MAC420/5744: Introdução à Computação Gráfica

Marcel P. Jackowski
mjack@ime.usp.br

Aula #9: Rotações e quatérnios

Rotações

- Úteis para animação
  - Objetos
  - Câmera
- Descrição de modelos articulados
  - Corpo humano
  - Robôs
- Orientações de corpos rígidos
- 2D ou 3D

Representado orientações

- Existirá um único jeito de representar uma orientação em 3D ?
  - Análogo à coordenadas cartesianas?
- Na verdade existem várias formas:
  - Ângulos de Euler
  - Vetores de rotação (eixo/ângulo)
  - Matrizes 3x3
  - Quatérnios
  - etc

Rotações em 3D

- Matrizes ortogonais
  - $RR^T = R^TR = I$
  - $\det(R) = 1$

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

- Colunas (e linhas) da matriz formam uma base ortonormal no $\boldsymbol{R}^3$
- O produto de matrizes ortogonais será também ortogonal

Ângulos de Euler e matrizes

- Para construir uma matriz a partir dos ângulos de Euler, simplesmente multiplicamos uma sequência de matrizes de rotação:

$$\mathbf{R}_x \cdot \mathbf{R}_y \cdot \mathbf{R}_z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_x & s_x \\ 0 & -s_x & c_x \end{bmatrix} \cdot \begin{bmatrix} c_y & 0 & -s_y \\ 0 & 1 & 0 \\ s_y & 0 & c_y \end{bmatrix} \cdot \begin{bmatrix} c_z & s_z & 0 \\ -s_z & c_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_y c_z & c_y s_z & -s_y \\ s_x s_y c_z - c_x s_z & s_x s_y s_z + c_x c_z & s_x c_y \\ c_x s_y c_z + s_x s_z & c_x s_y s_z - s_x c_z & c_x c_y \end{bmatrix}$$

Ângulos de Euler

- Representação de uma orientação através de 3 números
- Uma sequência de rotações ao redor de eixos principais é chamada de sequência angular de Euler
- Assumindo um limite de até 3 rotações, sem rotações sucessivas no mesmo eixo, podemos utilizar quaisquer destas 12 combinações:

XYZ	XZY	XYX	XZX
YXZ	YZX	YXY	YZY
ZXY	ZYX	ZXZ	ZYZ

Comutatividade

- $R_x(90^o)\ R_y(90^o) = R_y(90^o)\ R_x(90^o)$?

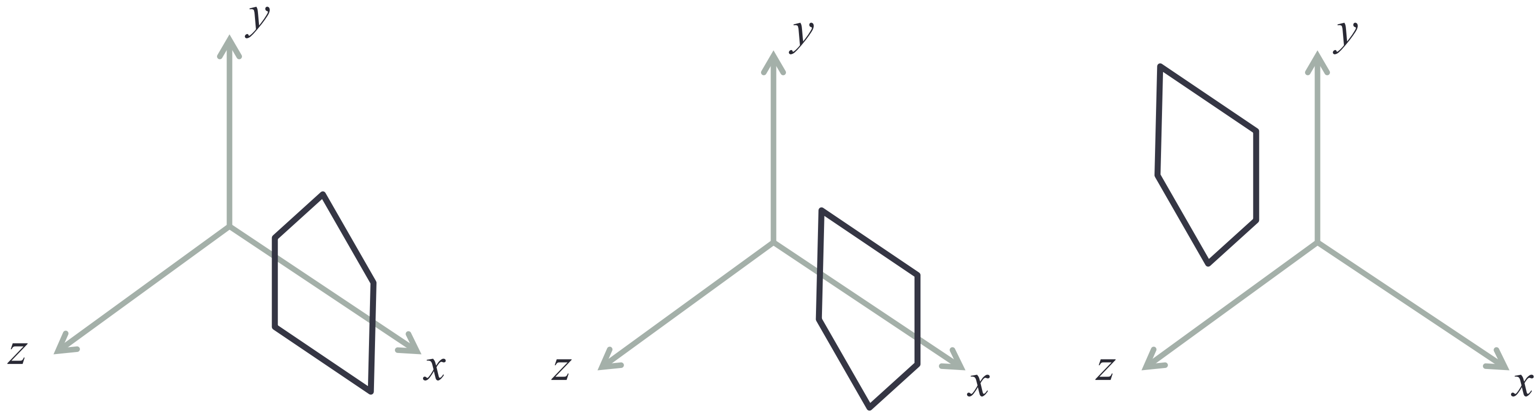
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & c_x & -s_x \\ 0 & s_x & c_x \end{bmatrix} \cdot \begin{bmatrix} c_y & 0 & s_y \\ 0 & 1 & 0 \\ -s_y & 0 & c_y \end{bmatrix} = \begin{bmatrix} c_y & 0 & s_y \\ s_x s_y & c_x & -s_x c_y \\ -c_x s_y & s_x & c_x c_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} c_y & 0 & s_y \\ 0 & 1 & 0 \\ -s_y & 0 & c_y \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_x & -s_x \\ 0 & s_x & c_x \end{bmatrix} = \begin{bmatrix} c_y & s_x s_y & c_x s_y \\ 0 & c_x & -s_x \\ -s_y & s_x c_y & c_x c_y \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

