

M. Eng. Fernando Henrique Gomes Zucatelli
Gabriela Bittencourt

Apostila de Quatérnios e Rotações em 3D com
Matlab®

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	6
2.3.1	Soma	6
2.3.2	Multiplicação por escalar	7
2.3.3	Parte escalar de um quatérnio	7
2.3.4	Parte vetorial de um quatérnio	7
2.3.5	Notação de quatérnio com parte escalar e vetorial	8
2.3.6	Conjugado	8
2.3.7	Módulo	8
2.3.8	Multiplicação entre quatérnios	9
2.3.9	Divisão entre quatérnios	10
3	Quatérnios como “complexo de complexo”	12
3.1	Isomorfismo entre quatérnios, plano complexo e espaço real de 4 dimensões	12
3.2	Representação de quatérnios com matrizes	13
4	Rotações	15
4.1	Rotações 2D	15
4.2	Rotações 3D - Ângulos de Euler	15
4.2.1	Exemplos de Rotações 3D com Ângulos de Euler - Revisando	18
4.3	Rotações 3D - Teorema de Euler	22
4.3.1	Teorema de Euler	22
4.4	Rotações 3D - Fórmula de Rodrigues	23
4.4.1	Entendendo o grupo de rotação $SO(3)$	26
4.5	Rotações 2D - Fórmula de Euler	26
4.6	Rotações 3D - Fórmula de Euler para Quatérnios	27
4.7	Rotações 3D - <i>Fórmula de Rodrigues</i> para Quatérnios	28
A	Resoluções de Exercícios Propostos	33
A.1	Exercício 6	33
A.2	Exercício 8	34
A.3	Exercício 9	34
A.4	Exercícios 22, 23, 24, 25	34
A.4.1	Código de resolução com código substituidor de expressões e código gerador de arquivos a partir de uma <i>string</i>	34
A.5	Exercício 30	35
A.6	Exercício 33	35
A.7	Exercício 43	36
A.8	Exercício 44	36

B	Normas	37
C	Códigos em Matlab para Álgebra de Quatérnios	41
C.1	Códigos com álgebra de quatérnios	41
C.2	Códigos com <i>plot</i> de coordenadas	43
C.3	Códigos com matrizes de rotação	45
C.4	Códigos auxiliares	47
C.5	Códigos geradores de Figuras	48

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
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} .	12
4.1	Exemplo de rotação em 2D com uso de números complexos.	16
4.2	Definição matriz de rotação R_x .	16
4.3	Definição matriz de rotação R_y .	17
4.4	Definição matriz de rotação R_z .	17
4.5	Três passos da rotação XYZ com ângulos de 90° .	19
4.6	Três passos da rotação XYZ com ângulos de 45° , 90° e 45° respectivamente.	19
4.7	Três passos da rotação XZY .	20
4.8	Três passos da rotação YXZ .	20
4.9	Três passos da rotação ZYX .	21
4.10	Três passos da rotação ZYZ .	21
4.11	Ilustração do Teorema de Euler.	22
4.12	Rotação do vetor \vec{v} , decomposto em suas projeções ortogonal e perpendicular ao vetor \hat{n} , resultado no vetor \vec{v}' . Repare que as projeções paralelas são iguais antes e após a rotação ao redor de \hat{n} .	23
4.13	Representação do grupo $SO(3)$ como esfera de raio π .	26
4.14	Resultado da rotação de \hat{e}_3 ao redor de \hat{e}_2 por um ângulo de $\frac{\pi}{2}$ rad.	30
B.1	Desigualdade triangular.	37
B.2	Ilustração dos casos com $\ \cdot\ _p = 1$ para as normas $p = \{1, 2, \infty\}$.	38
B.3	Exemplo de normas.	40

Lista de Algoritmos

2.1	Exemplo soma de quatérnios em Matlab [®]	6
2.2	Exemplo produto por escalar de quatérnios em Matlab [®]	7
2.3	Exemplo de extração da parte escalar de quatérnios em Matlab [®]	7
2.4	Exemplo de extração da parte vetorial de quatérnios em Matlab [®]	7
2.5	Exemplo de soma de escalar com vetor no Matlab [®]	8
3.1	Exemplo do isomorfismo descrito pela Eq. (3.6).	13
3.2	Exemplo do isomorfismo descrito pela Eq. (3.9).	14
4.1	Exemplo de teste básico de tempo de execução em Matlab [®]	18
A.1	Resolução dos Exercícios 24, 23, 24 e 25 com uso de simbólico no Matlab [®] e integração com L ^A T _E X	34
C.1	Rotação de um vetor por um ângulo com quatérnios.	41
C.2	Conjugado de quatérnio.	41
C.3	Divisão de quatérnio à direita.	41
C.4	Divisão de quatérnio à esquerda.	41
C.5	Inverso de quatérnio.	41
C.6	Módulo de quatérnio.	42
C.7	Plot coordenadas 3D com quatérnios.	42
C.8	Produto de quatérnios.	42
C.9	Rotação de quatérnios.	43
C.10	Plot de coordenadas 2D.	43
C.11	Plot de coordenadas 3D.	43
C.12	Plot setas 2D.	44
C.13	Plot setas 3D.	45
C.14	Rotação Fórmula de Rodrigues.	46
C.15	Matriz de rotação sobre eixo X	46
C.16	Matriz de rotação sobre eixo Y	46
C.17	Matriz de rotação sobre eixo Z	46
C.18	Matriz de rotação na sequência XYZ	46
C.19	Matriz de rotação na sequência XZY	46
C.20	Matriz de rotação na sequência YXZ	46
C.21	Matriz de rotação na sequência YZX	46
C.22	Matriz de rotação na sequência ZXY	47
C.23	Matriz de rotação na sequência ZYX	47
C.24	Matriz de rotação na sequência ZXZ	47
C.25	Matriz de rotação na sequência ZYZ	47
C.26	Encapsulamento da função que gera arquivos em L ^A T _E X a partir de uma <i>string</i>	47
C.27	Nomenclatura e padronização dos rótulos dos eixos XYZ	47
C.28	Figura do exemplo de rotação 3D em Matlab [®]	48
C.29	Figura do exemplo de rotação 2D em Matlab [®]	49
C.30	Figura do exemplo de coordenadas 2D em Matlab [®]	50
C.31	Inserção da ponta da seta em um ponto qualquer em 2D.	51
C.32	Figura do exemplo de coordenadas 3D em Matlab [®]	51
C.33	Inserção da ponta da seta em um ponto qualquer em 3D.	52
C.34	Figura do isomorfismo $\mathbb{R}^2 \cong \mathbb{C}$	52
C.35	Figura do isomorfismo $\mathbb{R}^4 \cong \mathbb{C}^2$	53

C.36 Geração dos exemplos de 3 a 8. 53

C.37 Figura do isomorfismo $\mathbb{R}^4 \cong \mathbb{C}^2$ 55

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.

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

A [Figura 1.1](#) ilustra as relações entre esses dois sistemas de coordenadas. O código gerador encontra-se no [Apêndice C.5](#).

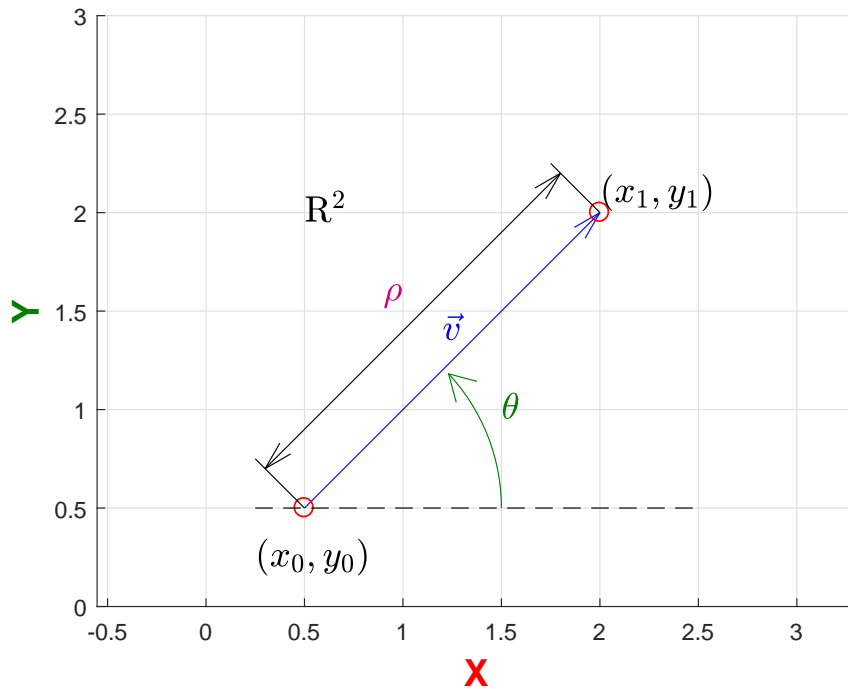


Figura 1.1: Vetor \vec{v} em duas dimensões com tamanho ρ . Elaborado em Matlab[®].

A transformação de coordenadas polares para cartesianas é dada por

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

e a de coordenadas polares para cartesianas 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 função (ou comando) `atan2`¹ é conhecido como arco tangente de quatro quadrantes 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_to_cartesiana(ρ, θ).`

Código 2 Faça um código para realizar a conversão de coordenadas cartesianas para polares:
`[ρ, θ] = fc_cartesiana_to_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 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, 2015).

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.

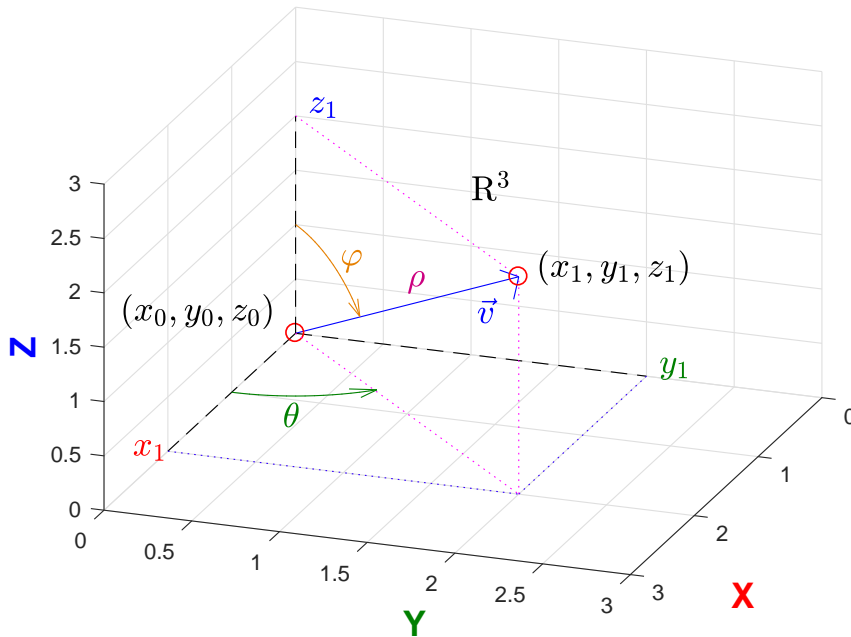


Figura 1.2: Vetor \vec{v} em três dimensões com tamanho ρ . Elaborado em Matlab®

¹Verifique no Matlab®: “help atan2”.

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 polares para cartesianas 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

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

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

$[x, y, z] = \text{fc_esfericas_to_cartesiana}(\rho, \theta, \varphi)$.

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

$[\rho, \theta, \varphi] = \text{fc_cartesiana_to_polar}(x, y, z)$.

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 desenha a seta dada pelo par (ρ, θ, φ) :

$\text{fc_setas_3D}(x_0, y_0, z_0, \rho, \theta, \varphi)$.

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 lembrando as relações dos conjuntos numéricos até os complexos

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

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 a divisão já não é mais descrita por 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 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

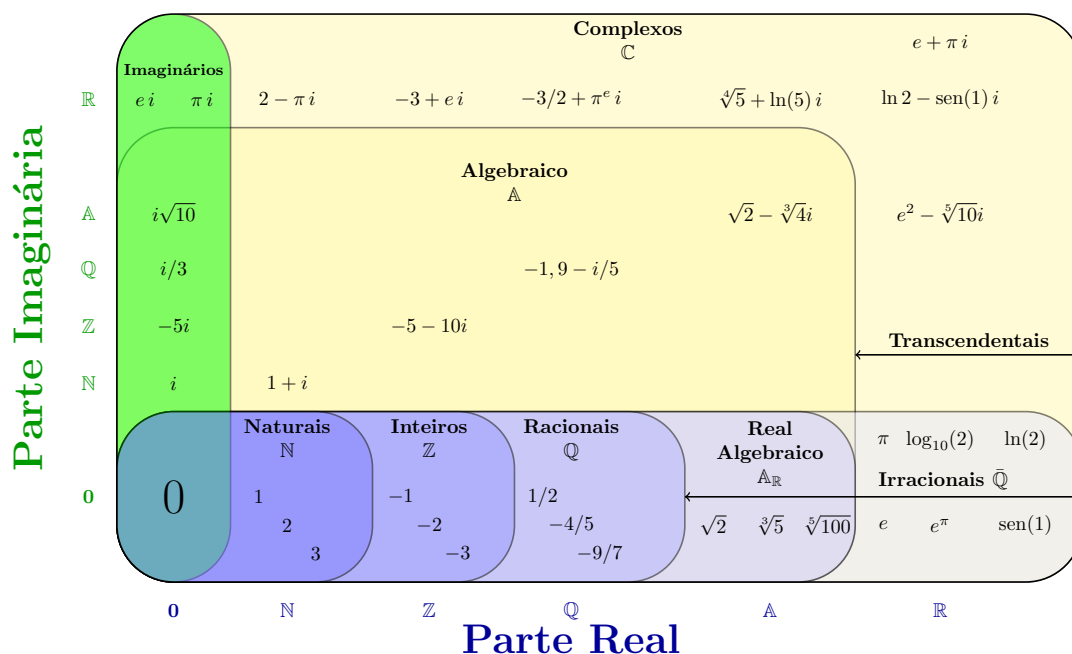


Figura 2.1: Diagrama de Venn dos principais subconjuntos numéricos com complexos.

