpreter','latex','Color',cor(2,:), 'FontSize',T);
9     else
10        text(rt(1), rt(2), rt(3),str,'Color',cor(2,:), 'FontSize',T);
11    end
12 end
13 end

```

D.3 Códigos com matrizes de rotação

Algoritmo D.16: Rotação Fórmula de Rodrigues.

```

1 function R = fc_lib_Rot_Rodrigues(u,phi)
2 uX = [0,      -u(3),  u(2);
3       u(3),   0,      -u(1);
4       -u(2),  u(1),   0];
5 R = cos(phi)*eye(3) + (1-cos(phi))*u*u' + sin(phi)*uX;
6 end

```

Algoritmo D.17: Matriz de rotação sobre eixo **X**.

```

1 function Rx = fc_lib_Rx(A)
2 Rx = [1  0  0;
3       0 cos(A) -sin(A);
4       0 sin(A) +cos(A)];
5 end

```

Algoritmo D.18: Matriz de rotação sobre eixo **Y**.

```

1 function Ry = fc_lib_Ry(A)
2 Ry = [+cos(A)  0 +sin(A) ;
3       0        1  0;
4       -sin(A)  0 +cos(A)];
5 end

```

Algoritmo D.19: Matriz de rotação sobre eixo **Z**.

```

1 function Rz = fc_lib_Rz(A)
2 Rz = [cos(A) -sin(A) 0;
3       sin(A) +cos(A) 0;
4       0        0    1];
5 end

```

Algoritmo D.20: Matriz de rotação na sequência **XYZ**.

```

1 function R = fc_lib_R_xyz(th,phi,psi)
2 R = [...
3     cos(phi)*cos(psi), -cos(phi)*sin(psi), sin(phi);
4     cos(th)*sin(psi) + cos(psi)*sin(phi)*sin(th), ...
5     cos(psi)*cos(th) - sin(phi)*sin(psi)*sin(th), -cos(phi)*sin(th);
6     sin(psi)*sin(th) - cos(psi)*cos(th)*sin(phi), ...
7     cos(psi)*sin(th) + cos(th)*sin(phi)*sin(psi), cos(phi)*cos(th)];
8 end

```

Algoritmo D.21: Matriz de rotação na sequência **XZY**.

```

1 function R = fc_lib_R_xzy(th,psi,phi)
2 R = [...
3     cos(phi)*cos(psi),      -sin(psi),      cos(psi)*sin(phi);
4     sin(phi)*sin(th) + cos(phi)*cos(th)*sin(psi), cos(psi)*cos(th), ...
5     cos(th)*sin(phi)*sin(psi) - cos(phi)*sin(th);
6     cos(phi)*sin(psi)*sin(th) - cos(th)*sin(phi), cos(psi)*sin(th), ...

```

```

7      cos(phi)*cos(th) + sin(phi)*sin(psi)*sin(th)];
8  end

```

Algoritmo D.22: Matriz de rotação na sequência **YXZ**.

```

1  function R = fc_lib_R_yxz(phi,th,psi)
2  R = [...
3      cos(phi)*cos(psi) + sin(phi)*sin(psi)*sin(th), ...
4      cos(psi)*sin(phi)*sin(th) - cos(phi)*sin(psi), cos(th)*sin(phi);
5      cos(th)*sin(psi), cos(psi)*cos(th), -sin(th);
6      cos(phi)*sin(psi)*sin(th) - cos(psi)*sin(phi), ...
7      sin(phi)*sin(psi) + cos(phi)*cos(psi)*sin(th), cos(phi)*cos(th)];
8  end

```

Algoritmo D.23: Matriz de rotação na sequência **YZX**.

```

1  function R = fc_lib_R_zyx(phi,psi,th)
2  R = [...
3      cos(phi)*cos(psi), sin(phi)*sin(th) - cos(phi)*cos(th)*sin(psi), ...
4      cos(th)*sin(phi) + cos(phi)*sin(psi)*sin(th);
5      sin(psi), cos(psi)*cos(th), -cos(psi)*sin(th);
6      -cos(psi)*sin(phi), cos(phi)*sin(th) + cos(th)*sin(phi)*sin(psi), ...
7      cos(phi)*cos(th) - sin(phi)*sin(psi)*sin(th)];
8  end

```

Algoritmo D.24: Matriz de rotação na sequência **ZXY**.

```

1  function R = fc_lib_R_zxy(psi,th,phi)
2  R = [...
3      cos(phi)*cos(psi) - sin(phi)*sin(psi)*sin(th), -cos(th)*sin(psi), ...
4      cos(psi)*sin(phi) + cos(phi)*sin(psi)*sin(th);
5      cos(phi)*sin(psi) + cos(psi)*sin(phi)*sin(th), cos(psi)*cos(th), ...
6      sin(phi)*sin(psi) - cos(phi)*cos(psi)*sin(th);
7      -cos(th)*sin(phi), sin(th), cos(phi)*cos(th)];
8  end

```

Algoritmo D.25: Matriz de rotação na sequência **ZYX**.

```

1  function R = fc_lib_R_zyx(phi,psi,th)
2  R = [...
3      cos(phi)*cos(psi), cos(psi)*sin(phi)*sin(th) - cos(th)*sin(psi), ...
4      sin(psi)*sin(th) + cos(psi)*cos(th)*sin(phi);
5      cos(phi)*sin(psi), cos(psi)*cos(th) + sin(phi)*sin(psi)*sin(th), ...
6      cos(th)*sin(phi)*sin(psi) - cos(psi)*sin(th);
7      -sin(phi), cos(phi)*sin(th), ...
8      cos(phi)*cos(th)];
9  end

```

Algoritmo D.26: Matriz de rotação na sequência **ZXZ**.

```

1  function R = fc_lib_R_zxz(psi,th,phi)
2  R = [...
3      cos(phi)*cos(psi) - cos(th)*sin(phi)*sin(psi), ...
4      -cos(psi)*sin(phi) - cos(phi)*cos(th)*sin(psi), sin(psi)*sin(th);
5      cos(phi)*sin(psi) + cos(psi)*cos(th)*sin(phi), ...
6      cos(phi)*cos(psi)*cos(th) - sin(phi)*sin(psi), -cos(psi)*sin(th);
7      sin(phi)*sin(th), cos(phi)*sin(th), ...
8      cos(th)];
9  end

```

Algoritmo D.27: Matriz de rotação na sequência **ZYZ**.

```

1  function R = fc_lib_R_zyz(psi,phi,th)
2  R = [...
3      cos(phi)*cos(psi)*cos(th) - sin(psi)*sin(th), ...
4      -cos(th)*sin(psi) - cos(phi)*cos(psi)*sin(th), cos(psi)*sin(phi);

```

```

5      cos(psi)*sin(th) + cos(phi)*cos(th)*sin(psi), ...
6      cos(psi)*cos(th) - cos(phi)*sin(psi)*sin(th), sin(phi)*sin(psi);
7      -cos(th)*sin(phi), sin(phi)*sin(th), ...
8      cos(phi)];
9  end

```

D.4 Códigos auxiliares

Algoritmo D.28: Encapsulamento da função que gera arquivos em L^AT_EX a partir de uma *string*.