$$s_x = s_y = \text{sen}(90^o) = 1$$
$$c_x = c_y = \cos(90^o) = 0$$

Equivalências

- $T = R_y(180^o)\ R_x(180^o)$



$R_x(180^o)$ $R_y(180^o)$

$T = R_y(180^o)\ R_x(180^o) = R_z(180^o)$

Equivalências

- $R_y(180^o)\, R_x(180^o) = R_z(180^o)$

$$\begin{bmatrix} cy & 0 & sy \\ 0 & 1 & 0 \\ -sy & 0 & cy \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & cx & -sx \\ 0 & sx & cx \end{bmatrix} = \begin{bmatrix} cy & sxsy & cxsy \\ 0 & cx & -sx \\ -sy & sxcy & cxcy \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} cz & sz & 0 \\ -sz & cz & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$sx = sy = \text{sen}(180^o) = 0$

$cx = cy = \cos(180^o) = -1$

Ordem das rotações

- Como a multiplicação de matrizes não é comutativa, a ordem das operações é importante
- As rotações são realizadas assumindo um eixo fixo do mundo, e não local ao objeto
- Para usarmos os ângulos de Euler, devemos escolher uma das 12 representações
- A melhor sequência depende do que pretendemos fazer, pois podem existir diferenças práticas dependendo desta escolha.