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. Os algebraicos são todos os números que são soluções de (2.1) com a condição de $a_i \in \mathbb{Q}$. Os números que não podem ser obtidos com essa restrição são os denominados transcendentais.

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, 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 quatérnio, são exemplos de hipercomplexos os bi-quatérnios, álgebras exteriores, álgebras de grupo, matrizes e octônios ([WEISSTEIN, 2019](#)).

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, 2019d](#))
- Super-reais ([WIKIPÉDIA, 2019a](#))

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

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 do canal de YouTube® **3Blue1Brown** ([SANDERSON, 2018b](#); [SANDERSON, 2018a](#)) que explicam sobre quatérnios do ponto de vista geométrico, apresentando com belas visualizações 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. Hamilton formulou em 1844 ([HAMILTON, 1844](#)).

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 dos produtos das unidades imaginárias (i, j, k) .

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

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.

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

Exercício 1 *Mostre que $ijk = -1$*

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

2.3 Operações com quatérnios

2.3.1 Soma

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

$$\begin{aligned} q &= q_0 e_0 + q_1 e_1 + q_2 e_2 + q_3 e_3 \\ p &= p_0 e_0 + p_1 e_1 + p_2 e_2 + p_3 e_3 \\ (q \pm p) &= (q_0 \pm p_0) e_0 + (q_1 \pm p_1) e_1 + (q_2 \pm p_2) e_2 + (q_3 \pm p_3) e_3 \end{aligned} \quad (2.9)$$

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

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

Algoritmo 2.1: 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

2.3.2 Multiplicação por escalar

Seja $k \in \mathbb{K} = \mathbb{R}$, o produto por escalar é definido.²

$$p = kq = (kq_0)e_0 + (kq_1)e_1 + (kq_2)e_2 + (kq_3)e_3 \quad (2.10)$$

Código 8 A operação de produto por escalar se preserva entre os quatérnios e o \mathbb{R}^4 conforme segue exemplo em Matlab[®]

Algoritmo 2.2: Exemplo produto por escalar de quatérnios em Matlab[®]

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

cujos resultados são:

p =

5 10 15 20

2.3.3 Parte escalar de um quatérnio

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

$$S(q) = S(a + xi + yj + zk) = a \in \mathbb{R} \quad (2.11)$$

Código 9 Em Matlab[®], basta extrair a primeira posição do elemento conforme o exemplo:

Algoritmo 2.3: Exemplo de extração da parte escalar de quatérnios em Matlab[®]

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

cujos resultados são:

a =

10

2.3.4 Parte vetorial de um quatérnio

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

$$V(q) = V(a + xi + yj + zk) = xi + yj + zk = \vec{v} \in \mathbb{R}^3 \quad (2.12)$$

Código 10 Em Matlab[®], basta extrair o vetor composto da segunda à quarta posição do elemento conforme o exemplo:

Algoritmo 2.4: Exemplo de extração da parte vetorial de quatérnios em Matlab[®]

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

cujos resultados são:

v =

20 30 40

²O corpo \mathbb{K} utilizado ao longo da apostila será o corpo dos reais \mathbb{R} .

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 auxílios das 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. O Matlab[®] redefine essa soma para facilitar o uso e no caso dos quatérnios a notação explicita o fato de um quatérnio ser formado pelas partes escalar e vetorial. A notação que usaremos para evidenciar as partes escalar e vetorial e evitar qualquer problema de interpretação será

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

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 11 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.³

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 $\mathbf{0}_{\mathbb{H}}$.⁴

Código 12 Crie um função que retorna o módulo de um quatérnio.

`r = fc_quat_modulo(q)`

Dica: “*help norm*”.

³Mais precisamente, norma é aplicado a qualquer elemento de espaço vetorial, o que inclui matrizes, tensores e funções.

⁴ \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, $pq \neq qp$, porém, a vale propriedade distributiva.⁵

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.⁶

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

apliquemos os resultados da Tabela 2.1

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

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

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

Sejam $\vec{v} = [v_1, v_2, v_3]^\top$ e $\vec{u} = [u_1, u_2, u_3]^\top \in \mathbb{R}^3$, define-se o produto interno entre vetores do \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 vetores do \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 13 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 3 Mostre que no caso geral $qp \neq pq$.

Exercício 4 Apresente um exemplo em que $qp \neq pq$.

⁵OBS: 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.

⁶Os autores gostam e recomendam essa forma de escrita tabulada por entenderem que ela facilita consideravelmente a compreensão das relações importantes da operação.

Exercício 5 *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 6 (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 7 (Desafio) *Como verificar numericamente que $\forall q, p \in \mathbb{H}, S(qp) = S(pq)$?*

Exercício 8 *Mostre que $\bar{q}q = q\bar{q}, \forall q \in \mathbb{H}$.*

Exercício 9 *Mostre que $\bar{q}q = |q|^2, \forall q \in \mathbb{H}$.*

Exercício 10 *Mostre que o produto de quatérnios é associativo, i.e., $\forall q, p, r \in \mathbb{H}, (qp)r = q(pr)$.*

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

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 11 (Desafio) *Mostre que $\overline{(qp)} = \bar{p}\bar{q}$*

Exercício 12 (Bônus) *Usemos por um breve momento vetores que admitam componentes complexas. Seja $\langle \underline{q}, \underline{p} \rangle$ o produto interno de quatérnio tal que $\langle \underline{q}, \underline{p} \rangle = \underline{q}^H \underline{p}$, onde $\underline{q} = [a, xi, yj, zk]^T$ e $\underline{q}^H = (\underline{q}^T)^* = (\bar{\underline{q}})^T$ é o **transposto conjugado**⁷ de \underline{q} , i.e., calcula-se o conjugado do transposto de \underline{q} que por sua vez é igual ao transposto do conjugado de \underline{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 sua norma $|\hat{q}| = 1$. Para se obter norma unitária 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 8, 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

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

cujo resultado combinado com o Exercício 9 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

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

cujas propriedades são idênticas 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.

⁷também denotado por $(\cdot)^\dagger$.

Código 14 Crie uma função que retorna o inverso de um quatérnio.

$r = \text{fc_quat_inverso}(q)$

Código 15 Crie uma função que retorna o produto pelo inverso de um quatérnio à esquerda.

$r = \text{fc_quat_divisao_esq}(p, q)$

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

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

$r = \text{fc_quat_divisao_dir}(q, p)$

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

Exercício 13 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 14 Calcule o inverso de

(a) $q = \frac{1}{2} + \frac{1}{2}i + \frac{1}{2}j + \frac{1}{2}k$,

(b) $p = 0 + \frac{\sqrt{3}}{3}i + \frac{\sqrt{3}}{3}j + \frac{\sqrt{3}}{3}k$.

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 $\vec{v} = [v_x, v_y]^\top \in \mathbb{R}^2$ 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.

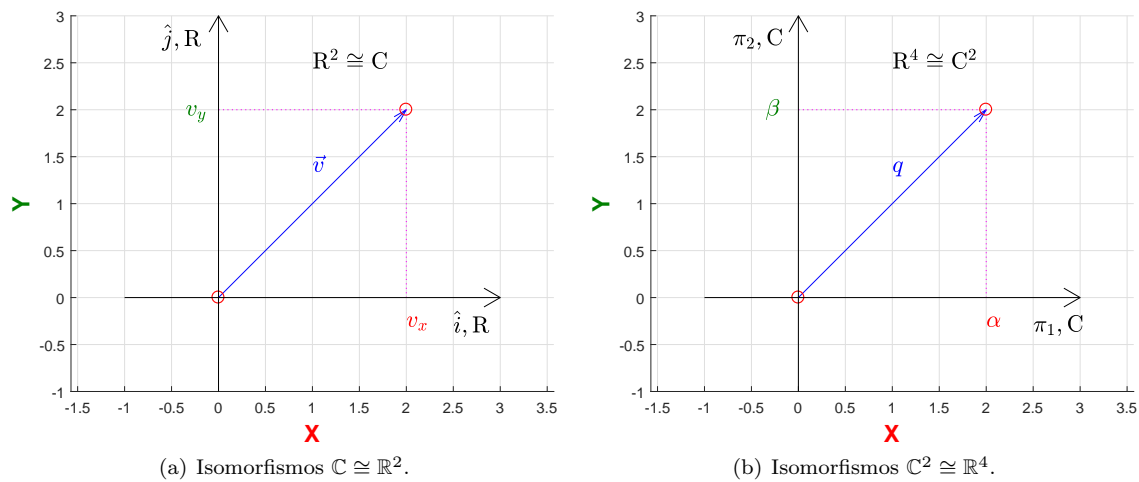


Figura 3.1: Isomorfismos $\mathbb{C} \cong \mathbb{R}^2$ e entre $\mathbb{C}^2 \cong \mathbb{R}^4$. \mathbb{C}^2 pode ser interpretado como um plano em que cada eixo é um plano complexo \mathbb{C} .

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

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

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.

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 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

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

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

a primeira coluna de A_3 é exatamente o número complexo z_3 .

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

$$M_1 = \begin{bmatrix} \alpha & \beta \\ -\bar{\beta} & \bar{\alpha} \end{bmatrix} = \begin{bmatrix} (q_0 + q_1 j) & (q_2 + q_3 j) \\ (-q_2 + q_3 j) & (q_0 - q_1 j) \end{bmatrix} = \left[\begin{array}{c|ccc} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{array} \right] \quad (3.9)$$

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 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 número complexo q_3 .

Exercício 15 *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.*

$$M_2 = \left[\begin{array}{c|ccc} q_0 & q_1 & q_2 & q_3 \\ \hline -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{array} \right] \quad (3.12)$$

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 16 *Mostre que $\det(M_1) = |q|^2$*

Exercício 17 *Mostre que $|\lambda q| = |\lambda| |q|, \forall \lambda \in \mathbb{R}$*

Exercício 18 *Mostre que $|pq| = |p| |q|, \forall p, q \in \mathbb{H}$*

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

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

para $z_2 = i \equiv 1 \angle (\pi/2)$ tem-se uma rotação de 90°

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

O caso em que $z_1 = 2 + 2i$ é ilustrado na [Figura 4.1](#), cujo módulo $\rho_1 = \sqrt{2}$ e o ângulo $\theta_1 = 45^\circ$. Repara 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 , guinagem (*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 rotações de um ângulo ϕ ao redor de x , de θ ao redor de y e ψ ao redor de z respectivamente. 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”.

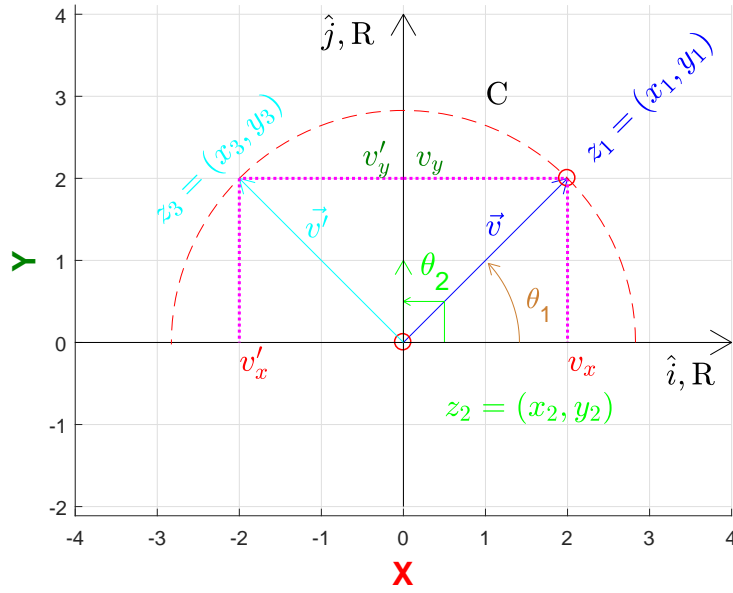


Figura 4.1: Exemplo de rotação em 2D com uso de números complexos.

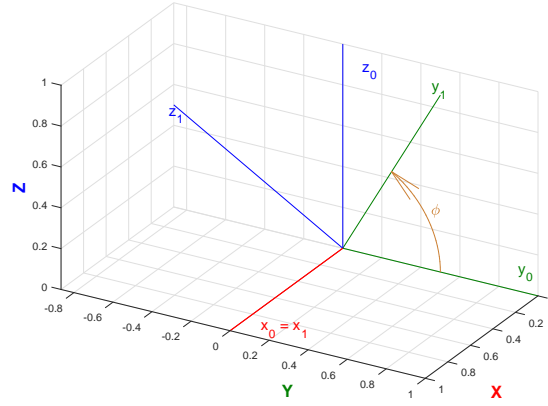


Figura 4.2: Definição matriz de rotação R_x .

Exercício 19 Para $R = \{R_x, R_y, R_z\}$

(a) *Mostre que* $RR^T = I$

(b) *Mostre que* $R^T R = I$

Exercício 20 *Mostre que* $\det(R) = 1$. Isso significa que R é uma matriz do grupo especial (Special).

Exercício 21 Do Exercício 19, 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 20, tem-se que R é uma matriz do grupo especial e ortogonal, ou seja, $R \in SO(n)$, com $n = 3$. $SO(n) = \{R \in \mathbb{R}^{n \times n} \mid RR^T = R^T R = I, n \in \mathbb{N}_*\}^1$.

Código 17 Faça uma função que retorne a matriz de rotação sobre o eixo **X** dado um ângulo ϕ .