```

1  function fc_lib_save_file_extensao(file_name, str_ext, string_to_file)
2  fid = fopen([file_name,str_ext],'w');
3  save_file_tex(fid, string_to_file);
4  fclose(fid);
5  end

```

Algoritmo D.29: Nomenclatura e padronização dos rótulos dos eixos XYZ

```

1  function fh = fc_lib_XYZ_RGB_axis(fh,T)
2  figure(fh); dkg = [0 0.5 0]; %Dark Green
3  xlabel('X','FontSize',T,'FontWeight','bold','Color','r');
4  ylabel('Y','FontSize',T,'FontWeight','bold','Color',dkg);
5  zlabel('Z','FontSize',T,'FontWeight','bold','Color','b');
6  end

```

D.5 Códigos geradores de Figuras

Código gerador da Figura 1.1.

Algoritmo D.30: Figura do exemplo de coordenadas 2D em Matlab®.

```

1  clear; close all; clc;
2  %% figura
3  fh = figure;
4  fh = fc_lib_XYZ_RGB_axis(fh,16);
5  grid on; hold all; axis([0,3,0,3]);
6  %% pontos
7  x0 = 0.5; y0 = 0.5; x1 = 2; y1 = 2;
8  %% vetor que liga (x0,y0) a (x1,y1)
9  %quiver(x0, y0, x1-x0 +0.15, y1-y0 +0.15,'Color','b'); %0.15 cobre a redução da seta
10 plot([x0, x1],[y0, y1],'b');
11 fc_lib_plot_ponta_setas_2D(fh,x1,y1,-11*pi/12,+1*pi/4,0.15,'b','--');
12 str = '$\vec{v}$';
13 text(x1/2+0.2,y1/2+0.4,str,'Interpreter','latex','FontSize',16,'Color','b');
14 % -- pontos de construção
15 plot([x0,x1],[y0,y1],'ro','MarkerSize',8);
16 str = '$(x_0,y_0)$';
17 text(x0 -0.25,y0 -0.25,str,'Interpreter','latex','FontSize',16);
18 str = '$(x_1,y_1)$';
19 text(x1,y1 +0.1,str,'Interpreter','latex','FontSize',16);
20 %str = '$R^2$';
21 str = '$\mathrm{R}^2$';
22 text(x0,y1,str,'Interpreter','latex','FontSize',16);
23 %text(x0,y1,str,'FontSize',16);
24 %% linhas de medida
25 plot([x0-0.2,x1-0.2],[y0+0.2, y1+0.2],'k');
26 str = '$\rho$';
27 text(0.9,1.6,str,'Interpreter','latex','FontSize',16,'Color',[0.8 0 0.5]);
28 % --- linha de chamada
29 plot([x0,x0-0.25],[y0,y0+0.25],'k');
30 plot([x1,x1-0.25],[y1,y1+0.25],'k');
31 plot([x0-0.25, x1+0.5],[y0,y0],'--k');
32 fc_lib_plot_ponta_setas_2D(fh,x0-0.2,y0+0.2,-11*pi/12,-3*pi/4,0.15,'k','--');
33 fc_lib_plot_ponta_setas_2D(fh,x1-0.2,y1+0.2,-11*pi/12,+1*pi/4,0.15,'k','--');
34 %% circunferência em z=0
35 t = 0:0.05:pi/4;
36 x = 1*cos(t)+x0; y = 1*sin(t)+y0;

```

```

37 cor = [0 0.5 0]; linha = '-';
38 plot(x,y,'Color',cor,'LineStyle',linha);
39 str = '$\theta$';
40 text(1.5,1,str,'Interpreter','latex','FontSize',16,'Color',cor);
41 % --- seta
42 xe = x(end); ye = y(end); S = 0.15;
43 alfa = -5*pi/6; th = degtorad(135);
44 fc_lib_plot_ponta_setas_2D(fh,xe,ye,alfa,th,S,cor,linha);
45 %% perspectiva e salvar
46 axis equal;
47 saveas(gcf,'fig.coordenadas_2D','eps2');

```

Geração de setas em 2D.

Algoritmo D.31: Inserção da ponta da seta em um ponto qualquer em 2D.

```

1 function fh = fc_lib_plot_ponta_setas_2D(fh,xe,ye,alfa,th,S,cor,linha)
2 figure(fh);
3 P1 = [xe+S*cos(th - alfa), xe, xe+S*cos(th + alfa)], ...
4      [ye+S*sin(th - alfa), ye, ye+S*sin(th + alfa)];
5 plot(P1(:,1),P1(:,2),'Color',cor,'LineStyle',linha);
6 end

```

Código gerador da Figura 1.2.

Algoritmo D.32: Figura do exemplo de coordenadas 3D em Matlab®.

```

1 clear; close all; clc;
2 %% figura
3 fh = figure;
4 fh = fc_lib_XYZ_RGB_axis(fh,16);
5 grid on; hold all; axis([0,3,0,3,0,3]);
6 AZ = 110; EL = 30; view([AZ,EL]);
7 %% pontos
8 x0 = 0; y0 = 0; z0 = 0; x1 = 2; y1 = 2; z1 = 2;
9 str = '$\mathrm{R}^3$';
10 text(x0,2*y1/4,3*z1/4,str,'Interpreter','latex','FontSize',16);
11 %% vetor que liga (x0,y0) a (x1,y1)
12 plot3([x0, x1],[y0, y1],[z0, z1],'b'); phi = -atan2(2*sqrt(2)) - pi/2;
13 fc_lib_plot_ponta_setas_3D(fh,x1,y1,z1,-1*pi/8,-0*pi/8,-7*pi/4,phi,0.25,'b','-');
14 fc_lib_plot_ponta_setas_3D(fh,x1,y1,z1,-0*pi/8,-1*pi/16,-7*pi/4,phi,0.4,'b','-');
15 str = '$\vec{v}$';
16 text(13*x1/16,13*y1/16,11*z1/16,str,'Interpreter','latex','FontSize',16,'Color','b');
17 %% -- pontos de construção
18 plot3([x0,x1],[y0,y1],[z0,z1],'ro','MarkerSize',8);
19 str = '$(x_0,y_0,z_0)$';
20 text(x0,y0-1,z0,str,'Interpreter','latex','FontSize',16);
21 str = '$(x_1,y_1,z_1)$';
22 text(x1,y1+0.1,z1+0.1,str,'Interpreter','latex','FontSize',16);
23 %% medidas sobre os eixos
24 str = '$x_1$';
25 text(x1,-0.2,0,str,'Interpreter','latex','FontSize',16,'Color','r');
26 str = '$y_1$';
27 text(-0.2,y1,0,str,'Interpreter','latex','FontSize',16,'Color',[0 0.5 0]);
28 str = '$z_1$';
29 text(-0.2,0,z1,str,'Interpreter','latex','FontSize',16,'Color','b');
30 %% projeções
31 plot3([x1,x1],[y1,y1],[0,z1],':m');
32 plot3([x0,x1],[y0,y1],[0,0],':m');
33 plot3([0,x1],[0,y1],[z1,z1],':m');
34 %% tracejados
35 plot3([x0,x0],[y0,y0],[0,z1],'--k');
36 plot3([x0,x1],[y0,y0],[0,0],'--k');
37 plot3([x0,x0],[y0,y1],[0,0],'--k');
38 %% nos eixos
39 cor = [0.2 0 1]; % roxo
40 plot3([x1,x1],[0,y1],[0,0],':', 'Color', cor);
41 plot3([0,x1],[y1,y1],[0,0],':', 'Color', cor);
42 %% linhas de medida
43 str = '$\rho$';
44 text(x1/2,y1/2,5*z1/8,str,'Interpreter','latex','FontSize',16,'Color',[0.8 0 0.5]);

```