Ângulos de Euler

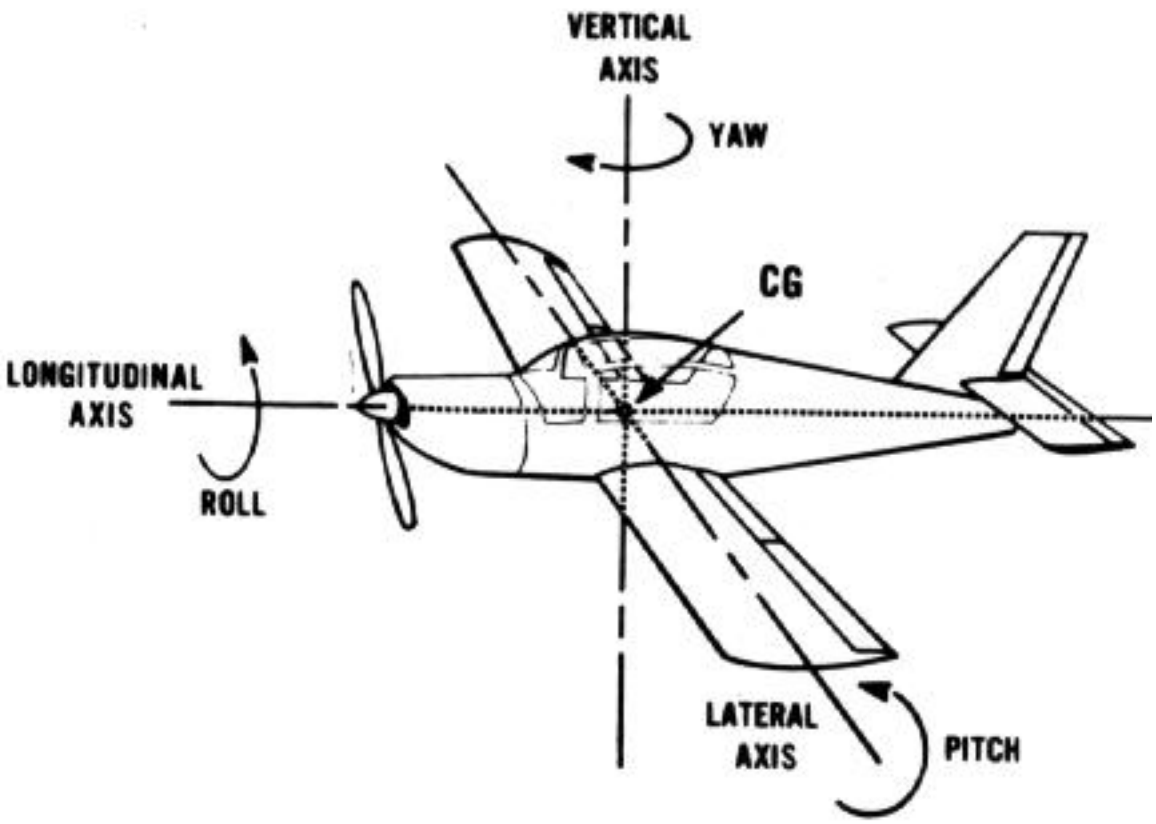
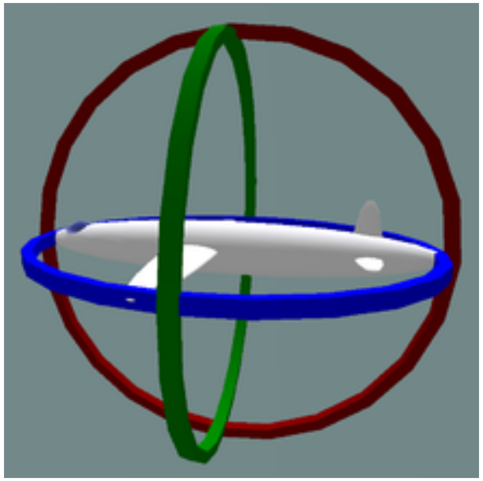