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

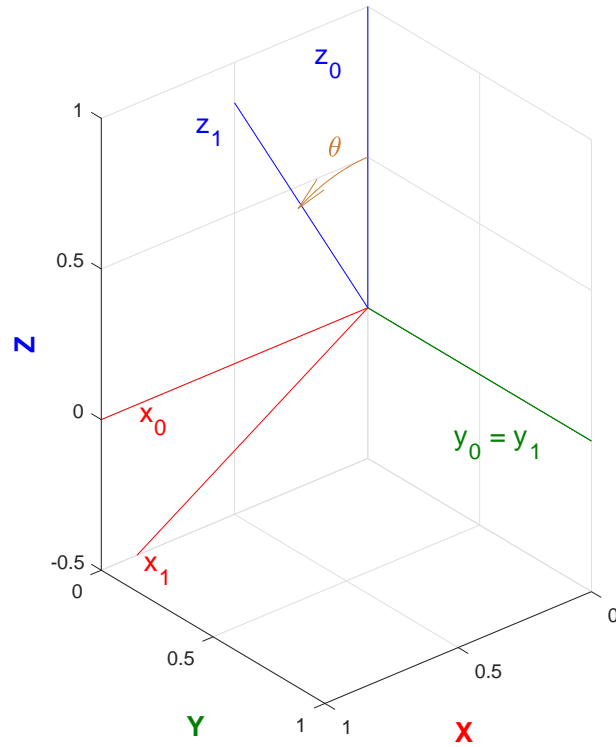
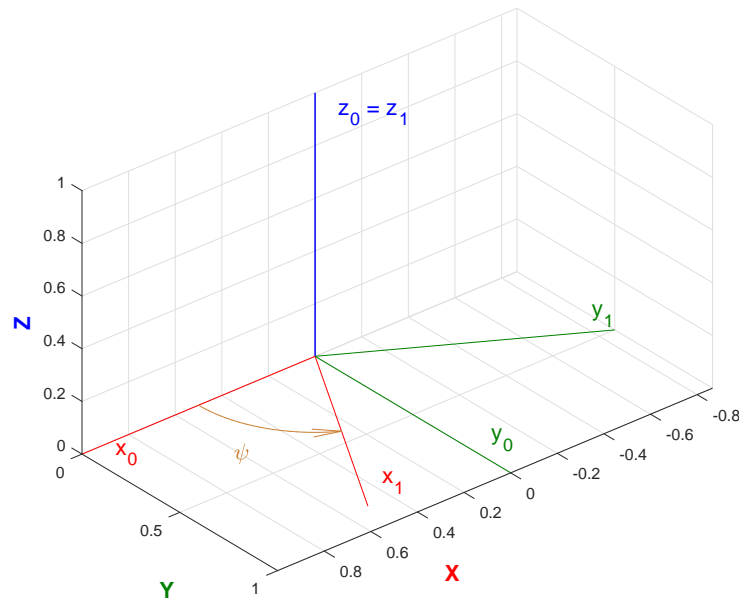
Código 18 Faça uma função que retorne a matriz de rotação sobre o eixo **Y** dado um ângulo θ .

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

Código 19 Faça uma função que retorne a matriz de rotação sobre o eixo **Z** dado um ângulo ψ .

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

¹ \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. Assim, uma sequência de rotação **XYZ** significa que a base original de coordenadas é rotacionado 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

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

Para se criar as matrizes de rotação de cada sequência é interessante aproveitar os códigos 17, 18 e 19, 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 20 O *Algoritmo 4.1* apresenta um exemplo de como testar variações de códigos curtos. O Matlab® possui a ferramenta *profile viewer* acessível pela linha de comando e pela opção *Run and Time* para analisar o tempo de execução de códigos. 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.

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

```

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