```

45 %% circunferência em z=0 para theta
46 cor = [0 0.5 0]; linha = '-'; % verde escuro
47 t = 0:0.05:(pi/4+0.02); u = pi/2*ones(size(t)); L = 1;
48 x = x0 + L*cos(t).*sin(u); y = y0 + L*sin(t).*sin(u); z = zeros(size(t));
49 plot3(x,y,z,'Color',cor,'LineStyle',linha);
50 str = '$\theta$';
51 text(1.3,0.4,0,str,'Interpreter','latex','FontSize',16, 'Color',cor);
52 %% --- seta para theta
53 xe = x(end); ye = y(end); ze = 0; S = 0.15;
54 alfa = -5*pi/6; beta = 0;
55 th = degtorad(135); phi = degtorad(90);
56 fc_lib_plot_ponta_setas_3D(fh,xe,ye,ze,alfa,beta,th,phi,S,cor,linha)
57 %% circunferência ao longo de v para phi
58 cor = [0.9 0.5 0]; %laranja
59 phie = acos(z1/sqrt(x1^2+y1^2+z1^2));
60 u = 0:0.05:phie; t = pi/4*ones(size(u)); L = 1;
61 x = x0 + L*cos(t).*sin(u); y = y0 + L*sin(t).*sin(u); z = z0 + L*cos(u);
62 plot3(x,y,z,'Color',cor,'LineStyle',linha);
63 str = '$\varphi$';
64 text(x1/5,y1/5,4*z1/8,str,'Interpreter','latex','FontSize',16, 'Color',cor);
65 %% --- seta para phi
66 xe = x(end); ye = y(end); ze = z(end); S = 0.15;
67 alfa = 0; beta = -5*pi/6; th = degtorad(45); phi = degtorad(140);
68 fc_lib_plot_ponta_setas_3D(fh,xe,ye,ze,alfa,beta,th,phi,S,cor,linha)
69 %% perspectiva e salvar
70 axis([0,3,0,3,0,3]); %axis equal;
71 saveas(gcf,'../Figuras/fig-coordenadas_3D','eps2');

```

Geração de setas em 3D.

Algoritmo D.33: Inserção da ponta da seta em um ponto qualquer em 3D.

```

1 % linha = '-'; % 'LineStyle'
2 % fc_lib_plot_ponta_setas_3D(fh,xe,ye,ze,alfa,beta,th,phi,S,cor,linha)
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 function fh = fc_lib_plot_ponta_setas_3D(fh,xe,ye,ze,alfa,beta,th,phi,S,cor,linha,Tlin)
5 figure(fh);
6 P1 = [ [xe+S*cos(th - alfa)*sin(phi - beta), xe, xe+S*cos(th + alfa)*sin(phi + beta)], ...
7       [ye+S*sin(th - alfa)*sin(phi - beta), ye, ye+S*sin(th + alfa)*sin(phi + beta)], ...
8       [ze+S*cos(phi - beta), ze, ze+S*cos(phi + beta)]];
9 plot3(P1(:,1),P1(:,2),P1(:,3),'Color',cor,'LineStyle',linha,'LineWidth',Tlin);
10 end

```

Códigos geradores da Figura 3.1.

Algoritmo D.34: Figura do isomorfismo $\mathbb{R}^2 \cong \mathbb{C}$.

```

1 clear; close all; clc;
2 %% figura
3 fh = figure;
4 fh = fc_lib_XYZ_RGB_axis(fh,16);
5 grid on; hold all; axis([-1,3,-1,3]);
6 %% pontos
7 x0 = 0; y0 = 0; x1 = 2; y1 = 2;
8 %% vetor que liga (x0,y0) a (x1,y1)
9 plot([x0, x1],[y0, y1],'b');
10 fc_lib_plot_ponta_setas_2D(fh,x1,y1,-11*pi/12,+1*pi/4,0.15,'b','-');
11 str = '$\vec{v}$';
12 text(x1/2,y1/2+0.4,str,'Interpreter','latex','FontSize',16,'Color','b');
13 % -- pontos de construção
14 plot([x0,x1],[y0,y1],'ro','MarkerSize',8);
15 str = '$\mathrm{R}^2 \cong \mathrm{C}$';
16 text(x1/2,5*y1/4,str,'Interpreter','latex','FontSize',16);
17 %% projeções
18 plot([x0,y1],[y1,y1],':m');
19 plot([x1,x1],[x1,0],':m');
20 %% eixos
21 xx = x1+1; yy = y1+1; S = 0.2; dd = 1;
22 plot([x0-dd, xx],[y0,y0],'-k');
23 plot([x0, x0],[y0-dd,yy],'-k');
24 % --- setas

```

```

25 alfa = -5*pi/6; th = degtorad(0);
26 fc_lib_plot_ponta_setar_2D(fh,xx,y0,alfa,th,S,'k','-');
27 th = degtorad(90);
28 fc_lib_plot_ponta_setar_2D(fh,x0,yy,alfa,th,S,'k','-');
29 % --- grandezas dos eixos
30 str = '$\hat{i}$ , $\mathrm{R}$';
31 text(xx-0.5,y0-0.3,str,'Interpreter','latex','FontSize',16);
32 str = '$\hat{j}$ , $\mathrm{R}$';
33 text(x0-0.6,yy-0.25,str,'Interpreter','latex','FontSize',16);
34 %% medidas sobre os eixos
35 str = '$v_x$';
36 text(x1,-0.25,str,'Interpreter','latex','FontSize',16, 'Color', 'r');
37 str = '$v_y$';
38 text(-0.35,y1,str,'Interpreter','latex','FontSize',16, 'Color', [0 0.5 0]);
39 %% perspectiva e salvar
40 axis equal;
41 saveas(gcf,'fig_isomorfismo_R2_C1','eps2');

```

Algoritmo D.35: Figura do isomorfismo $\mathbb{R}^4 \cong \mathbb{C}^2$.