Figure 3–8 Axes of the Airplane



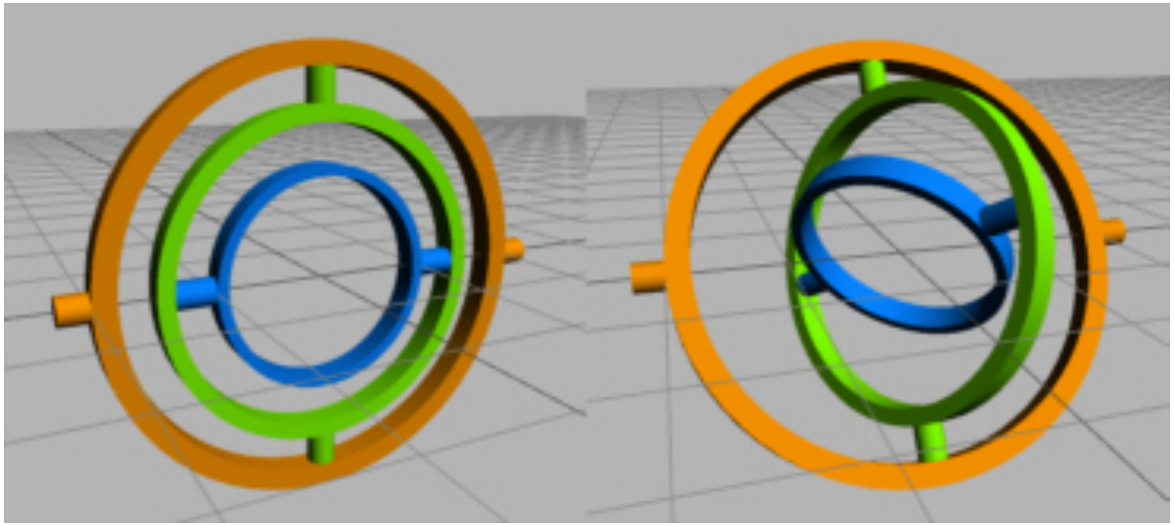
"Gimbals"

RGB=(YAW,PITCH,ROLL)

"Gimbal lock"

- Um problema dos ângulos de Euler é chamado de "gimbal lock"
- Ao longo de várias rotações, é possível que dois dos eixos se alinhem, resultando em uma perda temporária de graus de liberdade

"Gimbal Lock"



"Gimbal lock"

- Exemplo com a ordenação Z-X-Z

$$R = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Z_1X_2Z_3 = \begin{bmatrix} c_1c_3 - c_2s_1s_3 & -c_1s_3 - c_2c_3s_1 & s_1s_2 \\ c_3s_1 + c_1c_2s_3 & c_1c_2c_3 - s_1s_3 & -c_1s_2 \\ s_2s_3 & c_3s_2 & c_2 \end{bmatrix}$$

"Gimbal lock"

$$\beta = 0$$

$$\sin\beta = 0$$

$$\cos\beta = 1$$

$$R = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} \cos\alpha\cos\gamma - \sin\alpha\sin\gamma & -\cos\alpha\sin\gamma - \sin\alpha\cos\gamma & 0 \\ \sin\alpha\cos\gamma + \cos\alpha\sin\gamma & -\sin\alpha\sin\gamma + \cos\alpha\cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} \cos(\alpha+\gamma) & -\sin(\alpha+\gamma) & 0 \\ \sin(\alpha+\gamma) & \cos(\alpha+\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

“Gimbal lock”

- Exemplo com a ordenação X-Y-Z

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\beta = \frac{\pi}{2}$$

“Gimbal lock”

$$\beta = \frac{\pi}{2}$$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} 0 & 0 & 1 \\ \sin\alpha\cos\gamma + \cos\alpha\sin\gamma & -\sin\alpha\sin\gamma + \cos\alpha\cos\gamma & 0 \\ -\cos\alpha\cos\gamma + \sin\alpha\sin\gamma & \cos\alpha\sin\gamma + \sin\alpha\cos\gamma & 0 \end{bmatrix}$$

$$R = \begin{bmatrix} 0 & 0 & 1 \\ \sin(\alpha+\gamma) & \cos(\alpha+\gamma) & 0 \\ -\cos(\alpha+\gamma) & \sin(\alpha+\gamma) & 0 \end{bmatrix}$$

Ângulos de Euler e interpolação

- Representação compacta (3 valores)
- Requer operações trigonométricas para transformar para a forma matricial
- Podemos interpolar entre os três valores independentemente
  - Isso resultará em uma interpolação que seguirá um caminho diferente dependendo do esquema (12 opções) que você escolheu
- Isto pode ou não ser um problema, dependendo da aplicação
  - Pode sofrer do efeito de "gimbal lock"