```

Um dos resultados foi:

```

Teste 1
Elapsed time is 0.371764 seconds.
Elapsed time is 0.739950 seconds.
Elapsed time is 1.122249 seconds.
Elapsed time is 1.549532 seconds.
Teste 2
Elapsed time is 0.153817 seconds.
Elapsed time is 0.263659 seconds.
Elapsed time is 0.367134 seconds.
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 de aleatoriedade na geração do teste e também do carregamento das funções na memória para que se possa obter uma conclusão mais precisa.

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.

4.2.1 Exemplos de Rotações 3D com Ângulos de Euler - Revisando

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 **X** = 0.

A base após a primeira parte da sequência de rotações é denotada por ξ_1 e se localiza em **X** = 2, a base após a segunda rotação é denotada por ξ_2 e localizada em **X** = 4 e finalmente o resultado das três rotações combinadas é denotado por ξ_3 e localizada em **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 **XYZ** com 90° ao redor de cada eixo. A primeira rotação é ao redor de **X** a segunda rotação ao redor de **Y** e a terceira é ao redor **Z**.

Os resultados da base a cada rotação são apresentadas a seguir

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

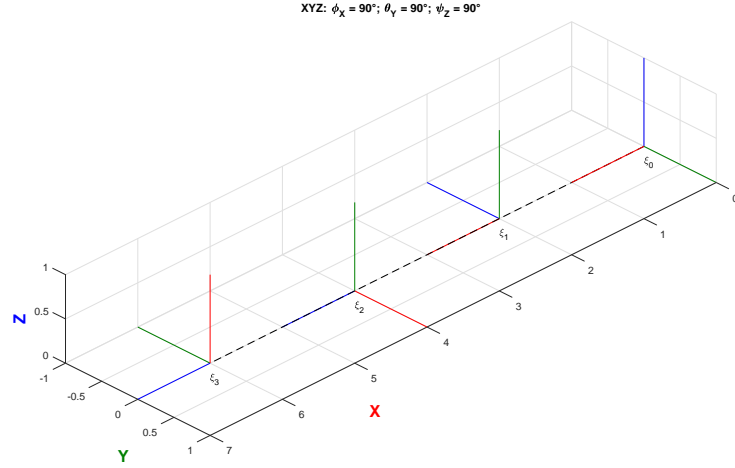


Figura 4.5: Três passos da rotação **XYZ** com ângulos de 90° .

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.

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

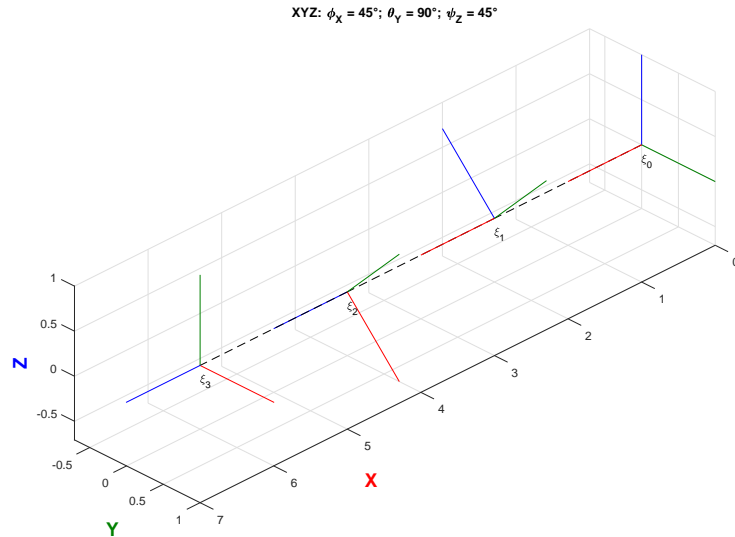


Figura 4.6: Três passos da rotação **XYZ** com ângulos de 45° , 90° e 45° respectivamente.

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

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

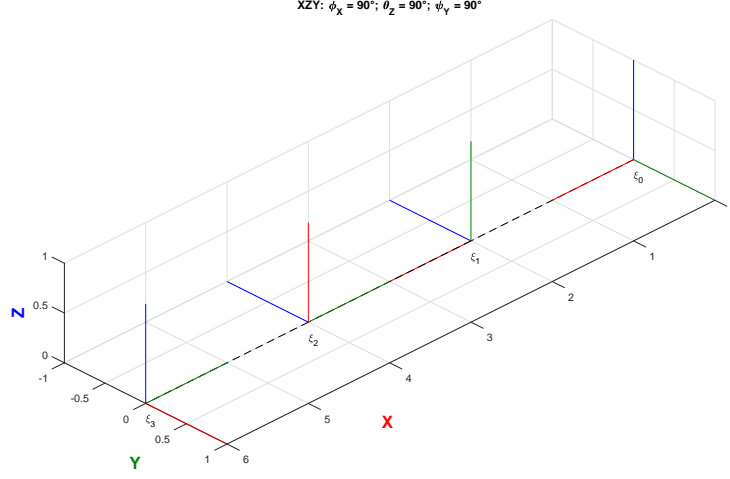


Figura 4.7: Três passos da rotação **XZY**.

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

Os resultados da base a cada rotação são apresentadas a seguir

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

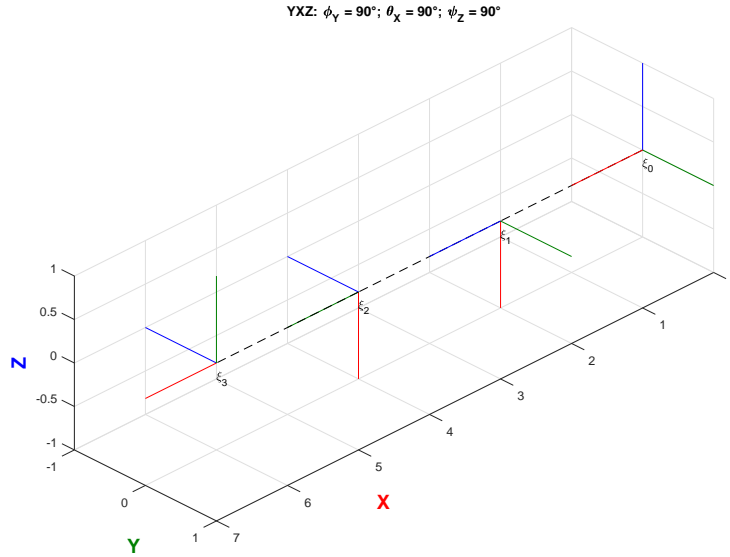
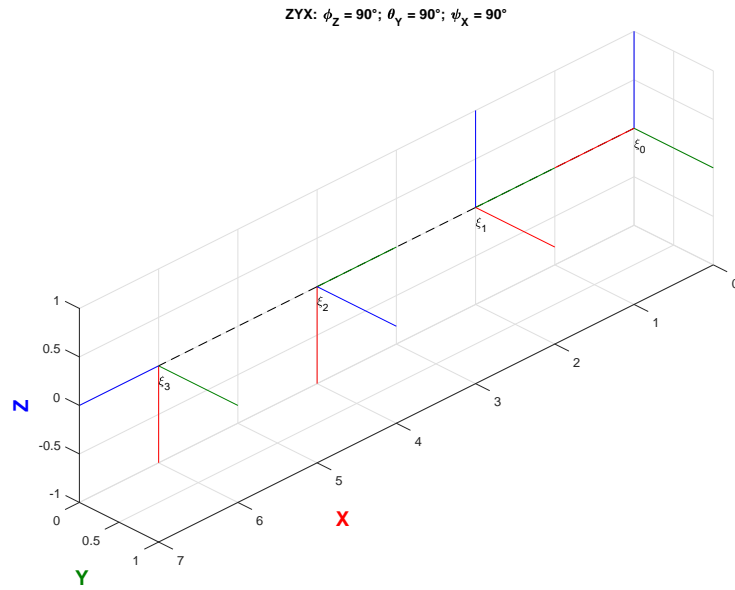


Figura 4.8: Três passos da rotação **YXZ**.

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

Os resultados da base a cada rotação são apresentadas a seguir

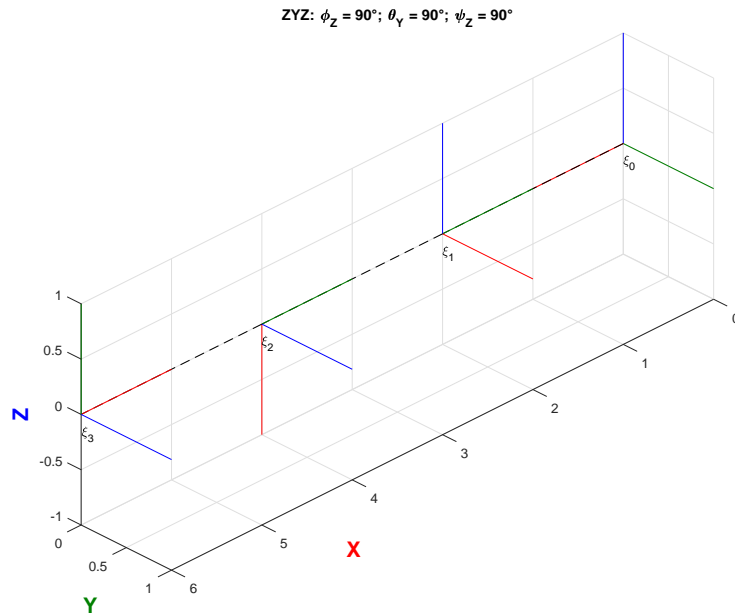
$$\xi_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \xi_1 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \xi_2 = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}; \quad \xi_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1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

Figura 4.9: Três passos da rotação ZYX .

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 .

Exercício 22 Obtenha a matriz de rotação $R_{xzy}(\phi, \psi, \theta) = R_x(\phi)R_z(\psi)R_y(\theta)$ do Exemplo 5.

Exercício 23 Obtenha a matriz de rotação $R_{yxz}(\phi, \theta, \psi) = R_y(\phi)R_x(\theta)R_z(\psi)$ do Exemplo 6.

Exercício 24 Obtenha a matriz de rotação $R_{zyx}(\psi, \phi, \theta) = R_z(\psi)R_y(\phi)R_x(\theta)$ do Exemplo 7.

Exercício 25 Obtenha a matriz de rotação $R_{zyz}(\psi, \phi, \theta) = R_z(\psi)R_y(\phi)R_z(\theta)$ do Exemplo 8.

Exercício 26 Refaça o Exemplo 5 com $\theta_X = 90^\circ, \phi_Z = 45^\circ, \psi_Y = 90^\circ$.

Exercício 27 Refaça o Exemplo 6 com $\theta_Y = 90^\circ, \phi_X = 45^\circ, \psi_Z = 90^\circ$.

Exercício 28 Refaça o Exemplo 7 com $\theta_Z = 90^\circ, \phi_Y = 45^\circ, \psi_X = 90^\circ$.

Exercício 29 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 outra forma de realizar as rotações.

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 .

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

de onde tem-se então a seguinte relação:

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

onde R denota a matriz de rotação que rotaciona \underline{p} em \underline{q} . A Figura 4.11 apresenta essa situação.

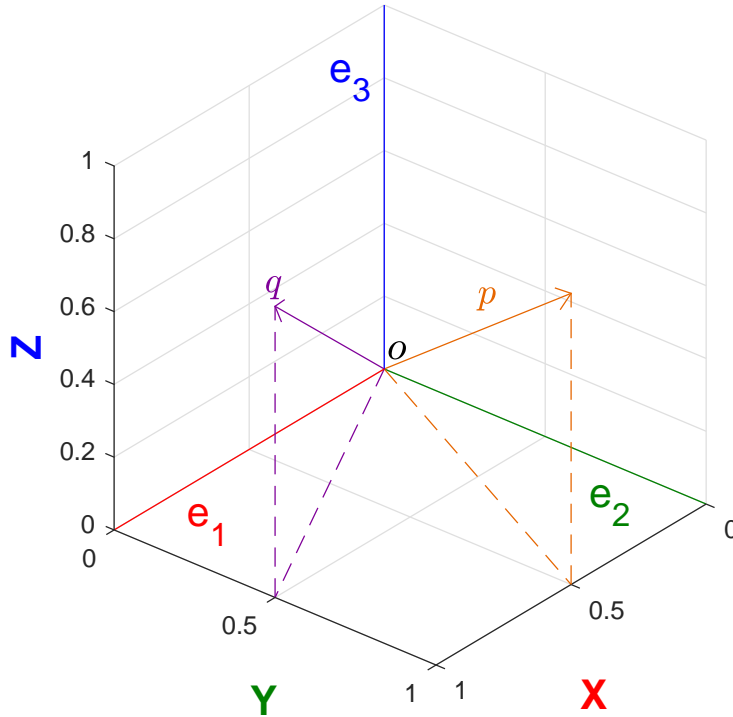


Figura 4.11: Ilustração do Teorema de Euler.

Exercício 30 Verifique que $\lambda = 1$ é autovalor para qualquer matriz de rotação R .

4.4 Rotações 3D - Fórmula de Rodrigues

O problema inverso 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.

BATATA 20190112: Início do trecho que necessita de revisão

A **Figura 4.12** apresenta a rotação do vetor \vec{v} ao redor de \hat{n} visualizando a componentes paralela a \hat{n} , \vec{v}_{\parallel} , e a componente perpendicular a \hat{n} , \vec{v}_{\perp} (BIASI; GATTASS, 2002). O código gerador da **Figura 4.12** encontra-se no Apêndice C.5.

A nomenclatura utilizada na **Figura 4.12** é apresentada a seguir

- \hat{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 $\hat{n} \times \vec{v}$.

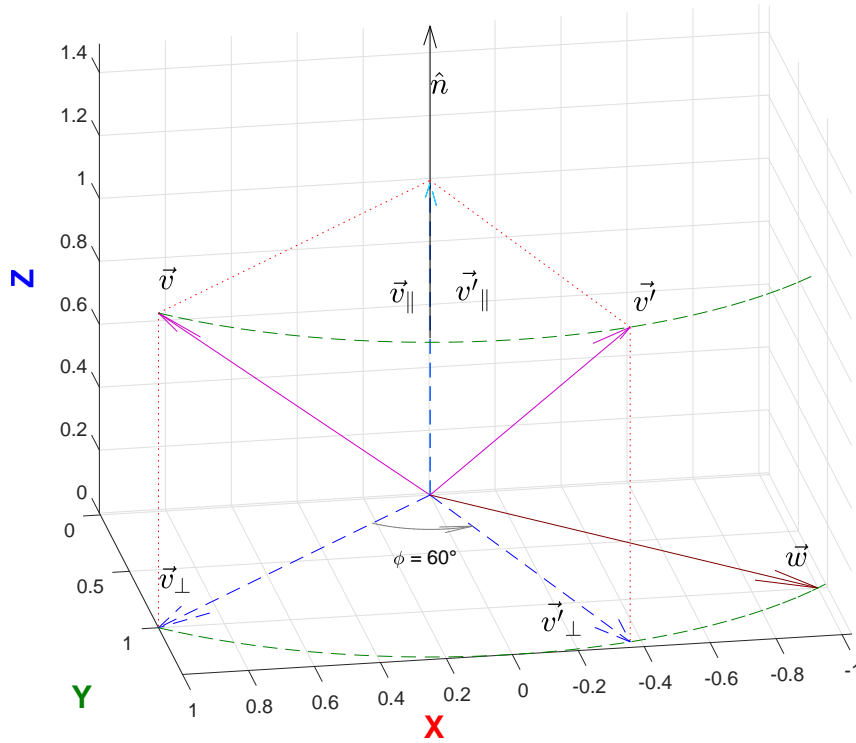


Figura 4.12: Rotação do vetor \vec{v} , decomposto em suas projeções ortogonal e perpendicular ao vetor \hat{n} , resultado no vetor \vec{v}' . Repare que as projeções paralelas são iguais antes e após a rotação ao redor de \hat{n} .

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 a \hat{n} são respectivamente \vec{v}_{\parallel} e \vec{v}_{\perp} dadas por

$$\vec{v}_{\parallel} = \langle \vec{v}, \hat{n} \rangle \cdot \hat{n} \quad (4.17)$$

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

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 será a adição de ' ao vetor rotacionado, ou seja

$$\vec{v}' = \vec{v}'_{\parallel} + \vec{v}'_{\perp} \quad (4.19)$$

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

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

Consequentemente, falta ser determinado 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).

$$\vec{w} = \hat{n} \times \vec{v}_{\perp} = \hat{n} \times (\vec{v} - \vec{v}_{\parallel}) = \hat{n} \times \vec{v} - \hat{n} \times \vec{v}_{\parallel} = \hat{n} \times \vec{v} \quad (4.21)$$

Assim, a decomposição da rotação do vetor perpendicular é

$$\vec{v}'_{\perp} = \cos(\phi) \vec{v}_{\perp} + \sin(\phi) \vec{w} \quad (4.22)$$

O resultado das rotações pode ser reescrito em função de \vec{v} , \hat{n} e ϕ

$$\vec{v}'_{\parallel} = \langle \vec{v}, \hat{n} \rangle \cdot \hat{n} \quad (4.23)$$

$$\vec{v}'_{\perp} = \cos(\phi) \vec{v}_{\perp} + \sin(\phi) (\hat{n} \times \vec{v}) \quad (4.24)$$

O resultado da rotação de \vec{v} é portanto

$$\vec{v}' = \vec{v}'_{\parallel} + \vec{v}'_{\perp} \quad (4.25)$$

$$= \langle \vec{v}, \hat{n} \rangle \cdot \hat{n} + \cos(\phi) \vec{v}_{\perp} + \sin(\phi) (\hat{n} \times \vec{v}) \quad (4.26)$$

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

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

Definição: $\overset{\times}{u} := (\vec{u} \times)$, é a matriz que produz o mesmo resultado de se aplicar o produto vetorial $(\vec{u} \times)$ ao vetor \vec{v} .²

$$\overset{\times}{u} = \begin{bmatrix} 0 & -u_3 & +u_2 \\ +u_3 & 0 & -u_1 \\ -u_2 & +u_1 & 0 \end{bmatrix} \quad (4.29)$$

Exercício 31 Para $\vec{u} = [u_1, u_2, u_3]^T$ e $\vec{v} = [v_1, v_2, v_3]^T$. *Mostre que*

$$\vec{u} \times \vec{v} = \begin{bmatrix} 0 & -u_3 & +u_2 \\ +u_3 & 0 & -u_1 \\ -u_2 & +u_1 & 0 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \overset{\times}{u} \cdot \vec{v} \quad (4.30)$$

Exercício 32 *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 33 *Mostre que $\langle \vec{v}, \hat{n} \rangle \hat{n} = \hat{n} \hat{n}^T \vec{v}$. Para isso, *Mostre que* $\langle \vec{v}, \vec{u} \rangle \vec{u} = \vec{u} \vec{u}^T \vec{v}$, $\forall \vec{u}, \vec{v} \in \mathbb{R}^3$.*

Retomando \vec{v}'

$$\vec{v}' = \cos(\phi) \vec{v} + (1 - \cos(\phi)) \langle \vec{v}, \hat{n} \rangle \cdot \hat{n} + \sin(\phi) \hat{n} \times \vec{v} \quad (4.31)$$

$$= \cos(\phi) \vec{v} + \hat{n} \hat{n}^T (1 - \cos(\phi)) \vec{v} + \sin(\phi) \overset{\times}{n} \vec{v} \quad (4.32)$$

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

$$\vec{v}' = \left(\cos(\phi) I + \hat{n} \hat{n}^T (1 - \cos(\phi)) + \sin(\phi) \overset{\times}{n} \right) \vec{v} = R \vec{v} \quad (4.33)$$

$$\Rightarrow R = \text{Rot}(\hat{n}, \phi) = \cos(\phi) I + \hat{n} \hat{n}^T (1 - \cos(\phi)) + \sin(\phi) \overset{\times}{n} \quad (4.34)$$

A matriz $\text{Rot}(\hat{n}, \phi)$ apresentada em (4.34) é conhecida como *Fórmula (de rotação) de Rodrigues*³.

²Alguns livros denotam essa matriz por \hat{u} e utilizam a nomenclatura de *tensor (antissimétrico do produto vetorial)*, e.g., (BÖGE et al., 2017).

³Existe uma *Fórmula de Rodrigues* para os polinômios de Legendre que é formalmente denominada *Fórmula de Ivory-Jacobi* (WIKIPÉDIA, 2019c).

Exemplo 9 *Exemplo da Fórmula de Rodrigues. Encontrar a matriz de rotação de qualquer vetor \vec{v} ao redor de \hat{n} e ϕ e depois aplicar para \vec{v} dados por*

$$\hat{n} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \quad \theta = \frac{\pi}{2}; \quad \vec{v} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.35)$$

para qualquer ângulo ϕ é apresentada a seguir.

A Fórmula de Rodrigues apresentada em (4.34) é $\text{Rot}(\hat{n}, \phi) = \cos(\phi)I + \hat{n} \hat{n}^\top (1 - \cos(\phi)) + \sin(\phi) \hat{n}^\times$ de onde tem-se

$$\begin{aligned} \hat{n} \cdot \hat{n}^\top &= \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} \cdot \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad \hat{n}^\times = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \\ R &= \begin{bmatrix} \cos(\phi) & 0 & 0 \\ 0 & \cos(\phi) & 0 \\ 0 & 0 & \cos(\phi) \end{bmatrix} + \begin{bmatrix} 1 - \cos(\phi) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\sin(\phi) \\ 0 & \sin(\phi) & 0 \end{bmatrix} \\ R &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} = R_x(\phi) \end{aligned} \quad (4.36)$$

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

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.37)$$

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

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

Código 21 *Faça uma função que retorne $\text{Rot}(\hat{n}, \phi)$ a partir da Fórmula de Rodrigues*

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

Exercício 34 *Aplique a Fórmula de Rodrigues para $\hat{n} = \hat{j} = \hat{e}_2$*

Exercício 35 *Aplique a Fórmula de Rodrigues para $\hat{n} = \hat{k} = \hat{e}_3$*

Exercício 36 *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 .

$$R = \text{Rot}(\hat{n}, \phi) = \begin{bmatrix} c - n_1^2(c-1) & -n_3 s - n_1 n_2(c-1) & n_2 s - n_1 n_3(c-1) \\ n_3 s - n_1 n_2(c-1) & c - n_2^2(c-1) & -n_1 s - n_2 n_3(c-1) \\ -n_2 s - n_1 n_3(c-1) & n_1 s - n_2 n_3(c-1) & c - n_3^2(c-1) \end{bmatrix} \quad (4.39)$$

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

O traço Tr da matriz de rotação

$$\begin{aligned} \text{Tr} &= \text{Trace}(R) = c + n_1^2(1-c) + c - n_2^2(c-1) + c - n_3^2(c-1) \\ &= 3c + (1-c) \underbrace{(n_1^2 + n_2^2 + n_3^2)}_{|\hat{n}|_2^2=1} = 3c + 1 - c = 2c + 1 \\ \Rightarrow c &= \cos(\phi) = \frac{\text{Tr} - 1}{2} \end{aligned} \quad (4.40)$$

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

$$A = R - R^\top = \begin{bmatrix} 0 & -2n_3s & 2n_2s \\ 2n_3s & 0 & -2n_1s \\ -2n_2s & 2n_1s & 0 \end{bmatrix} = 2s \cdot \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix} \quad (4.41)$$

$$R - R^\top = 2\sin(\phi) \hat{n}^\times$$

$$\Rightarrow \hat{n}^\times = \frac{1}{2\sin(\phi)} \cdot (R - R^\top)$$

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:

$$\begin{aligned} \text{Tr} = 3 &\Rightarrow \phi = 0 \Rightarrow \nexists \text{ único eixo} \\ \text{Tr} = 1 &\Rightarrow \phi = \pi \Rightarrow \hat{n} \text{ ou } -\hat{n} \end{aligned} \quad (4.42)$$

4.4.1 Entendendo o grupo de rotação $\text{SO}(3)$

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 $\text{SO}(3)$ 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

$$\text{Rot}(\hat{n}, \phi) = \text{Rot}(-\hat{n}, 2\pi - \phi) \quad (4.43)$$

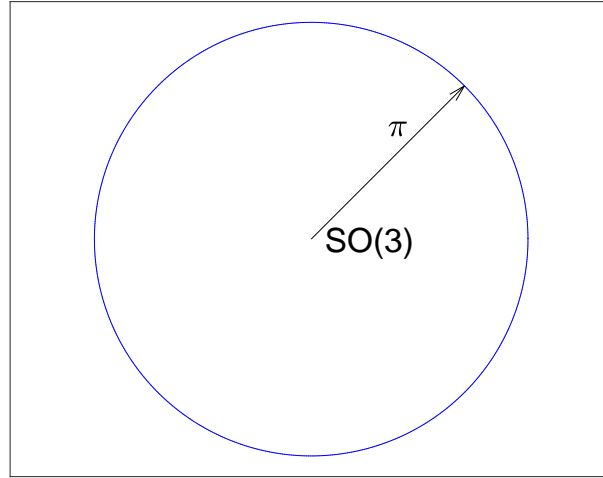


Figura 4.13: Representação do grupo $\text{SO}(3)$ como esfera de raio π .

4.5 Rotações 2D - Fórmula de Euler

Retomemos as rotações em 2 dimensões porém agora com uso da fórmula de Euler⁴ ([BIASI; GATTASS, 2002](#)).

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

$$z = r \cdot e^{i\theta}, \text{ com } \begin{cases} r^2 = x^2 + y^2 \\ \tan \theta = \frac{y}{x} \Rightarrow \theta = \text{atan2}(y, x) \end{cases} \quad (4.44)$$

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

⁴O 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 $e_{(\cdot)}$.

Exercício 37 *Mostre que $e^{i\theta} = \cos(\theta) + i \sin(\theta)$. Dica: Use a série de Taylor.*

Exercício 38 *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 39 *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}

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

$$\begin{aligned} q_1 q_2 &= (a_1, \vec{v}_1) (a_2, \vec{v}_2) \\ &= (a_1 a_2 - \langle \vec{v}_1, \vec{v}_2 \rangle, a_1 \vec{v}_2 + a_2 \vec{v}_1 + \vec{v}_1 \times \vec{v}_2) \end{aligned} \quad (4.45)$$

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

$$q_1 q_2 = (-\langle \vec{v}_1, \vec{v}_2 \rangle, \vec{v}_1 \times \vec{v}_2) \quad (4.46)$$

com base nesse resultado, calculemos agora q^2

$$q^2 = q q = (-\langle \vec{v}, \vec{v} \rangle, \vec{v} \times \vec{v}) = -\|\vec{v}\|^2 \quad (4.47)$$

repare que

$$q^2 \neq |q|^2 = a^2 + \|\vec{v}\|^2 \quad (4.48)$$

as demais potenciais podem ser calculadas com base nas anteriores

$$q^3 = q^2 q = -\|\vec{v}\|^2 q = -\|\vec{v}\|^2 (0, \vec{v}) = -\|\vec{v}\|^2 \cdot \vec{v} \quad (4.49)$$

$$q^4 = q^2 q^2 = (-\|\vec{v}\|^2) \cdot (-\|\vec{v}\|^2) = \|\vec{v}\|^4 \quad (4.50)$$

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 $k \in \mathbb{N}$, $q^n = q^{2k}$ é sempre um escalar e $q^n = q^{2k+1}$ é sempre um escalar vezes o 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

$$\begin{aligned} e^q &= 1 + q + \frac{q^2}{2} + \frac{q^3}{3!} + \frac{q^4}{4!} + \dots \\ &= 1 + \vec{v} - \frac{\|\vec{v}\|^2}{2} - \frac{\|\vec{v}\|^2}{3!} \vec{v} + \frac{\|\vec{v}\|^4}{4!} + \frac{\|\vec{v}\|^4}{5!} \vec{v} - \frac{\|\vec{v}\|^6}{6!} - \frac{\|\vec{v}\|^6}{7!} \vec{v} + \dots \\ &= \left(1 - \frac{\|\vec{v}\|^2}{2} + \frac{\|\vec{v}\|^4}{4!} - \frac{\|\vec{v}\|^6}{6!} + \dots\right) + \left(1 - \frac{\|\vec{v}\|^2}{3!} + \frac{\|\vec{v}\|^4}{5!} - \frac{\|\vec{v}\|^6}{7!} + \dots\right) \vec{v} \\ &= \cos(\|\vec{v}\|) + \frac{\|\vec{v}\|}{\|\vec{v}\|} \left(1 - \frac{\|\vec{v}\|^2}{3!} + \frac{\|\vec{v}\|^4}{5!} - \frac{\|\vec{v}\|^6}{7!} + \dots\right) \vec{v} \\ &= \cos(\|\vec{v}\|) + \left(\|\vec{v}\| - \frac{\|\vec{v}\|^3}{3!} + \frac{\|\vec{v}\|^5}{5!} - \frac{\|\vec{v}\|^7}{7!} + \dots\right) \frac{\vec{v}}{\|\vec{v}\|} \\ &= \cos(\|\vec{v}\|) + \sin(\|\vec{v}\|) \frac{\vec{v}}{\|\vec{v}\|} \\ &= \cos(\|\vec{v}\|) + \sin(\|\vec{v}\|) \hat{v} \end{aligned} \quad (4.51)$$

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

$$e^q = e^{xi+yj+zk} = \cos(\|\vec{v}\|) + \frac{\text{sen}(\|\vec{v}\|)}{\|\vec{v}\|} \cdot (xi + yj + zk) \in \mathbb{H} \quad (4.52)$$