```

1 clear; close all; clc;
2 %% figura
3 fh = figure;
4 fh = fc_lib_XYZ_RGB_axis(fh,16);
5 grid on; hold all; axis([-1,3,-1,3]);
6 %% pontos
7 x0 = 0; y0 = 0; x1 = 2; y1 = 2;
8 %% vetor que liga (x0,y0) a (x1,y1)
9 plot([x0, x1],[y0, y1],'b');
10 fc_lib_plot_ponta_setar_2D(fh,x1,y1,-11*pi/12,+1*pi/4,0.15,'b','-');
11 str = '$q$';
12 text(x1/2,y1/2+0.4,str,'Interpreter','latex','FontSize',16,'Color','b');
13 % -- pontos de construção
14 plot([x0,x1],[y0,y1],'ro','MarkerSize',8);
15 str = '$\mathrm{R}^4 \cong \mathrm{C}^2$';
16 text(x1/2,5*y1/4,str,'Interpreter','latex','FontSize',16);
17 %% projeções
18 plot([x0,y1],[y1,y1],':m');
19 plot([x1,x1],[x1,0],':m');
20 %% eixos
21 xx = x1+1; yy = y1+1; S = 0.2; dd = 1;
22 plot([x0-dd, xx],[y0,y0],'-k');
23 plot([x0, x0],[y0-dd,yy],'-k');
24 % --- setas
25 alfa = -5*pi/6; th = degtorad(0);
26 fc_lib_plot_ponta_setar_2D(fh,xx,y0,alfa,th,S,'k','-');
27 th = degtorad(90);
28 fc_lib_plot_ponta_setar_2D(fh,x0,yy,alfa,th,S,'k','-');
29 % --- grandezas dos eixos
30 str = '$\pi_1, \mathrm{C}$';
31 text(xx-0.5,y0-0.3,str,'Interpreter','latex','FontSize',16);
32 str = '$\pi_2, \mathrm{C}$';
33 text(x0-0.7,yy-0.25,str,'Interpreter','latex','FontSize',16);
34 %% medidas sobre os eixos
35 str = '$\alpha$';
36 text(x1,-0.25,str,'Interpreter','latex','FontSize',16, 'Color', 'r');
37 str = '$\beta$';
38 text(-0.35,y1,str,'Interpreter','latex','FontSize',16, 'Color', [0 0.5 0]);
39 %% perspectiva e salvar
40 axis equal;
41 saveas(gcf,'fig_isomorfismo_R4_C2','eps2');

```

Código gerador da [Figura 4.1](#).

Algoritmo D.36: Figura do exemplo de rotação 2D em Matlab®.

```

1 clear; close all; clc;
2 %% figura
3 fh = figure;
4 fh = fc_lib_XYZ_RGB_axis(fh,16);

```

```

5 grid on; hold all; axis([-4,4,-1,4]); alfa = -5*pi/6;
6 %% pontos
7 x0 = 0; y0 = 0; x1 = 2; y1 = 2; v = [x1; y1];
8 %% vetor que liga (x0,y0) a (x1,y1)
9 plot([x0, x1],[y0, y1],'b');
10 fc_lib_plot_ponta_setas_2D(fh,x1,y1,alfa,pi/4,0.2,'b','-');
11 str = '$z_1 = (x_1,y_1)$';
12 text(x1+0.25,y1+0.25,str,'Interpreter','latex','FontSize',16,'Color','b',...
13      'Rotation',45);
14 str = '$\vec{v}$';
15 text(x1-1,y1-0.6, str,'Interpreter','latex','FontSize',16,'Color','b');
16 % -- pontos de construção
17 plot([x0,x1],[y0,y1],'ro','MarkerSize',8);
18 str = '$\mathrm{C}$';
19 text(x1/2,3*y1/2,str,'Interpreter','latex','FontSize',16);
20 %% unidade imaginária
21 plot([0 0], [0 1], 'g');
22 fc_lib_plot_ponta_setas_2D(fh,0,1,alfa,pi/2,0.2,'g','-');
23 str = '\theta_2'; % = 90°
24 text(0.2,0.9, str,'FontSize',16,'Color','g');
25 str = '$z_2 = (x_2,y_2)$';
26 text(0.5,-0.8, str,'Interpreter','latex','FontSize',16,'Color','g');
27 plot([0.5 0.5 0],[0 0.5 0.5],'g');
28 fc_lib_plot_ponta_setas_2D(fh,0,0.5,alfa,pi,0.1,'g','-');
29 %% projeções
30 plot([x0,y1],[y1,y1],':m','LineWidth',1.5);
31 plot([x1,x1],[y1,0],':m','LineWidth',1.5);
32 %% eixos
33 dd = 2; xx = x1+dd; yy = y1+dd; S = 0.3;
34 plot([x0-dd-2, xx],[y0,y0],'-k');
35 plot([x0, x0],[y0-dd,yy],'-k');
36 % --- setas
37 th = 0; fc_lib_plot_ponta_setas_2D(fh,xx,y0,alfa,th,S,'k','-');
38 th = degtorad(90); fc_lib_plot_ponta_setas_2D(fh,x0,yy,alfa,th,S,'k','-');
39 % --- grandezas dos eixos
40 str = '$\hat{i}$ , $\mathrm{R}$';
41 text(xx-0.8,y0-0.4,str,'Interpreter','latex','FontSize',16);
42 str = '$\hat{j}$ , $\mathrm{R}$';
43 text(x0-1,yy-0.25,str,'Interpreter','latex','FontSize',16);
44 %% medidas sobre os eixos
45 str = '$v_x$';
46 text(x1,-0.25,str,'Interpreter','latex','FontSize',16, 'Color', 'r');
47 str = '$v_y$';
48 text(0.15,y1+0.25,str,'Interpreter','latex','FontSize',16, 'Color', [0 0.5 0]);
49 %% ângulo theta em z=0
50 t = 0:0.05:pi/4; R = sqrt(2); x = R*cos(t); y = R*sin(t);
51 cor = [0.8 0.5 0.2];
52 plot(x,y,'Color',cor,'LineStyle','-');
53 str = '\theta_1'; % = 45°
54 text(1.5,0.5,str,'FontSize',14, 'Color',cor);
55 % --- seta
56 xe = x(end); ye = y(end); S = 0.15; th = 3*pi/4;
57 fc_lib_plot_ponta_setas_2D(fh,xe,ye,alfa,th,S,cor,'-');
58 %% meia circunferência em z=0
59 t = 0:0.05:(pi+0.05); R = norm(v,2); x = R*cos(t); y = R*sin(t);
60 plot(x,y,'Color','r','LineStyle','--');
61 %% rotação 90 graus
62 th = pi/2; n = [0;0;1]; n = n/norm(n,2); cor = 'c';
63 r = fc_lib_quat_angulo_vetor_rotacao([v;0],th,n);
64 plot([x0, r(1)],[y0, r(2)],cor);
65 % --- setas
66 th = 3*pi/4; fc_lib_plot_ponta_setas_2D(fh,r(1),r(2),alfa,th,0.2,cor,'-');
67 str = '$z_3 = (x_3,y_3)$';
68 text(r(1)-1,r(2)-0.5,str,'Interpreter','latex','FontSize',16, 'Color',cor,...
69      'Rotation',45);
70 str = '$\vec{v}$';
71 text(r(1)+0.8,r(2)-0.6, str,'Interpreter','latex','FontSize',16,'Color',cor);
72 %% projeções
73 plot([x0,r(1)],[r(2),r(2)],':m','LineWidth',1.5);
74 plot([r(1),r(1)],[r(2),0],':m','LineWidth',1.5);
75 %% medidas sobre os eixos

```