Representação matricial

- Forma eficiente de aplicar rotações à objetos gométricos
- Amplamente utilizada nos pipelines de processamento gráfico em software e hardware (e.g. GPUs)
- Por quê não utilizamos sempre matrizes?
  - Problemas numéricos
  - Questões relacionadas a interação com usuário
  - Problemas com interpolação

Números complexos

- $\mathbb{C} = \{a + bi : a, b \in \mathbb{R}, \quad i^2 = -1\}$
- Any complex number has a length, given by the Pythagorean formula:

$$|a + bi| = \sqrt{a^2 + b^2}.$$

- We can add and subtract in $\mathbb{C}$. For example:

$$a + bi + c + di = (a + c) + (b + d)i.$$

- We can also multiply, which is much messier:

$$(a + bi)(c + di) = (ac - bd) + (ad + bc)i$$

Números complexos

Fortunately, there is a better way to multiply complex numbers, thanks to Leonhard Euler:

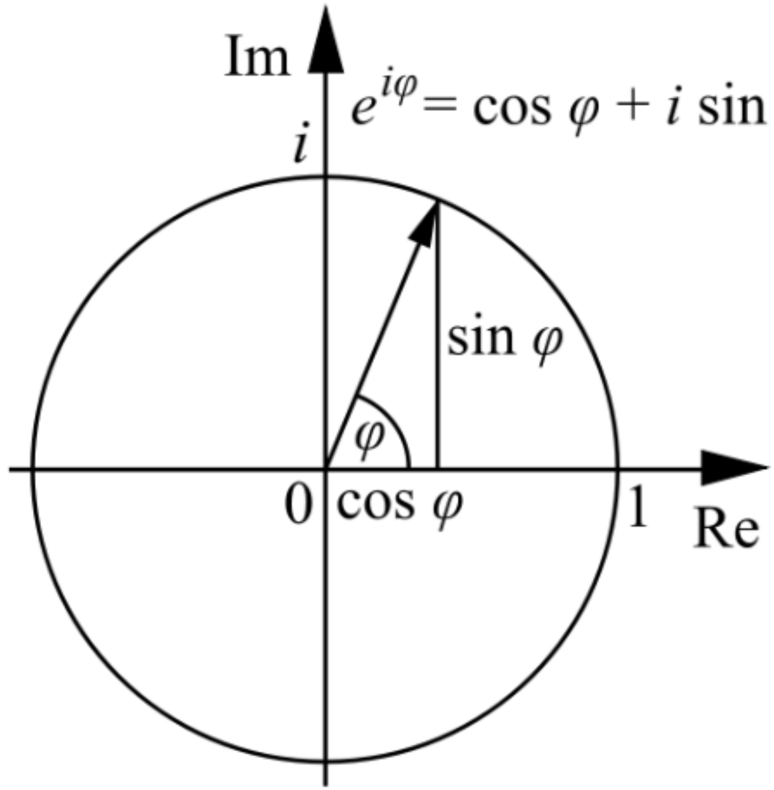
Figure: Handman's portrait of Euler. *Wikimedia Commons.*

who proved:

$$e^{i\varphi} = \cos\varphi + i\sin\varphi$$

Interpretação geométrica

Geometrically, this formula says $e^{i\varphi}$ lies on the unit circle in $\mathbb{C}$:



Produto entre números complexos

- $e^{i\varphi}$ has unit length.
- If we multiply by a positive number, r, we get a complex number of length r:

$$re^{i\varphi}.$$

- By adjusting the length r and angle φ, we can write any complex number in this way!
- In a calculus class, this trick goes by the name **polar coordinates**.

Produto entre números complexos

And this gives a great way to multiply complex numbers:

- Remember our formula was:

$$(a+bi)(c+di) = (ac-bd) + (ad+bc)i$$

- Instead, we can write each factor in polar coordinates:

$$a+bi = re^{i\varphi}, \quad c+di = se^{i\theta}$$

- And now:

$$(a+bi)(c+di) = re^{i\varphi}se^{i\theta} = rse^{i(\varphi+\theta)}.$$

- In words: to multiply two complex numbers, *multiply their lengths and add their angles!*

Produto entre números complexos

In particular, if we multiply a given complex number z by

$$e^{i\varphi}$$

which has unit length 1, the result:

$$e^{i\varphi} z$$

has the same length as z.