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

$$e^{a+bi} = e^a \cdot e^{bi} = e^a \cdot (\cos(b) + i \text{sen}(b)) \quad (4.53)$$

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 $\|\vec{v}\| = \sqrt{x^2 + y^2 + z^2}$

$$e^q = e^{a+xi+yj+zk} = e^a \cdot e^{xi+yj+zk} = e^a \cdot \left(\cos(\|\vec{v}\|) + \frac{\text{sen}(\|\vec{v}\|)}{\|\vec{v}\|} \vec{v} \right) \quad (4.54)$$

Exemplo 11 Aplicações da exponencial

(a)

$$e^{i+j} = \cos(\sqrt{2}) + \frac{\text{sen}(\sqrt{2})}{\sqrt{2}} (i + j) \approx 0,156 + 0,698(i + j) \quad (4.55)$$

(b)

$$e^{3j+4k} = \cos(5) + \frac{\text{sen}(5)}{5} (3j + 4k) \approx 0,284 - 0,575j - 0,767k \quad (4.56)$$

(c)

$$e^{ix} = \cos(x) + \frac{\text{sen}(x)}{x} (ix) = \cos(x) + i \text{sen}(x) \quad (4.57)$$

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

Código 22 Crie uma função que retorne a exponencial de um quatérnio.

`r = fc_quat_exp(q).`

Exercício 40 Demonstre que $e^{a+xi+yj+zk} = e^a \cdot e^{xi+yj+zk}$ através da série de potências.

Exercício 41 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 42 (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 $\vec{v} = \|\vec{v}\| \hat{n} = \|q\| \text{sen}(\theta) \hat{n}$.

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

$$q^k = |q|^k \cdot e^{k\theta \hat{n}} = |q|^k \cdot (\cos(k\theta) + \text{sen}(k\theta) \cdot \hat{n}) \quad (4.58)$$

(f) Calcule $q^{1/2}$ 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{u} e ângulo ϕ . Essa seção é baseada principalmente em (MATHOMA, 2016).

Aqui usaremos a notação de \hat{n} como vetor unitário perpendicular a \vec{v} , \hat{n} é um vetor normal.

Como consequência da Equação (4.22) tem-se que a decomposição do vetor rotacionado \vec{v}' em função de \vec{v} e da projeção ortogonal $\hat{n} \times \vec{v}$

$$\vec{v}' = \cos(\phi) \vec{v} + \text{sen}(\phi) (\hat{n} \times \vec{v}) \quad (4.59)$$

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

$$pq = (0, \hat{n}) \cdot (0, \vec{v}) = (0 - \langle \hat{n}, \vec{v} \rangle, 0\vec{v} + 0\hat{n} + \hat{n} \times \vec{v}) = (-\langle \hat{n}, \vec{v} \rangle, \hat{n} \times \vec{v}) \quad (4.60)$$

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

$$\hat{n} \perp \vec{v} \Rightarrow \langle \hat{n}, \vec{v} \rangle = 0 \Rightarrow pq = (0, \hat{n} \times \vec{v}) \quad (4.61)$$

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

$$\begin{aligned} \vec{v}' &= \cos(\phi)\vec{v} + \sin(\phi)(\hat{n} \times \vec{v}) \\ &= \cos(\phi)q + \sin(\phi)pq \\ &= (\cos(\phi) + \sin(\phi)p)q \\ &= \left(\cos(\phi) + \sin(\phi)\frac{\vec{n}}{\|\vec{n}\|} \right) q; \quad \phi = \|\vec{n}\| \\ r &= (e^{\phi\hat{n}})q \end{aligned} \quad (4.62)$$

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

$$\hat{n} = \frac{\vec{n}}{\|\vec{n}\|} \Rightarrow \vec{n} = \|\vec{n}\|\hat{n} = \phi\hat{n} \quad (4.63)$$

então é possível determinar que

$$e^{\phi\hat{n}} = e^{\vec{n}} = e^{(0, \vec{n})} = e^s = p \quad (4.64)$$

e com base no resultado de (4.52), define-se $s = (0, \vec{n}) \in \mathbb{H}$.

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

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

$$q = (0, \hat{e}_3) = (0, [0, 0, 1]^\top) = k, \quad \vec{v} = k = \hat{e}_3 \quad (4.65)$$

$$p = e^{\phi\hat{n}}, \quad \hat{n} = j = \hat{e}_2, \quad \phi = \frac{\pi}{2} \quad (4.66)$$

O resultado da rotação utilizando a exponencial de quatérnio é

$$\begin{aligned} r &= e^{\phi\hat{n}}q \\ &= (\cos(\phi) + \sin(\phi)\hat{n})q \\ &= \left(\cos\left(\frac{\pi}{2}\right) + \sin\left(\frac{\pi}{2}\right)\hat{n} \right) q \\ &= (1j) \cdot k = j \cdot k \\ &= \hat{e}_2 \cdot \hat{e}_3 = \hat{e}_1 = i \\ \therefore r &= \hat{e}'_3 = \hat{e}_1 \end{aligned} \quad (4.67)$$

O resultado é apresentado na [Figura 4.14](#).

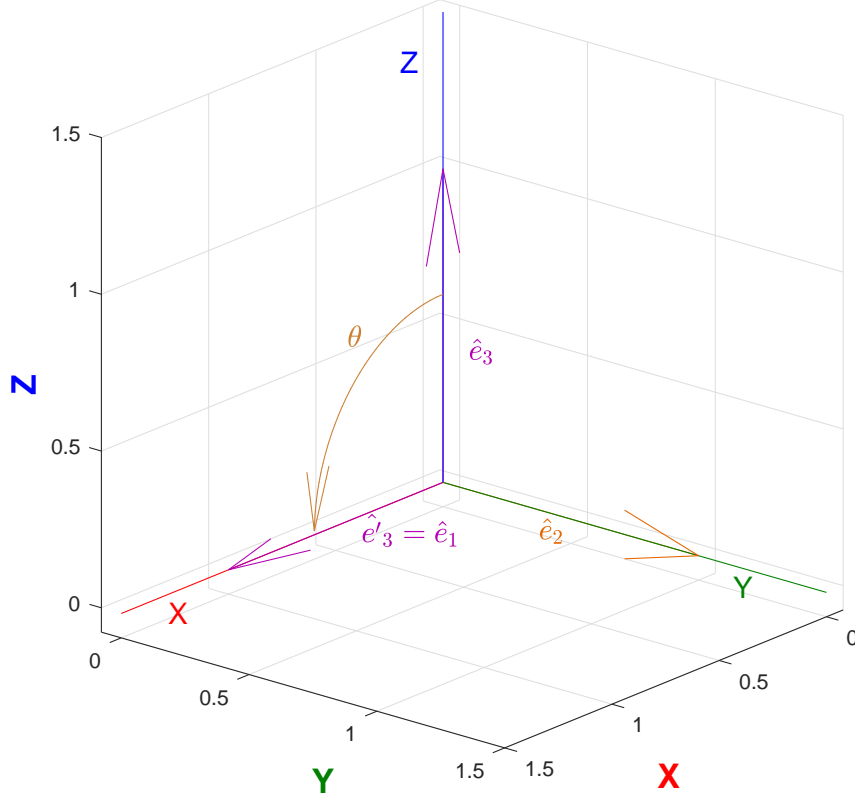


Figura 4.14: Resultado da rotação de \hat{e}_3 ao redor de \hat{e}_2 por um ângulo de $\frac{\pi}{2}$ rad.

Exercício 43 *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 $\hat{n} \perp \vec{v}$*

$$e^{\phi p} q = q e^{-\phi p} \quad (4.68)$$

Exercício 44 *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 $\hat{n} \parallel \vec{u}$*

$$e^{\theta p} q = q e^{\theta p} \quad (4.69)$$

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 rotaciona

$$\begin{aligned} \vec{v} &= \vec{v}_{\parallel} + \vec{v}_{\perp} \\ \vec{v}' &= \vec{v}'_{\parallel} + \vec{v}'_{\perp} \end{aligned} \quad (4.70)$$

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

$$\vec{v}'_{\parallel} = \vec{v}_{\parallel} \quad (4.71)$$

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

$$r = pq = e^{\phi \hat{n}} \cdot q = e^{\phi \hat{n}} \cdot (0, \vec{v}) = e^{\phi \hat{n}} \cdot (0, \vec{v}_{\parallel} + \vec{v}_{\perp}) \quad (4.72)$$

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

$$q = (0, \vec{v}), \quad q_{\parallel} = (0, \vec{v}_{\parallel}), \quad q_{\perp} = (0, \vec{v}_{\perp}) \quad \Rightarrow q = q_{\parallel} + q_{\perp} \quad (4.73)$$

$$r = (0, \vec{v}'), \quad r_{\parallel} = (0, \vec{v}'_{\parallel}), \quad r_{\perp} = (0, \vec{v}'_{\perp}) \quad \Rightarrow r = r_{\parallel} + r_{\perp} \quad (4.74)$$

com base nos resultados anteriores, tem-se que

$$r = r_{\parallel} + r_{\perp} = q_{\parallel} + e^{\phi \hat{n}} q_{\perp} = q_{\parallel} + \cos(\phi) q_{\perp} + \sin(\phi) p q_{\perp} \quad (4.75)$$

que é a *Fórmula de Rodrigues*.

Por notação, sejam $c_\phi = \cos(\phi)$ e $s_\phi = \sin(\phi)$. O resultado de (4.75) é compactado como

$$r = q_{\parallel} + c_\phi q_{\perp} + s_\phi p q_{\perp} \quad (4.76)$$

O primeiro passo para obtermos a *Fórmula de Rodrigues* da (4.34) é analisar o termo $p q_{\perp}$

$$\begin{aligned} p q_{\perp} &= p(q - q_{\parallel}) = p q - p q_{\parallel} \\ p q &= (0, \hat{n}) \cdot (0, \vec{v}) = \hat{n} \times \vec{v} = \hat{n} \cdot \vec{v} \\ p q_{\parallel} &= (0, \hat{n}) \cdot (0, \vec{v}_{\parallel}) = \hat{n} \times \vec{v}_{\parallel} = \mathbf{0} \\ \Rightarrow s_\phi p q_{\perp} &= s_\phi \cdot p q = s_\phi \cdot \hat{n} \cdot \vec{v} \end{aligned} \quad (4.77)$$

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

$$q_{\parallel} = (0, \vec{v}_{\parallel}) = (0, \langle \hat{n}, \vec{v} \rangle \hat{n}) = (0, \vec{v}_{\parallel}) = (0, \hat{n} \langle \hat{n}, \vec{v} \rangle) = (0, \vec{v}_{\parallel}) = (0, \hat{n} \hat{n}^T \vec{v}) \quad (4.78)$$

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

$$\begin{aligned} r &= p q = e^{\phi \hat{n}} q = q_{\parallel} + (c_\phi + s_\phi p) q_{\perp} \\ (0, \vec{v}) &= \left(0, \left(c_\phi I + (1 - c_\phi) \hat{n} \hat{n}^T + s_\phi \hat{n} \times \right) \vec{v} \right) \\ &= (0, \text{Rot}(\hat{n}, \phi) \cdot \vec{v}) \end{aligned} \quad (4.79)$$

A *Fórmula de Rodrigues* ainda pode ser escrita em uma **forma compacta** com quatérnios. Retomando a partir de (4.75) tem-se

$$\begin{aligned} r &= q_{\parallel} + e^{\phi \hat{n}} q_{\perp} \\ &= e^{\frac{\phi}{2} \hat{n}} \underbrace{e^{-\frac{\phi}{2} \hat{n}} q_{\parallel}}_{\text{comutativo}} + e^{\frac{\phi}{2} \hat{n}} \underbrace{e^{\frac{\phi}{2} \hat{n}} q_{\perp}}_{\text{anticomutativo}} \end{aligned} \quad (4.80)$$

com o resultado do Exercício 43 para o caso anticomutativo e do Exercício 44 para o caso comutativo tem-se

$$\begin{aligned} r &= e^{\frac{\phi}{2} \hat{n}} q_{\parallel} e^{-\frac{\phi}{2} \hat{n}} + e^{\frac{\phi}{2} \hat{n}} q_{\perp} e^{-\frac{\phi}{2} \hat{n}} \\ &= e^{\frac{\phi}{2} \hat{n}} (q_{\parallel} + q_{\perp}) e^{-\frac{\phi}{2} \hat{n}} \\ &= e^{\frac{\phi}{2} \hat{n}} q e^{-\frac{\phi}{2} \hat{n}} \end{aligned} \quad (4.81)$$

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

$$\begin{aligned} s &= e^{\frac{\phi}{2} \hat{n}} = \cos\left(\frac{\phi}{2}\right) + \sin\left(\frac{\phi}{2}\right) \hat{n} \\ s^* &= e^{-\frac{\phi}{2} \hat{n}} = \cos\left(\frac{\phi}{2}\right) - \sin\left(\frac{\phi}{2}\right) \hat{n} \\ \Rightarrow r &= s q s^* \end{aligned} \quad (4.82)$$

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 23 Crie uma função que implementa a rotação entre quatérnios dada por $r = s q s^*$.

`r = fc_quat_rotacao(q, s)`

Código 24 Crie uma função que implementa construa a rotação de um vetor \vec{v} ao redor de um vetor unitário \hat{n} por um ângulo θ .

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

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

Defina $\hat{n} = \frac{\vec{u}}{\|\vec{u}\|} = \frac{\vec{u}}{\sqrt{3}} = \frac{[1, 1, 1]^\top}{\sqrt{3}}$ 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

$$\begin{aligned} \vec{v}' &= c_\phi \vec{v} + s_\phi (\hat{n} \times \vec{v}) \\ &= \cos\left(\frac{\pi}{3}\right) \cdot \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} + \frac{\sin(\pi/3)}{\sqrt{3}} \cdot \begin{bmatrix} n_2 v_3^0 - n_3 v_2 \\ n_3 v_1 - n_1 v_3^0 \\ n_1 v_2 - n_2 v_1 \end{bmatrix} \\ &= \left(\frac{1}{2}\right) \cdot \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} + \frac{\sqrt{3}}{2} \cdot \frac{1}{\sqrt{3}} \cdot \begin{bmatrix} -v_2 \\ v_1 \\ v_2 - v_1 \end{bmatrix} \\ &= \left(\frac{1}{2}\right) \cdot \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} + \left(\frac{1}{2}\right) \cdot \begin{bmatrix} 1 \\ 1 \\ -1 - 1 \end{bmatrix} = \left(\frac{1}{2}\right) \cdot \begin{bmatrix} 2 \\ 0 \\ -2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \end{aligned} \quad (4.83)$$

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

$$q = (0, [1, -1, 0]^\top) \quad (4.84)$$

$$\hat{n} = \frac{\vec{u}}{\|\vec{u}\|} = \frac{\vec{u}}{\sqrt{3}} = \frac{[1, 1, 1]^\top}{\sqrt{3}} \quad (4.85)$$

$$s = \cos(\phi) + \sin(\phi)\hat{n} = \left(\sqrt{\frac{3}{2}}, \frac{1}{2} \cdot \frac{[1, 1, 1]^\top}{\sqrt{3}}\right) \quad (4.86)$$

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

$$r = sqs^* = (0, [1, 1547; -0,5774; -0,5774]^\top) \cdot s^* = (0, [1, 0, -1]^\top) \Rightarrow \vec{v}' = [1, 0, -1]^\top \quad (4.87)$$

O resultado concorda com o obtido no Exemplo 13.

Exercício 45 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 46 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 6

Solução:

(a)

$$\begin{aligned}\vec{v} \times \vec{u} &= (v_2u_3 - v_3u_2)e_1 + (v_3u_1 - v_1u_3)e_2 + (v_1u_2 - v_2u_1)e_3 \\ \vec{u} \times \vec{v} &= (u_2v_3 - u_3v_2)e_1 + (u_3v_1 - u_1v_3)e_2 + (u_1v_2 - u_2v_1)e_3 \\ \vec{u} \times \vec{v} &= -(v_2u_3 - v_3u_2)e_1 - (v_3u_1 - v_1u_3)e_2 - (v_1u_2 - v_2u_1)e_3 \\ &= -\vec{v} \times \vec{u}\end{aligned}\tag{A.1}$$

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

$$q = (a, \vec{v}), p = (b, \vec{u}) \in \mathbb{H}\tag{A.2}$$

Os produtos de quatérnios qp e pq são

$$\begin{aligned}qp &= (ab - \langle \vec{v}, \vec{u} \rangle, a\vec{u} + b\vec{v} + \vec{v} \times \vec{u}) \\ pq &= (ba - \langle \vec{u}, \vec{v} \rangle, b\vec{v} + a\vec{u} + \vec{u} \times \vec{v}) \\ &= (ab - \langle \vec{v}, \vec{u} \rangle, a\vec{u} + b\vec{v} - \vec{v} \times \vec{u})\end{aligned}\tag{A.3}$$

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

$$\vec{v} \times \vec{u} = -\vec{v} \times \vec{u} \Rightarrow \vec{v} \times \vec{u} = \mathbf{0}\tag{A.4}$$

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

$$\begin{aligned}\vec{u} &= c\vec{v}, \quad \forall c \in \mathbb{R} \\ \vec{v} \times \vec{u} &= \vec{v} \times (c\vec{v}) = c\vec{v} \times \vec{v} = c\mathbf{0} = \mathbf{0}\end{aligned}\tag{A.5}$$

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

$$qp = (ab - \langle \vec{v}, \vec{u} \rangle, a\vec{u} + b\vec{v}) = pq\tag{A.6}$$

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

$$qp = (ab - c\langle \vec{v}, \vec{v} \rangle, ac\vec{v} + b\vec{v}) = (ab - c\|\vec{v}\|^2, (ac + b)\vec{v}) = pq\tag{A.7}$$

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

$$q = a + xi + yj + zk, \quad p = b + c(xi + yj + zk)\tag{A.8}$$

que em valores numéricos pode ser exemplificado

$$q = 1 + 2i + 5j + 7k, \quad p = 7 + 20i + 50j + 70k \Rightarrow qp = -773 + 34i + 85j + 119k = pq\tag{A.9}$$

(d)

$$q = 1 + 2i + 5j + 7k, \quad p = b + c(2i + 5j + 7k), \quad b, c \in \mathbb{R}\tag{A.10}$$

A.2 Exercício 8

Solução: Partindo da solução do caso em que dois quatérnios comutam apresentado em (A.7) 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

$$p = \bar{q} = (a, -\vec{v}) \Rightarrow b = a, \quad c = -1 \quad (\text{A.11})$$

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.3 Exercício 9

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

$$q\bar{q} = (aa - c\|\vec{v}\|^2, (ac + b)\vec{v}) = (aa - (-1)\|\vec{v}\|^2, (a(-1) + b)\vec{v}) = (a^2 + \|\vec{v}\|^2, \mathbf{0}) = |q|^2 \in \mathbb{R}^+ \quad (\text{A.12})$$

A.4 Exercícios 22, 23, 24, 25

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_save_file_extensao`. `fc_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.

$$R_{xyz}(\phi, \psi, \theta) = \begin{bmatrix} \cos(\psi) \cos(\theta) & -\sin(\psi) & \cos(\psi) \sin(\theta) \\ \sin(\phi) \sin(\theta) + \cos(\phi) \cos(\theta) \sin(\psi) & \cos(\phi) \cos(\psi) & \cos(\phi) \sin(\psi) \sin(\theta) - \cos(\theta) \sin(\phi) \\ \cos(\theta) \sin(\phi) \sin(\psi) - \cos(\phi) \sin(\theta) & \cos(\psi) \sin(\phi) & \cos(\phi) \cos(\theta) + \sin(\phi) \sin(\psi) \sin(\theta) \end{bmatrix} \quad (\text{A.13})$$

$$R_{yzx}(\psi, \phi, \theta) = \begin{bmatrix} \cos(\psi) \cos(\theta) + \sin(\phi) \sin(\psi) \sin(\theta) & \cos(\psi) \sin(\phi) \sin(\theta) - \cos(\theta) \sin(\psi) & \cos(\phi) \sin(\theta) \\ \cos(\phi) \sin(\psi) & \cos(\phi) \cos(\psi) & -\sin(\phi) \\ \cos(\theta) \sin(\phi) \sin(\psi) - \cos(\psi) \sin(\theta) & \sin(\psi) \sin(\theta) + \cos(\psi) \cos(\theta) \sin(\phi) & \cos(\phi) \cos(\theta) \end{bmatrix} \quad (\text{A.14})$$

$$R_{zyx}(\psi, \phi, \theta) = \begin{bmatrix} \cos(\psi) \cos(\theta) & \cos(\psi) \sin(\phi) \sin(\theta) - \cos(\phi) \sin(\psi) & \sin(\phi) \sin(\psi) + \cos(\phi) \cos(\psi) \sin(\theta) \\ \cos(\theta) \sin(\psi) & \cos(\phi) \cos(\psi) + \sin(\phi) \sin(\psi) \sin(\theta) & \cos(\phi) \sin(\psi) \sin(\theta) - \cos(\psi) \sin(\phi) \\ -\sin(\theta) & \cos(\theta) \sin(\phi) & \cos(\phi) \cos(\theta) \end{bmatrix} \quad (\text{A.15})$$

$$R_{zyz}(\psi, \phi, \theta) = \begin{bmatrix} \cos(\phi) \cos(\psi) \cos(\theta) - \sin(\phi) \sin(\psi) & -\cos(\phi) \sin(\psi) - \cos(\psi) \cos(\theta) \sin(\phi) & \cos(\psi) \sin(\theta) \\ \cos(\psi) \sin(\phi) + \cos(\phi) \cos(\theta) \sin(\psi) & \cos(\phi) \cos(\psi) - \cos(\theta) \sin(\phi) \sin(\psi) & \sin(\phi) \sin(\theta) \\ -\cos(\phi) \sin(\theta) & \sin(\phi) \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (\text{A.16})$$

A.4.1 Código de resolução com código substituidor de expressões e código gerador de arquivos a partir de uma *string*

Algoritmo A.1: Resolução dos Exercícios 24, 23, 24 e 25 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);
```