```

76 str = '$v_x'$';
77 text(r(1),-0.25,str,'Interpreter','latex','FontSize',16, 'Color', 'r');
78 str = '$v_y'$';
79 text(-0.5,r(2)+0.25,str,'Interpreter','latex','FontSize',16, 'Color', [0 0.5 0]);
80 %% perspectiva e salvar
81 axis equal;
82 saveas(gcf,'fig.rotacao-2D','eps2');

```

Códigos da Figura 4.11 sobre o Teorema de Euler.

Algoritmo D.37: Figura sobre o Teorema de Euler.

```

1 clear; close all; clc;
2 %%
3 th = pi/2;
4 Ry = fc_lib_Ry(th);
5 r = [0; 0; 0];
6 I = eye(3);
7 E = 1;
8 T = 16;
9 strnulo = '';
10 trac = '--';
11 passo = 0.05;
12 AZ = 130; EL = 30; %view([AZ,EL]);
13 dkg = [0, 0.5, 0]; %Dark Green;
14 %dko = [0.9 0.4 0]; %Dark Orange 01
15 dko = [0.8 0.3 0]; %Dark Orange 02
16 vio = [0.5 0 0.6]; %Violet
17 %% Ry
18 f1 = figure(1); grid on; hold all;
19 f1 = fc_lib_body_orientacao(f1, r, I, E, strnulo);
20 %% p
21 x = 0.5; y = 1; z = 0.8;
22 r1 = [0 x; 0 y; 0 z];
23 str = '$p$';
24 rt = [x/2, y/2, 10*z/16];
25 cor = dko; S = 0.1; linha = '-'; flagtex = 1;
26 th = atan2(y,x); phi = acos(z/sqrt(x^2+y^2+z^2)); alfa = 0; beta = -11*pi/12;
27 f1 = fc_lib_plot_vetor_3D(f1,r1,str,14,rt,...
28     alfa,beta,th,phi,S,[cor; cor],linha,1,flagtex);
29 % --- tracejados
30 plot3([0,x],[0,y],[0,0],'Color',cor,'LineStyle',trac);
31 plot3([x,x],[y,y],[0,z],'Color',cor,'LineStyle',trac);
32 plot3([0,x],[0,y],[z,z],'Color',cor,'LineStyle',trac);
33 %% q
34 x = 1; y = 0.5; z = 0.8;
35 r1 = [0 x; 0 y; 0 z];
36 str = '$q$';
37 rt = [x/2 + 0.3, y/2 + 0.05, 14*z/16];
38 cor = vio; S = 0.1; linha = '-'; flagtex = 1;
39 th = atan2(y,x); phi = acos(z/sqrt(x^2+y^2+z^2)); alfa = 0; beta = -11*pi/12;
40 f1 = fc_lib_plot_vetor_3D(f1,r1,str,14,rt,...
41     alfa,beta,th,phi,S,[cor; cor],linha,1,flagtex);
42 % --- tracejados
43 plot3([0,x],[0,y],[0,0],'Color',cor,'LineStyle',trac);
44 plot3([x,x],[y,y],[0,z],'Color',cor,'LineStyle',trac);
45 plot3([0,x],[0,y],[z,z],'Color',cor,'LineStyle',trac);
46 %% Nomes coloridos e ponto de vista
47 f1 = fc_lib_XYZ_RGB.axis(f1,T);
48 view([AZ, EL]); axis equal;
49 %% --- textos
50 text(-0.01,0,0.05,'$o$', 'FontSize',T, 'Color','k','Interpreter','latex');
51 text(0.85*E,0.1,0,'e_1', 'FontSize',T, 'Color','r');
52 text(0.1,0.85*E,0,'e_2', 'FontSize',T, 'Color',dkg,'HorizontalAlignment','right');
53 text(0.25,0,0.9*E,'e_3', 'FontSize',T, 'Color','b');
54 %% --- salvar
55 saveas(f1,'../Figuras/fig.teoremaEuler_p_q','eps2');

```

Código gerador da Figura 4.12.

Algoritmo D.38: Figura do exemplo de rotação 3D em Matlab®.