It is rotated by φ degrees.

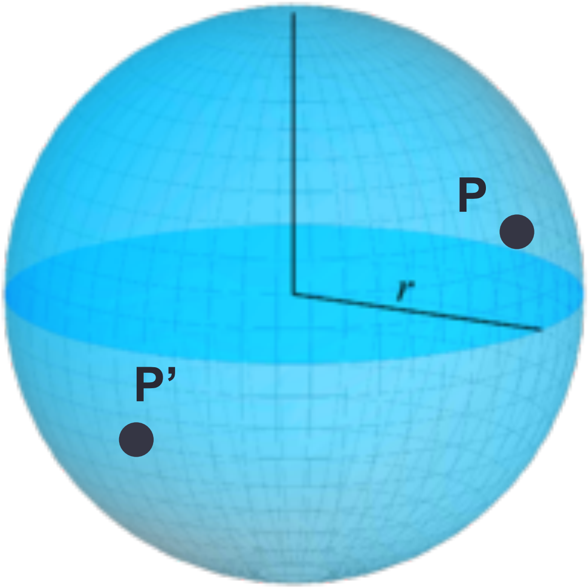
Produtos em *C* e rotações no R^2

So, we can use *complex arithmetic* (multiplication) to do a *geometric operation* (rotation).

The 19th century Irish mathematician and physicist William Rowan Hamilton was fascinated by the role of $\mathbb{C}$ in two-dimensional geometry.

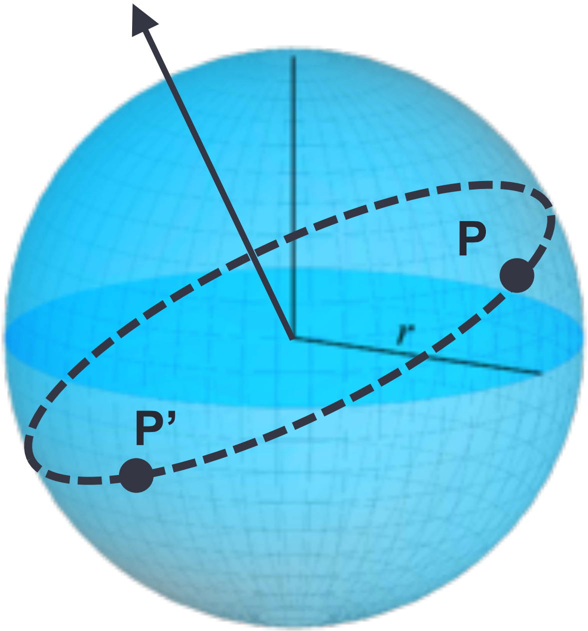
Rotações na esfera

- Achar o menor caminho que leva P a P’, na superfície da esfera



Rotações na esfera

- O menor caminho entre dois pontos em uma esfera é o grande círculo que passa pelos dois pontos
- Para cada círculo, existirá um vetor normal correspondente
- A rotação ao redor deste vetor nos deslocará do primeiro ponto (P) para o segundo ponto (P').