¹help latex

```

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 - XYZ
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_XYZ', '.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.Rz(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.5 Exercício 30

Solução:

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

Problema de autovalor:

$$R\vec{p} = \lambda\vec{p}, \quad R \in M(3, \mathbb{R}), \lambda \in \mathbb{R} \quad (\text{A.17})$$

$$\Rightarrow \vec{p}(1 - \lambda) = \mathbf{0} \quad (\text{A.18})$$

$$\vec{p} \neq \mathbf{0} \Rightarrow \lambda = 1 \quad (\text{A.19})$$

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

A.6 Exercício 33

Solução:

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

$$\langle \vec{v}, \vec{u} \rangle \cdot \vec{u} = \underbrace{(v_1 u_1 + v_2 u_2 + v_3 u_3)}_C \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} C, \quad C \in \mathbb{R} \quad \langle \vec{v}, \vec{u} \rangle \cdot \vec{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (\text{A.20})$$

Por outro lado

$$\vec{u} \cdot \vec{u}^\top \cdot \vec{v} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \underbrace{(v_1 u_1 + v_2 u_2 + v_3 u_3)}_C \quad (\text{A.21})$$

de onde se concluir que

$$\langle \vec{v}, \vec{u} \rangle \vec{u} = \vec{u} \vec{u}^\top \vec{v} \quad (\text{A.22})$$

Vale destacar que o produto interno é comutativo

$$\langle \vec{v}, \vec{u} \rangle = \langle \vec{u}, \vec{v} \rangle = \vec{u}^\top \cdot \vec{v} = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (\text{A.23})$$

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.

$$\langle \vec{u}, \vec{u} \rangle = \vec{u}^\top \cdot \vec{u} = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = u_1^2 + u_2^2 + u_3^2 \in \mathbb{R} \quad (\text{A.24})$$

$$\vec{u} \cdot \vec{u}^\top = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} = \begin{bmatrix} u_1 u_1 & u_1 u_2 & u_1 u_3 \\ u_2 u_1 & u_2 u_2 & u_2 u_3 \\ u_3 u_1 & u_3 u_2 & u_3 u_3 \end{bmatrix} \in \mathbb{R}^{3 \times 3} \quad (\text{A.25})$$

A.7 Exercício 43

Solução:

$$e^{\phi p} q = q e^{-\phi p} \quad (\text{A.26})$$

$$(\cos(\phi), \sin(\phi)\hat{n})(0, \vec{v}) = (0, \vec{v})(\cos(\phi), \sin(\phi)\hat{n}) \quad (\text{A.27})$$

$$(0 - \sin(\phi) \langle \hat{n}, \vec{v} \rangle, \cos(\phi)\vec{v} + \sin(\phi)\hat{n} \times \vec{v}) = (0 - \sin(\phi) \langle \vec{v}, \hat{n} \rangle, \cos(\phi)\vec{v} - \sin(\phi)\vec{v} \times \hat{n}) \quad (\text{A.28})$$

$$\hat{n} \perp \vec{v} \Rightarrow \langle \hat{n}, \vec{v} \rangle = \langle \vec{v}, \hat{n} \rangle = 0; \quad \vec{v} \times \hat{n} = -\hat{n} \times \vec{v} \text{ antissimétrico} \quad (\text{A.29})$$

$$\Rightarrow (0, \cos(\phi)\vec{v} + \sin(\phi)\hat{n} \times \vec{v}) = (0, \cos(\phi)\vec{v} + \sin(\phi)\hat{n} \times \vec{v}) \quad (\text{A.30})$$

A.8 Exercício 44

Solução:

$$e^{\theta p} q = q e^{\theta p} \quad (\text{A.31})$$

$$(\cos(\theta), \sin(\theta)\hat{n})(0, \vec{u}) = (0, \vec{u})(\cos(\theta), \sin(\theta)\hat{n}) \quad (\text{A.32})$$

$$(-\sin(\theta) \langle \hat{n}, \vec{u} \rangle, \cos(\theta)\vec{u} + \sin(\theta)\hat{n} \times \vec{u}) = (-\sin(\theta) \langle \hat{n}, \vec{u} \rangle, \cos(\theta)\vec{u} - \sin(\theta)\vec{u} \times \hat{n}) \quad (\text{A.33})$$

$$\langle \hat{n}, \vec{v} \rangle = \langle \vec{v}, \hat{n} \rangle; \quad \vec{u} \parallel \hat{n} \Rightarrow \vec{u} \times \hat{n} = \hat{n} \times \vec{u} = \mathbf{0} \quad (\text{A.34})$$

$$\Rightarrow (-\sin(\theta) \langle \hat{n}, \vec{u} \rangle, \cos(\theta)\vec{u}) = (-\sin(\theta) \langle \hat{n}, \vec{u} \rangle, \cos(\theta)\vec{u}) \quad (\text{A.35})$$

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 .

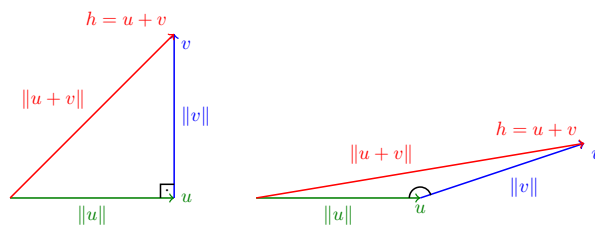


Figura B.1: Desigualdade triangular.

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

$$\begin{aligned} \|v\|_1 &= \sum_{i=1}^n (|v_i|) \\ \|v\|_2 &= \sum_{i=1}^n (|v_i|^2)^{\frac{1}{2}} \\ \|v\|_p &= \sum_{i=1}^n (|v_i|^p)^{\frac{1}{p}}, \quad 1 \leq p < \infty \\ \|v\|_\infty &= \max_{i=1}^n (|v_i|) \end{aligned} \quad (\text{B.4})$$

¹Normas em espaços de dimensão finita e infinita.

Relação entre as normas

$$\|v\|_\infty \leq \|v\|_2 \leq \|v\|_1 \leq \sqrt{n}\|v\|_2 \leq n\|v\|_\infty \quad (\text{B.5})$$

As principais Normas para vetores 1, 2, e ∞ , norma p .

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

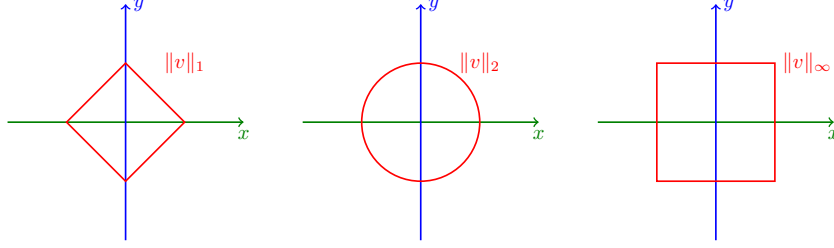


Figura B.2: Ilustração dos casos com $\|\cdot\|_p = 1$ para as normas $p = \{1, 2, \infty\}$.

Fonte: (WIKIPÉDIA, 2018d).

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:

$$\begin{aligned} \|x + y\|_1 &= |x| + |y| = 1 \\ x > 0, y > 0 : x + y &= 1 \Rightarrow y = 1 - x \\ x > 0, y < 0 : x - y &= 1 \Rightarrow y = x - 1 \\ x < 0, y > 0 : -x + y &= 1 \Rightarrow y = 1 + x \\ x < 0, y < 0 : -x - y &= 1 \Rightarrow y = -1 - x \end{aligned} \quad (\text{B.6})$$

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

$$\|x + y\|_2 = \sqrt{x^2 + y^2} = 1 \Rightarrow x^2 + y^2 = 1 \quad (\text{B.7})$$

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

$$\begin{aligned} \|x + y\|_\infty &= \max\{|x|, |y|\} = 1 \\ x > 0, y > 0 : \max\{|x|, |y|\} &= 1 \Rightarrow x = 1 \text{ ou } y = 1 \\ x > 0, y < 0 : \max\{|x|, |y|\} &= 1 \Rightarrow x = 1 \text{ ou } y = -1 \\ x < 0, y > 0 : \max\{|x|, |y|\} &= 1 \Rightarrow x = -1 \text{ ou } y = 1 \\ x < 0, y < 0 : \max\{|x|, |y|\} &= 1 \Rightarrow x = -1 \text{ ou } y = -1 \end{aligned} \quad (\text{B.8})$$

Definição de produto interno

Seja o produto interno uma operação que leva dois vetores de um espaço vetorial \mathbb{V} em um número real não-negativo \mathbb{R}^+ .

$$\begin{aligned} \langle \cdot, \cdot \rangle : \mathbb{V} \times \mathbb{V} &\rightarrow \mathbb{R}^+ \\ \langle v, u \rangle = 0 &\Leftrightarrow v = \mathbf{0} \text{ ou } u = \mathbf{0} \end{aligned} \quad (\text{B.9})$$

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

$$\langle v, u \rangle = v^\top M u = v^\top M u^\top = u^\top v^\top M^\top = u^\top M^\top (v^\top)^\top = u^\top M v = \langle u, v \rangle \quad (\text{B.10})$$

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

$$\langle v, u \rangle = v^\top u = \sum_{i=1}^n v_i u_i = \sum_{i=1}^n u_i v_i = u^\top v = \langle u, v \rangle \quad (\text{B.11})$$

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

$$\langle v, u \rangle = v^H u = \sum_{i=1}^n \bar{v}_i u_i = \sum_{i=1}^n \bar{u}_i v_i = u^H v = \langle u, v \rangle \quad (\text{B.12})$$

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

$$A = A^H = \overline{(A^T)} = (\bar{A})^T \quad (\text{B.13})$$

Norma induzida por um produto interno com o espaço vetorial. Para o caso de cada entrada no operador produto interno ser a mesma tem-se o seguinte resultado.

$$\|v\|^2 = \langle v, v \rangle = 0 \Leftrightarrow v = \mathbf{0} \quad (\text{B.14})$$

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

$$z_1^T \cdot z_1 = \begin{bmatrix} a, & bi \end{bmatrix} \cdot \begin{bmatrix} a \\ bi \end{bmatrix} = a^2 - b^2 \neq |z_1|^2 \quad (\text{B.15})$$

$$z_1^H \cdot z_1 = \begin{bmatrix} a, & -bi \end{bmatrix} \cdot \begin{bmatrix} a \\ bi \end{bmatrix} = a^2 + b^2 = |z_1|^2 \in \mathbb{R}^* \quad (\text{B.16})$$

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

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

$$\langle f, g \rangle = \int_{-\infty}^{\infty} f(x) \cdot g(x) dx \quad (\text{B.17})$$

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

$$\langle f, g \rangle = \int_{-\infty}^{\infty} \overline{f(x)} \cdot g(x) dx \quad (\text{B.18})$$

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

A definição apresentada na Equação (B.18) é a definição do espaço $L^2(D)$ ([WIKIPÉDIA, 2018c](#)), o qual é um espaço de Hilbert ([WIKIPÉDIA, 2018b](#)) dotado do produto interno apresentado.

Repare que para funções vetoriais é necessário acrescentar a transposição.

A [Tabela B.1](#) apresenta a nomenclatura normalmente empregada na classificação de tensores.

Tabela B.1: Nomes dos elementos tensoriais

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

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.

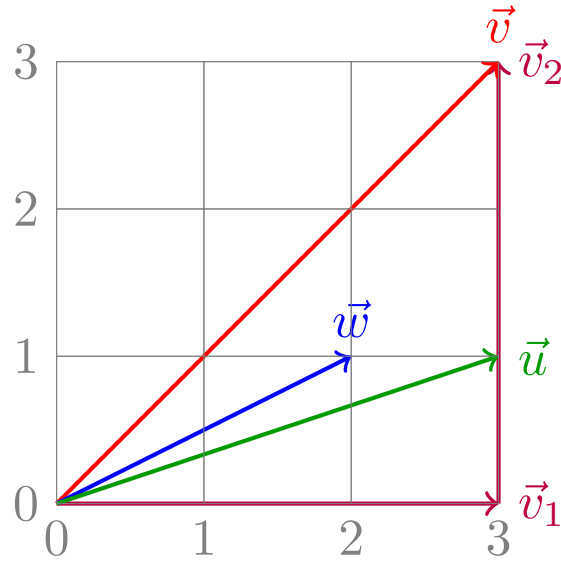


Figura B.3: Exemplo de normas.

Exemplo do uso das principais normas:

$$\begin{aligned}\|\vec{v}\|_1 &= |\vec{v}_1| + |\vec{v}_2| = 3 + 3 = 6 \\ \|\vec{v}\|_2 &= \sqrt{|\vec{v}_1|^2 + |\vec{v}_2|^2} = \sqrt{9 + 9} = \sqrt{18} \approx 4,2426\end{aligned}\tag{B.19}$$

$$\begin{aligned}\|\vec{v}\|_\infty &= \max\{|\vec{v}_1|, |\vec{v}_2|\} = \max\{3, 3\} = 3 \\ \|\vec{u}\|_1 &= |3| + |1| = 3 + 1 = 4 \\ \|\vec{u}\|_2 &= \sqrt{3^2 + 1^2} = \sqrt{9 + 1} = \sqrt{10} \approx 3,1622\end{aligned}\tag{B.20}$$

$$\begin{aligned}\|\vec{u}\|_\infty &= \max\{3, 1\} = 3 \\ \|\vec{w}\|_1 &= |2| + |1| = 2 + 1 = 3 \\ \|\vec{w}\|_2 &= \sqrt{2^2 + 1^2} = \sqrt{4 + 1} = \sqrt{5} \approx 2,2361 \\ \|\vec{w}\|_\infty &= \max\{2, 1\} = 2\end{aligned}\tag{B.21}$$

Apêndice C

Códigos em Matlab para Álgebra de Quatérnios

Os códigos que serão inseridos aqui não terão os comentários iniciais que uso para *help* e para exemplos: 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 .

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

C.1 Códigos com álgebra de quatérnios

Algoritmo C.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 C.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 C.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 C.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 C.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 C.6: Módulo de quatérnio.