```

1 clear; close all; clc;
2 %% figura
3 fh = figure; % dkg = [0 0.5 0];
4 set(gcf,'Units','Normalized','Position',[ 0.0813, 0.1107, 0.8045, 0.7292]);
5 fh = fc_lib_XYZ_RGB_axis(fh,16);
6 grid on; hold all; O = [0;0;0];
7 %% vetor normal
8 % OBS: quivers estão usando 1 décimo a mais para obter a medida desejada
9 quiver3(0,0,0.5,0,0,1.1,'k');
10 str = '$\hat{n}$';
11 text(0,0,1.1+0.2,str,'Interpreter','latex','FontSize',16);
12 % -- posicionando a câmera
13 AZ = 170; EL = 20; view([AZ,EL]);
14 %% vetor v
15 v = [1;1;1]; S = 0.2;
16 cor = [0.8 0 0.8];
17 str = '$\vec{v}$';
18 fh = fc_lib_plot_vetor_3D(fh,[O, v],str,16,v+[0;0;0.1],...
19     -11*pi/12,0,degtorad(45),degtorad(125),S,[cor;O'],'-',1);
20 %% rotação sobre n
21 th = pi/3; n = [0;0;1]; n = n/norm(n,2);
22 r = fc_lib_quat_angulo_vetor_rotacao(v,th,n);
23 str = '$\vec{v'}$';
24 fh = fc_lib_plot_vetor_3D(fh,[O, r],str,16,r+[-0;0.05;0.1],...
25     -11*pi/12,0,degtorad(45+60),degtorad(120),S,[cor;O'],'-',1);
26 %% Vetor ortogonal produto interno
27 w = fc_lib_quat_angulo_vetor_rotacao([1;1;1],pi/2,n);
28 str = '$\vec{w}$'; cor = [0.5 0 0];
29 fh = fc_lib_plot_vetor_3D(fh,[O, [w(1:2);0]],str,16,[w(1:2);0]+[0.1;0;0.1],...
30     -11*pi/12,0,degtorad(45+90),degtorad(90),S,[cor;O'],'-',1);
31 %% Projeção antes da rotação
32 P1 = [ v, [v(1:2); 0]];
33 P2 = [ v, [0; 0; v(3)]];
34 plot3(P1(1,:),P1(2,:),P1(3,:),':r');
35 plot3(P2(1,:),P2(2,:),P2(3,:),':r');
36 % --- vetores paralelo e perpendicular
37 cor = [0,0,1];
38 str = '$\vec{v}-\{\bot\}$';
39 fh = fc_lib_plot_vetor_3D(fh,[O, [v(1:2);0]],str,16,[v(1:2);0]+[0;0;0.15],...
40     -11*pi/12,0,degtorad(45),degtorad(90),S,[cor;O'],'--',1);
41 %% vec{v} parallel
42 quiver3(0,0,0,0,0,1.1,'--b');
43 str = '$\vec{v}-\{\parallel\}$';
44 text(0.15,0.15,0.7,str,'Interpreter','latex','FontSize',16);
45 %% Auxilio de orientação v'
46 P1 = [ r, [r(1:2); 0]];
47 P2 = [ r, [0, 0, r(3)]];
48 plot3(P1(1,:),P1(2,:),P1(3,:),':r');
49 plot3(P2(1,:),P2(2,:),P2(3,:),':r');
50 cor = [0,0,1];
51 str = '$\vec{v'}-\{\bot\}$';
52 fh = fc_lib_plot_vetor_3D(fh,[O, [r(1:2);0]],str,16,[r(1:2);0]+[0.28;0.05;0.1],...
53     -11*pi/12,0,degtorad(45+60),degtorad(90),S,[cor;O'],'--',1);
54 quiver3(0,0,0,0,0,1.1,'--c');
55 str = '$\vec{v'}-\{\parallel\}$';
56 text(-0.05,0.15,0.7,str,'Interpreter','latex','FontSize',16);
57 %% circunferência em z=0
58 t = pi/4:0.05:(3*pi/4 + 0.05);
59 x = sqrt(2)*cos(t); y = sqrt(2)*sin(t);
60 plot(x,y,'Color',[0 0.5 0],'LineStyle','--');
61 %% circunferência da seta de \phi
62 t = pi/4:0.05:(pi/4 + pi/3 + 0.05);
63 x = 0.3*cos(t); y = 0.3*sin(t);
64 cor = [0.5 0.5 0.5];
65 plot(x,y,'Color',cor,'LineStyle','-');
66 str = '\phi = 60°';
67 text(0.2,0.5,str);
68 % --- seta
69 xe = x(end); ye = y(end);
70 fc_lib_plot_ponta_setas_2D(fh,xe,ye,-11*pi/12,-2*pi/3-pi/4-0.1,S/2,cor,'-');

```

```

71 %% circunferência em z=0
72 t = pi/4:0.05:3*pi/4;
73 x = sqrt(2)*cos(t); y = sqrt(2)*sin(t); z = ones(size(t));
74 plot3(x,y,z,'Color',[0 0.5 0],'LineStyle','--');
75 %% perspectiva e salvar
76 axis equal;
77 saveas(gcf,'../Figuras/fig.rotacao.3D.decomposicao.paralelo.perpendicular','eps2');

```

Códigos geradores dos Exemplos de 3 a 8.

Algoritmo D.39: Geração dos exemplos de 3 a 8.

```

1 clear; close all; clc;
2 %% Dados e laço
3 t = [0 1 2 3]';
4 x = 2*t; y = zeros(size(t)); z = zeros(size(t));
5 flags = [1, 1, 0]; %[flag_plot_xy, flag_drawnow, flag_sphere]
6 pasta = 'Figuras.exem.rot';
7 if ~exist(pasta, 'dir'); mkdir(pasta); end
8 for k = 1:2 %1:2
9     for l = 1:8 %1:8
10        [TH, PHI, PSI] = fc_int_THPHIPSI(k,l);
11        o = fc_int_o_xyz(l);
12        fh = figure;
13        set(gcf,'Units','Normalized','Position',[0.05 0.05 0.9 0.85]);
14        fc_lib_plot_coordenadas_3D(x,y,z,TH,PHI,PSI,flags,o,fh);
15        titulo = fc_int_titulo(o,TH(end),PHI(end),PSI(end)); title(titulo);
16        fc_int_text(fh);
17        figname = sprintf('%s%sfig-o-%02d-ang-%02d',pasta,'/',l,k);
18        fig_ext = 'eps2';
19        saveas(gcf,figname,fig_ext);
20    end
21 end
22 %% função gerador do titulo
23 function titulo = fc_int_titulo(o,th,phi,psi)
24 titulo = sprintf('%s%s%s: \\phi-{\%s} = \%g°; \\theta-{\%s} = \%g°; \\psi-{\%s} = \%g°',...
25     upper(o(1)), upper(o(2)), upper(o(3)), ...
26     upper(o(1)), radtodeg(th), ...
27     upper(o(2)), radtodeg(phi), ...
28     upper(o(3)), radtodeg(psi));
29 end
30 %% função escrita na figura
31 function fc_int_text(fh)
32 figure(fh);
33 for i = 0:3
34     text(2*i+0.2,0.2,0,sprintf('\\xi-{\%d}',i));
35 end
36 end
37 %% ---
38 function [TH, PHI, PSI] = fc_int_THPHIPSI(k,l)
39 if k == 1
40     A = [0 pi/2 pi/2 pi/2]';
41     B = [0 0 pi/2 pi/2]';
42     C = [0 0 0 pi/2]';
43 elseif k == 2
44     A = [0 pi/4 pi/4 pi/4]';
45     B = [0 0 pi/2 pi/2]';
46     C = [0 0 0 pi/4]';
47 end
48 switch l
49     case 1
50         TH = A; PHI = B; PSI = C;
51     case 2
52         TH = A; PHI = C; PSI = B;
53     case 3
54         TH = B; PHI = A; PSI = C;
55     case 4
56         TH = C; PHI = A; PSI = B;
57     case 5
58         TH = C; PHI = B; PSI = A;
59     case 6

```