Quatérnios

- Conceito matemático inventado por William Rowan Hamilton (Dublin, 1843)
- Generaliza o cálculo vetorial e os números complexos
- Na prática, útil na representação de rotações
- Um quatérnio possui 4 componentes:

$$\mathbf{q} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}$$

Quatérnios (espaço imaginário)

- Uma extensão dos números complexos
- Dos quatro componentes, um deles é um número real, e os outros formam um vetor no espaço imaginário *ijk*

$$\mathbf{q} = q_0 + iq_1 + jq_2 + kq_3$$

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

$$i = jk = -kj$$

$$j = ki = -ik$$

$$k = ij = -ji$$

Quatérnios

- Certas vezes, eles são expressos por uma combinação de um valor escalar *s* e um vetor ***v***

$$\mathbf{q} = \langle s, \mathbf{v} \rangle \qquad \mathbf{p} = \langle 0, \mathbf{v} \rangle$$

$$s = q_0$$
$$\mathbf{v} = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}$$

Propriedades

$$q_1 q_2 \neq q_2 q_1$$

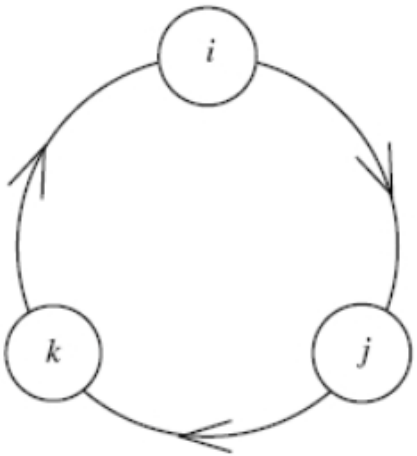
$$(q_1 q_2) q_3 = q_1 (q_2 q_3)$$

$$(q_1 + q_2) q_3 = q_1 q_3 + q_2 q_3$$

$$q_1 (q_2 + q_3) = q_1 q_2 + q_1 q_3$$

$$\alpha (q_1 + q_2) = \alpha q_1 + \alpha q_2$$

$$(\alpha q_1) q_2 = \alpha (q_1 q_2) = q_1 (\alpha q_2)$$



$$q^{-1} = \frac{1}{|q|^2} (q_0, -q)$$

Quatérnios unitários

- Para nossa conveniência, usaremos quatérnios unitários, por serem suficientes para os nossos propósitos

$$|\mathbf{q}| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1$$

- Eles correspondem a um conjunto de vetores que formam uma "superfície" na hiperesfera de raio 1
- Esta "superfície" é na verdade um volume (3D) no espaço 4D
  - Pode ser visualizado como uma extensão do conceito de uma superfície 2D em uma esfera em 3D.

Produto interno

- Funciona da mesma maneira que o produto interno entre dois vetores

$$\mathbf{p} \cdot \mathbf{q} = p_0 q_0 + p_1 q_1 + p_2 q_2 + p_3 q_3 = |\mathbf{p}||\mathbf{q}| \cos\varphi$$

- O ângulo entre dois quatérnios em 4D representa duas vezes o ângulo necessário para rotacionar em 3D

Produto entre quatérnios

- Podemos escrever um produto de quatérnios da seguinte forma:

$$\mathbf{pq} = (p_0 + p_1 i + p_2 j + p_3 k)(q_0 + q_1 i + q_2 j + q_3 k)$$
$$= p_0 q_0 + p_0 q_1 i + p_0 q_2 j + p_0 q_3 k +$$
$$p_1 q_0 i + p_1 q_1 i^2 + p_1 q_2 ij + p_1 q_3 jk +$$
$$p_2 q_0 j + p_2 q_1 ji + p_2 q_2 j^2 + p_2 q_3 jk +$$
$$p_3 q_0 k + p_3 q_1 ki + p_3 q_2 kj + p_3 q_3 k^2$$

$$= (p_0 + p)(q_0 + q)$$
$$= (p_0 q_0 + p_0 q + q_0 p + pq)$$
$$= (p_0 q_0 - p \cdot q + p_0 q + q_0 p + p \times q) = \left\langle p_0 q_0 - p \cdot q, p_0 q + q_0 p + p \times q \right\rangle$$