```

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

```

Algoritmo C.7: 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 C.8: 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 C.9: 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

```

C.2 Códigos com *plot* de coordenadas

Algoritmo C.10: Plot de coordenadas 2D.

```

1 function fc_lib_plot_coordenadas_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
7 if flag_plot_xy == 1
8     plot(x,y,'--k');
9 end
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
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 C.11: 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     % matriz de rotação 3D - X-Y-Z-Euler-Angles - Cardan-Angles
24     % matriz de rotação 3D - X-Y-Z-Euler-Angles - Cardan-Angles
25     switch o
26         case 'xyz'
27             %R = fc_lib_Rx(th)*fc_lib_Ry(phi)*fc_lib_Rz(psi);
28             R = fc_lib_R_xyz(th,phi,psi);
29         case 'xzy'
30             R = fc_lib_R_xzy(th,psi,phi);
31         case 'yxz'
32             R = fc_lib_R_yxz(phi,th,psi);
33         case 'yzx'
34             R = fc_lib_R_yzx(phi,psi,th);
35         case 'zyx'
36             R = fc_lib_R_zyx(phi,psi,th);
37         case 'zxy'
38             R = fc_lib_R_zxy(psi,th,phi);
39         case 'zxx'
40             R = fc_lib_R_zxz(psi,th,phi);
41         case 'zyz'
42             R = fc_lib_R_zyz(psi,phi,th);
43         otherwise
44             disp('Error');
45             return;
46     end
47     Body = R*Base;
48     % setas e plot
49     seta1 = [xyz, xyz+Body(:,1)];
50     seta2 = [xyz, xyz+Body(:,2)];
51     seta3 = [xyz, xyz+Body(:,3)];
52     plot3(seta1(1,:),seta1(2,:),seta1(3,:), 'Color', 'r');
53     plot3(seta2(1,:),seta2(2,:),seta2(3,:), 'Color', 'dkg');
54     plot3(seta3(1,:),seta3(2,:),seta3(3,:), 'Color', 'b');
55     % formas de apresentação
56     if flag_drawnow == 1
57         drawnow;
58     elseif flag_drawnow == 2
59         % temporizador maroto
60         tic, while toc < 0.5, end
61         drawnow;
62     elseif flag_drawnow == 3
63         % temporizador manual
64         input('Tecle algo para continuar: \n');
65         drawnow;
66     end
67 end
68 axis equal; hold off;
69 end

```

Algoritmo C.12: Plot setas 2D.

```

1  function 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
4  flag_plot_xy = flags(1);
5  flag_drawnow = flags(2);
6  figure(fh);