```
60         TH = B; PHI = C; PSI = A;
61     case 7
62         TH = B; PHI = C; PSI = A;
63     case 8
64         TH = C; PHI = B; PSI = A;
65     otherwise
66         TH = A; PHI = B; PSI = C;
67         disp('Erro');
68 end
69 end
70 %% ---
71 function o = fc_int_o_xyz(k)
72 lista = {'xyz', 'xzy', 'yxz', 'yzx', 'zyx', 'zxy', 'xxz', 'zyz'};
73 o = char(lista(k));
74 end
```

Apêndice E

Outros trabalhos e bibliotecas sobre Quatérnios em Matlab[®]

Ao longo da apostila, os quatérnios foram explicados e uma biblioteca em Matlab[®] foi construída.

A abordagem utilizada na criação da biblioteca dessa apostila seguiu a linha de aproveitar o isomorfismo entre \mathbb{H} e \mathbb{R}^4 com a ideia de criar funções de implementação simples e direta. Em nenhum momento foi criada uma nova classe ou um novo objeto do tipo “quatérnio”. Algumas das bibliotecas listadas a seguir utiliza alguma variação dessas técnicas.

A seguir encontra-se uma lista de pacotes que implementam a álgebra de quatérnios em diferentes contextos.

- **quataternion**: Pacote criado por *Mark Tincknell* que cria uma classe em Matlab[®] para executar a álgebra de quatérnios incluindo matrizes de rotação em 3D e transformações entre as representações;
- **Quaternions**: Pacote criado por *Przemyslaw Baranski* que implementa algumas funções de quatérnios em Matlab[®];
- **Quaternion Toolbox**: Pacote criado por *Jay St. Pierre* que implementa algumas funções de quatérnios em Matlab[®];
- **Robotics TOOLBOX for MATLAB, quataternion**: *Toolbox* criada por *Peter I. Corke* para uso de robótica em Matlab[®], entre diversas coisas implementadas encontra-se um bloco de quatérnios, inclusive com artigo no estilo de propaganda publicado em ([Corke, 2007](#));
- **Flight Parameters and Quaternion Math**: Pacotes da biblioteca *Aerospace Toolbox* do Matlab[®];
- **Rotations, Orientation, and Quaternions**: Explicações sobre rotações em 2D e 3D da biblioteca *Sensor Fusion and Tracking Toolbox* do Matlab[®].

Referências Bibliográficas

ACADEMICS, U. D. **Quaternions Lecture 09: The application of Unit Quaternions to rotations**. UC Davis Academics, 2014. Disponível em: <https://youtu.be/mHVwd8gYLnI>. Acesso em: 20 jan. 2018.

BAKER, A. An introduction to matrix groups and their applications. **School of Mathematics and Statistics - University of Glasgow**, Jun 2000. Disponível em: <http://www.maths.gla.ac.uk/~ajb/dvi-ps/lie-bern.pdf>. Acesso em: 25 fev. 2019.

BIASI, S. C. de; GATTASS, M. Utilização de quatérnios para representação de rotações em 3d. **PUC-RIO**, 2002. Disponível em: <https://webserver2.tecgraf.puc-rio.br/~mgattass/Quaternios.pdf>. Acesso em: 20 jan. 2018.

BÖGE, A. et al. **Technische Mechanik: Statik - Reibung - Dynamik - Festigkeitslehre - Fluidmechanik**. Germany: Springer Fachmedien Wiesbaden, 2017. (SpringerLink: Bücher). ISBN 9783658162030.

BUCHMANN, A. A brief history of quaternions and the theory of holomorphic functions of quaternionic variables. **Journal of Fixed Point Theory and Applications**, Mathematical Association of America, 2011. Disponível em: https://www.maa.org/sites/default/files/pdf/upload_library/46/HOMSIGMAA/Buchmann.pdf. Acesso em: 18 mar. 2019.

CAMERON, P. J. Matrix groups. **School of Mathematical Sciences - Queen Mary, University of London**, Jun 2006. Disponível em: <http://www.maths.qmul.ac.uk/~pjc/preprints/mgo.pdf>. Acesso em: 25 fev. 2019.

CONCI, A. A eficiência das rotações em c.g.: quatérnios: Cap 2 (apêndice). **UFF**, 2014. Disponível em: <http://www2.ic.uff.br/~aconci/quaternionUFF.pdf>. Acesso em: 20 jan. 2018.

Corke, P. Matlab toolboxes: robotics and vision for students and teachers. **IEEE Robotics Automation Magazine**, v. 14, n. 4, p. 16–17, Dec 2007. ISSN 1070-9932. Disponível em: <https://ieeexplore.ieee.org/abstract/document/4437745>. Acesso em: 08 maio 2019.

FAREBROTHER, R. W.; GROß, J.; TROSCHKE, S.-O. Matrix representation of quaternions. **Linear Algebra and its Applications**, v. 362, p. 251 – 255, 2003. ISSN 0024-3795. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0024379502005359>. Acesso em: 25 fev. 2018.

FISCHLER, S. Algebraic and transcendental numbers. **Département de Mathématiques et Applications - École Normale Supérieure**, Out 2000. Disponível em: https://www.math.u-psud.fr/~fischler/inde/inde_fischlerpondichery.pdf. Acesso em: 25 fev. 2019.

GUIO, P. Levi-civita symbol and cross product vector/tensor. **Department of Physics and Astronomy - University College London**, Out 2011. Disponível em: <http://www.ucl.ac.uk/~uca/ppgu/seminars/levi-civita.pdf>. Acesso em: 01 jan. 2019.

HAMILTON, W. R. On quaternions; or on a new system of imaginaries in algebra. **Edinburgh, and Dublin Philosophical Magazine and Journal of Science**, London, v. 25, n. 3, Ago 1844. Disponível em: <https://www.maths.tcd.ie/pub/HistMath/People/Hamilton/OnQuat/OnQuat.pdf>. Acesso em: 21 jan. 2018.

- JACKSON, B. **Sec. 10.9 - Euler Angles**. Brian Jackson, 2017. Disponível em: <https://youtu.be/PLWfDgX9E6s>. Acesso em: 20 jan. 2018.
- MATHOMA. **Euler's Formula for the Quaternions**. Mathoma, 2015. Disponível em: <https://youtu.be/88BA8aO3qXA>. Acesso em: 20 jan. 2018.
- MATHOMA. **3D Rotations and Quaternion Exponentials: Special Case**. Mathoma, 2016. Disponível em: <https://youtu.be/UaK2q22mMEg>. Acesso em: 20 jan. 2018.
- MEZA, M.; ZUCATELLI, F. H. G. Compensators design utilizing the frequency response methods and generating a summary report in latex: Interactive graphical user interface. **International Journal of Computer Science and Software Engineering**, USA, v. 5, n. 7, p. 126–134, Jul 2016. Disponível em: <http://ijcsse.org/published/volume5/issue7/p2-V5I7.pdf>. Acesso em: 04 ago. 2016.