```

```

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

```

Algoritmo C.13: Plot setas 3D.

```

1 function fc_lib_plot_setas_3D(x,y,z,r,th,phi,flags,fh,alfa)
2 % alfa = -5*pi/6; % alfa = -11*pi/12; % alfa = -23*pi/24;
3
4 flag_plot_xyz = flags(1);
5 flag_drawnow = flags(2);
6 figure(fh);
7
8 if flag_plot_xyz == 1
9     plot3(x,y,z,'--k');
10 end
11 xlabel('X','FontSize',16,'FontWeight','bold','Color','b');
12 ylabel('Y','FontSize',16,'FontWeight','bold','Color','r');
13 zlabel('Z','FontSize',16,'FontWeight','bold','Color','g');
14 hold all; grid on;
15 str = 'b';
16
17 for k = 1:length(x)
18     L = r(k);
19     S = L/10;
20     % cálculo do corpo da seta e de sua ponta
21     setax = [x(k), x(k)+L*cos(th(k))*sin(phi(k))];
22     setay = [y(k), y(k)+L*sin(th(k))*sin(phi(k))];
23     setaz = [z(k), z(k)+L*cos(phi(k))];
24     pontax = [setax(2)+S*cos(th(k))*sin(phi(k) - alfa), setax(2), ...
25         setax(2)+S*cos(th(k))*sin(phi(k) + alfa)];
26     pontay = [setay(2)+S*sin(th(k))*sin(phi(k) - alfa), setay(2), ...
27         setay(2)+S*sin(th(k))*sin(phi(k) + alfa)];
28     pontaz = [setaz(2)+S*cos(phi(k) - alfa), setaz(2), ...
29         setaz(2)+S*cos(phi(k) + alfa)];
30     % plot
31     plot3(setax,setay,setaz, str);
32     plot3(pontax,pontay,pontaz, str);
33     if flag_drawnow == 1
34         drawnow;
35     end
36 end
37 axis equal; hold off;
38 end

```

C.3 Códigos com matrizes de rotação

Algoritmo C.14: 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 C.15: 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 C.16: 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 C.17: 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 C.18: 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), cos(psi)*cos(th) - sin(phi)*sin(psi)*sin(th), -cos(phi)*sin(th);
5       sin(psi)*sin(th) - cos(psi)*cos(th)*sin(phi), cos(psi)*sin(th) + cos(th)*sin(phi)*sin(psi),
6       cos(phi)*cos(th)];
end

```

Algoritmo C.19: 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), cos(th)*sin(phi)*sin(psi) - cos(phi)*sin(th);
5       cos(phi)*sin(psi)*sin(th) - cos(th)*sin(phi), cos(psi)*sin(th), cos(phi)*cos(th) + sin(phi)*sin(psi)*sin(th)];
6 end

```

Algoritmo C.20: 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), cos(psi)*sin(phi)*sin(th) - cos(phi)*sin(psi), cos(th)*sin(phi);
4       cos(th)*sin(psi), cos(psi)*cos(th), -sin(th);
5       cos(phi)*sin(psi)*sin(th) - cos(psi)*sin(phi), sin(phi)*sin(psi) + cos(phi)*cos(psi)*sin(th), cos(phi)*cos(th)];
6 end

```

Algoritmo C.21: Matriz de rotação na sequência **YZX**.

```

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

```

Algoritmo C.22: 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), cos(psi)*sin(phi) + cos(phi)*sin(psi)*sin(th),
4     cos(phi)*sin(psi) + cos(psi)*sin(phi)*sin(th), cos(psi)*cos(th), sin(phi)*sin(psi) - cos(phi)*cos(psi)*sin(th),
5     -cos(th)*sin(phi), sin(th), cos(phi)*cos(th)];
6 end

```

Algoritmo C.23: 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), sin(psi)*sin(th) + cos(psi)*cos(th)*sin(phi),
4     cos(phi)*sin(psi), cos(psi)*cos(th) + sin(phi)*sin(psi)*sin(th), cos(th)*sin(phi)*sin(psi) - cos(psi)*cos(phi)*sin(th),
5     -sin(phi), cos(phi)*sin(th),
6     cos(phi)*cos(th)];
end

```

Algoritmo C.24: 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), -cos(psi)*sin(phi) - cos(phi)*cos(th)*sin(psi),
4     sin(psi)*sin(th);
5     cos(phi)*sin(psi) + cos(psi)*cos(th)*sin(phi), cos(phi)*cos(psi)*cos(th) - sin(phi)*sin(psi), -cos(psi)*cos(phi)*sin(th),
6     sin(phi)*sin(th), cos(phi)*sin(th), cos(th)];
end

```

Algoritmo C.25: 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), -cos(th)*sin(psi) - cos(phi)*cos(psi)*sin(th), cos(psi)*sin(th) + cos(phi)*sin(psi)*sin(th),
4     cos(psi)*sin(th) + cos(phi)*cos(th)*sin(psi), cos(psi)*cos(th) - cos(phi)*sin(psi)*sin(th), sin(phi)*sin(th) + cos(phi)*cos(psi)*sin(th),
5     -cos(th)*sin(phi), sin(phi)*sin(th), cos(phi)];
6 end

```

C.4 Códigos auxiliares

Algoritmo C.26: Encapsulamento da função que gera arquivos em \LaTeX 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 C.27: 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

```

C.5 Códigos geradores de Figuras

Código gerador da [Figura 4.12](#).

Algoritmo C.28: 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} paralelo
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.seta.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,'fig_rotacao.3D.decomposicao.paralelo.perpendicular','eps2');

```

Código gerador da Figura 4.1.

Algoritmo C.29: Figura do exemplo de rotação 2D em Matlab®.

```

1 clear; close all; clc; % EM DESENVOLVIMENTO
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.seta.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.seta.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.seta.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.seta.2D(fh,xx,y0,alfa,th,S,'k','-');
38 th = degtord(90); fc.lib.plot.ponta.seta.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','eps');

```

Código gerador da [Figura 1.1](#).

Algoritmo C.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');

```

com uso da geração de setas em 2D:

BATATA: É Capaz que as funções que geram as setas passem a usar esta função, apesar que a construção em cada caso é diferente

Algoritmo C.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 C.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');
13 fc_lib_plot_ponta_setas_3D(fh,x1,y1,z1,0,-5*pi/6,+1*pi/4,+1*pi/4,0.25,'b','-')
14 str = '$\vec{v}$';
15 text(13*x1/16,13*y1/16,11*z1/16,str,'Interpreter','latex','FontSize',16,'Color','b');
16 %% -- pontos de construção
17 plot3([x0,x1],[y0,y1],[z0,z1],'ro','MarkerSize',8);
18 str = '$(x_0,y_0,z_0)$';
19 text(x0,y0 -1,z0,str,'Interpreter','latex','FontSize',16);
20 str = '$(x_1,y_1,z_1)$';
21 text(x1,y1 +0.1,z1 +0.1,str,'Interpreter','latex','FontSize',16);
22 %% medidas sobre os eixos
23 str = '$x_1$';
24 text(x1,-0.2,0,str,'Interpreter','latex','FontSize',16,'Color','r');
25 str = '$y_1$';
26 text(-0.2,y1,0,str,'Interpreter','latex','FontSize',16,'Color',[0 0.5 0]);
27 str = '$z_1$';
28 text(-0.2,0,z1,str,'Interpreter','latex','FontSize',16,'Color','b');
29 %% projeções
30 plot3([x1,x1],[y1,y1],[0,z1],':m');
31 plot3([x0,x1],[y0,y1],[0,0],':m');
32 plot3([0,x1],[0,y1],[z1,z1],':m');
33 %% tracejados
34 plot3([x0,x0],[y0,y0],[0,z1], '--k');
35 plot3([x0,x1],[y0,y0],[0,0], '--k');
36 plot3([x0,x0],[y0,y1],[0,0], '--k');
37 %% nos eixos
38 cor = [0.2 0 1]; % roxo
39 plot3([x1,x1],[0,y1],[0,0],':', 'Color', cor);

```

```

40 plot3([0,x1],[y1,y1],[0,0],':', 'Color', cor);
41 %% linhas de medida
42 str = '$\rho$';
43 text(x1/2,y1/2,5*z1/8,str,'Interpreter','latex','FontSize',16,'Color',[0.8 0 0.5]);
44 %% circunferência em z=0 para theta
45 cor = [0 0.5 0]; linha = '-'; % verde escuro
46 t = 0:0.05:(pi/4+0.02); u = pi/2*ones(size(t)); L = 1;
47 x = x0 + L*cos(t).*sin(u); y = y0 + L*sin(t).*sin(u); z = zeros(size(t));
48 plot3(x,y,z,'Color',cor,'LineStyle',linha);
49 str = '$\theta$';
50 text(1.3,0.4,0,str,'Interpreter','latex','FontSize',16, 'Color',cor);
51 %% --- seta para theta
52 xe = x(end); ye = y(end); ze = 0; S = 0.15;
53 alfa = -5*pi/6; beta = 0;
54 th = degtorad(135); phi = degtorad(90);
55 fc_lib_plot_ponta_setas_3D(fh,xe,ye,ze,alfa,beta,th,phi,S,cor,linha)
56 %% circunferência ao longo de v para phi
57 cor = [0.9 0.5 0]; %laranja
58 phie = acos(z1/sqrt(x1^2+y1^2+z1^2));
59 u = 0:0.05:phie; t = pi/4*ones(size(u)); L = 1;
60 x = x0 + L*cos(t).*sin(u); y = y0 + L*sin(t).*sin(u); z = z0 + L*cos(u);
61 plot3(x,y,z,'Color',cor,'LineStyle',linha);
62 str = '$\varphi$';
63 text(x1/5,y1/5,4*z1/8,str,'Interpreter','latex','FontSize',16, 'Color',cor);
64 %% --- seta para phi
65 xe = x(end); ye = y(end); ze = z(end); S = 0.15;
66 alfa = 0; beta = -5*pi/6; th = degtorad(45); phi = degtorad(140);
67 fc_lib_plot_ponta_setas_3D(fh,xe,ye,ze,alfa,beta,th,phi,S,cor,linha)
68 %% perspectiva e salvar
69 axis([0,3,0,3,0,3]); %axis equal;
70 saveas(gcf,'fig.coordenadas_3D','eps2');

```

com uso da geração de setas em 3D.

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

```

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

```

Códigos geradores da [Figura 3.1](#).

Algoritmo C.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_setas_2D(fh,xx,y0,alfa,th,S,'k','-');
27 th = degtorad(90);
28 fc_lib_plot_ponta_setas_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 C.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_setas_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_setas_2D(fh,xx,y0,alfa,th,S,'k','-');
27 th = degtorad(90);
28 fc_lib_plot_ponta_setas_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');

```

Códigos geradores dos Exemplos de 3 a 8.

Algoritmo C.36: 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: %stheta-[%s] = %g°; %sphi-[%s] = %g°; %spsi-[%s] = %g°',...
25 titulo = sprintf('%s%s%s: \\phi-[%s] = %g°; \\theta-[%s] = %g°; \\psi-[%s] = %g°',...
26     upper(o(1)), upper(o(2)), upper(o(3)), ...
27     upper(o(1)), radtodeg(th), ...
28     upper(o(2)), radtodeg(phi), ...
29     upper(o(3)), radtodeg(psi));
30 end
31 %% função escrita na figura
32 function fc_int_text(fh)
33 figure(fh);
34 for i = 0:3
35     text(2*i+0.2,0.2,0,sprintf('\\xi-[%d]',i));
36 end
37 end
38 %% ---
39 function [TH, PHI, PSI] = fc_int_THPHIPSI(k,l)
40 if k == 1
41     A = [0 pi/2 pi/2 pi/2]';
42     B = [0 0 pi/2 pi/2]';
43     C = [0 0 0 pi/2]';
44 elseif k == 2
45     A = [0 pi/4 pi/4 pi/4]';
46     B = [0 0 pi/2 pi/2]';
47     C = [0 0 0 pi/4]';
48 end
49 switch l
50     case 1
51         TH = A; PHI = B; PSI = C;
52     case 2
53         TH = A; PHI = C; PSI = B;
54     case 3
55         TH = B; PHI = A; PSI = C;
56     case 4
57         TH = C; PHI = A; PSI = B;
58     case 5
59         TH = C; PHI = B; PSI = A;
60     case 6
61         TH = B; PHI = C; PSI = A;
62     case 7
63         TH = B; PHI = C; PSI = A;
64     case 8
65         TH = C; PHI = B; PSI = A;
66     otherwise
67         TH = A; PHI = B; PSI = C;
68         disp('Erro');
69 end
70 end
71 %% ---

```

```

72 function o = fc_int_o_xyz(k)
73 lista = {'xyz', 'xzy', 'yxz', 'yzx', 'zyx', 'zxy', 'zxz', 'zyz'};
74 o = char(lista(k));
75 end

```

Algoritmo C.37: 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_setas_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_setas_2D(fh,xx,y0,alfa,th,S,'k','-');
27 th = degtorad(90);
28 fc_lib_plot_ponta_setas_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');

```

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.

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.

CONCI, A. A eficiência das rotações em c.g.: quatérnios: Cap 2 (apêndice). **UFF**, 2014. Disponível em: <http://www2.ic.uff.br/~aconci/quaternioUFF.pdf>. Acesso em: 20 jan. 2018.

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.

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/~ucappgu/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. Link digital para compilação de todos os artigos do autor. 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.

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: quaternions, 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. 3Blue1Brown. Disponível em: <https://youtu.be/XFDM1ip5HdU>. Acesso em: 20 jan. 2018.

SANDERSON, G. **Quaternions and 3d rotation, explained interactively**. 3Blue1Brown, 2018. 3Blue1Brown. 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. 3Blue1Brown. Disponível em: <https://youtu.be/d4EgbgTm0Bg>. Acesso em: 06 set. 2018.

STEVENS, M. **How To Count Past Infinity**. Vsauce, 2016. Vsauce. 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 (Dissertação (mestrado)) — Universidade Federal do ABC - PROFMAT, 2017. Disponível em: https://sca.profmtat-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. 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. Hypercomplex number. **MathWorld—A Wolfram Web Resource**, jan. 2019. Disponível em: <http://mathworld.wolfram.com/HypercomplexNumber.html>. Acesso em: 01 jan. 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. 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.