- PALAI, B.; PALAI, R. Euler's fixed point theorem: The axis of a rotation. **Journal of Fixed Point Theory and Applications**, Springer, v. 2, n. 2, p. 215–220, 2007. Disponível em: <https://link.springer.com/article/10.1007/s11784-007-0042-5>. Acesso em: 26 fev. 2019.
- PAMPENA, S. **Transcendental Numbers - Numberphile**. Numberphile, 2013. Disponível em: <https://youtu.be/seUU2bZtfgM>. Acesso em: 07 mar. 2019.
- PARKER, M. **All the Numbers - Numberphile**. Numberphile, 2019. Disponível em: <https://youtu.be/5TkIe60y2GI>. Acesso em: 07 mar. 2019.
- PERUMAL, L. Euler angles: conversion of arbitrary rotation sequences to specific rotation sequence. **Computer Animation and Virtual Worlds**, v. 25, n. 5-6, p. 521–529, 2014. Disponível em: <https://onlinelibrary.wiley.com/doi/full/10.1002/cav.1529>. Acesso em: 25 fev. 2019.
- ROBOTICS, A. **3 3 Euler Angles University of Pennsylvania Coursera**. AERIAL ROBOTICS, 2017. Disponível em: <https://youtu.be/3Zjf95Jw2UE>. Acesso em: 20 jan. 2018.
- ROBOTICS, A. **3 4 Axis Angle Representations for Rotations University of Pennsylvania Coursera**. AERIAL ROBOTICS, 2017. Disponível em: <https://youtu.be/zrDCI89bSp4>. Acesso em: 20 jan. 2018.
- ROSA, M. A. de F. Parte 2: quatérnios, a generalização quadrimensional. **IME - Unicamp**, Ago 2011. Disponível em: <https://www.ime.unicamp.br/~marcio/ss2011/ma770/cpxqtn/cq2.htm>. Acesso em: 20 jan. 2018.
- SALVADO, C. D. **Teorema Fundamental da Álgebra: Ferramentas para Demonstrar para Alunos do Ensino Médio**. Dissertação (Trabalho de Graduação (TCC)) — IMPA, 2016. Disponível em: https://impa.br/wp-content/uploads/2016/12/TCC_Claudio_Salvado.pdf. Acesso em: 25 jan. 2018.
- SANDERSON, G. **What does it feel like to invent math?** 3Blue1Brown, 2015. Disponível em: <https://youtu.be/XFDM1ip5HdU>. Acesso em: 20 jan. 2018.
- SANDERSON, G. **Quaternions and 3d rotation, explained interactively**. 3Blue1Brown, 2018. Disponível em: <https://youtu.be/zjMuIxRvygQ>. Acesso em: 26 out. 2018.
- SANDERSON, G. **What are quaternions, and how do you visualize them? A story of four dimensions**. 3Blue1Brown, 2018. Disponível em: <https://youtu.be/d4EgbgTm0Bg>. Acesso em: 06 set. 2018.
- STEVENS, M. **How To Count Past Infinity**. Vsauce, 2016. Disponível em: <https://youtu.be/SrU9YDoXE88>. Acesso em: 20 jan. 2018.
- TOLEDO, A. F. D. **Teorema Fundamental da Álgebra: Uma Abordagem Visual para o Ensino Médio**. Dissertação (mestrado) — Universidade Federal do ABC - PROFMAT, 2017. Disponível em: https://sca.profmato-sbm.org.br/sca_v2/get_tcc3.php?id=72759. Acesso em: 25 jan. 2018.

- TULEY, I. **Complex Numbers**. Isis Tuley, 2017. Disponível em: <http://slideplayer.com/slide/1664745/>. Acesso em: 28 jan. 2018.
- WEISSTEIN, E. W. Algebraic number. **MathWorld—A Wolfram Web Resource**, jan. 2015. Disponível em: <http://mathworld.wolfram.com/AlgebraicNumber.html>. Acesso em: 26 fev. 2019.
- WEISSTEIN, E. W. Hypercomplex number. **MathWorld—A Wolfram Web Resource**, jan. 2015. Disponível em: <http://mathworld.wolfram.com/HypercomplexNumber.html>. Acesso em: 01 jan. 2019.
- WEISSTEIN, E. W. Spherical coordinates. **MathWorld—A Wolfram Web Resource**, Ago 2015. Disponível em: <http://mathworld.wolfram.com/SphericalCoordinates.html>. Acesso em: 20 jan. 2018.
- WEISSTEIN, E. W. Euler angles. **MathWorld—A Wolfram Web Resource**, jan. 2019. Disponível em: <http://mathworld.wolfram.com/EulerAngles.html>. Acesso em: 26 fev. 2019.
- WIKIPÉDIA. Algebraic numbers. **Wikipedia**, fev. 2018. Disponível em: https://en.wikipedia.org/wiki/Algebraic_number. Acesso em: 26 fev. 2019.
- WIKIPÉDIA. Espaço l_p . **Wikipedia**, dez 2018. Disponível em: https://pt.wikipedia.org/wiki/Espa%C3%A7o_Lp. Acesso em: 20 dez. 2018.
- WIKIPÉDIA. Hilbert space. **Wikipedia**, dez. 2018. Disponível em: https://en.wikipedia.org/wiki/Hilbert_space. Acesso em: 20 dez. 2018.
- WIKIPÉDIA. L_p space. **Wikipedia**, dez. 2018. Disponível em: https://en.wikipedia.org/wiki/Lp_space. Acesso em: 20 dez. 2018.
- WIKIPÉDIA. Norma (matemática). **Wikipedia**, jun. 2018. Disponível em: [https://pt.wikipedia.org/wiki/Norma_\(matem%C3%A1tica\)](https://pt.wikipedia.org/wiki/Norma_(matem%C3%A1tica)). Acesso em: 30 jun. 2018.
- WIKIPÉDIA. Hyperreal number. **Wikipedia**, jan. 2019. Disponível em: https://en.wikipedia.org/wiki/Hyperreal_number. Acesso em: 01 jan. 2019.
- WIKIPÉDIA. Levi-civita symbol. **Wikipedia**, jan. 2019. Disponível em: https://en.wikipedia.org/wiki/Levi-Civita_symbol. Acesso em: 01 jan. 2019.
- WIKIPÉDIA. List of things named after leonhard euler. **Wikipedia**, jan. 2019. Disponível em: https://en.wikipedia.org/wiki/List_of_things_named_after_Leonhard_Euler#Euler's_theorems. Acesso em: 26 fev. 2019.
- WIKIPÉDIA. Rodrigues' formula - ivory-jacobi formula. **Wikipedia**, jan. 2019. Disponível em: https://en.wikipedia.org/wiki/Rodrigues%27_formula. Acesso em: 05 jan. 2019.
- WIKIPÉDIA. Surreal number. **Wikipedia**, jan. 2019. Disponível em: https://en.wikipedia.org/wiki/Surreal_number. Acesso em: 01 jan. 2019.
- ZUCATELLI, F. H. G.; MEZA, M. E.; FENILI, A. Lagrantexpac: A software tool to obtain the dynamic equations of mechanical systems. **International Journal of Computer Theory and Engineering**, Canada, v. 9, n. 4, p. 242–249, Ago 2017. ISSN 1793-8201. Disponível em: <http://www.ijcte.org/index.php?m=content&c=index&a=show&catid=87&id=1399>. Acesso em: 20 jul. 2017.