Quatérnios unitários como rotações

- Um quatérnio pode representar uma rotação θ ao redor de um eixo unitário **a**:

$$\mathbf{q} = \begin{bmatrix} \cos\frac{\theta}{2} & a_x \sin\frac{\theta}{2} & a_y \sin\frac{\theta}{2} & a_z \sin\frac{\theta}{2} \end{bmatrix}$$

ou

$$\mathbf{q} = \left\langle \cos\frac{\theta}{2}, \mathbf{a}\sin\frac{\theta}{2} \right\rangle$$

- Se **a** é unitário, **q** também será.
- A composição de duas rotações $\mathbf{q}_1$ e $\mathbf{q}_2$ é igual a $\mathbf{q}=\mathbf{q}_2\mathbf{q}_1$

Quatérnios como rotações ?

- Pense que o quatérnio de rotação representa uma esfera ao redor de um objeto
  - Movimentando o mouse ao longo da esfera, o objeto é rotacionado
- Motivações:
  - $R_y(45)R_x(90) \mathrel{!=} R_x(90)R_y(45)$
  - $R_y(180)R_x(180) = R_z(180)$
  - Gimbal lock ?

Quatérnios unitários e rotações

- Assuma $\mathbf{p}=\langle 0, \mathbf{v}\rangle$, um quatérnio que representa um ponto 3D (normalizado)
- Assuma um quatérnio de rotação unitário **r** que codifica a rotação ao redor de um certo eixo
- A transformação abaixo rotaciona o ponto p através de **r**, que resulta na nova posição $\mathbf{p}'=\mathbf{r}\mathbf{p}\mathbf{r}^{-1}$
- Novamente **p'** é um quatérnio com parte escalar 0, e parte vetorial que determina as coordenadas do novo ponto em 3D.

Quatérnio para matriz

- Para converter um quatérnio em matriz de rotação:

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

Quatérnio para eixo-ângulo

- O eixo e ângulo de rotação podem ser derivados a partir dos componentes do quatérnio:

$$\mathbf{q} = q_0 + q_1 i + q_2 j + q_3 k$$

$$\theta = 2\cos^{-1} q_0$$

$$(a_x, a_y, a_z) = \frac{1}{\sin(\theta/2)}(q_1, q_2, q_3)$$

Esferas

- Pense em uma pessoa na superfície de uma grande esfera
- Do ponto de vista desta pessoa, ela pode se mover em dois eixos ortogonais principais (frente/trás) e (esquerda/direita)
- Não existe a percepção de nenhum pólo fixo ou latitude/longitude
- Ela caminhará em um "plano" 2D infinito, mas eventualmente retornará ao ponto de partida.

Hiperesferas

- Se extendermos esse conceito para a movimentação na hiperesfera dos quatérnios unitários:
  - A pessoa tem agora 3 direções ortogonais para seguir
- Independentemente da sua orientação neste espaço, ela pode ser viajar usando uma combinação de frente/trás, esquerda/direita e cima/baixo
- Caso ela caminhe suficientemente longe, ela voltará ao seu ponto de partida.

Hiperesferas

- Agora, considere a posição de uma pessoa nesta hiperesfera como uma orientação
- Qualquer movimento ao longo dos eixos neste espaço curvo corresponde à um incremento rotacional ao longo de um dos eixos no espaço real
  - Distâncias na hiperesfera correspondem à ângulos no espaço 3D
- Movimentando-se ao longo de uma direção arbitrária corresponderá à uma rotação sob um eixo arbitrário
- Se você for muito longe em uma direção, você voltará ao ponto de partida
  - Correspondente à rotacionar 360 graus ao redor de um eixo

Hiperesferas

- Uma distância x ao longo da superfície da hiperesfera corresponde à rotação por $2x$ radianos
- Isso significa que o movimento ao longo de um arco de 90 graus na hiperesfera corresponde à uma rotação de 180 graus
  - Movimentado-se por 180 graus resultará em uma rotação de 360 graus em 3D
- Isso implica que $\mathbf{q}$ e $-\mathbf{q}$ representam a mesma orientação

Interpolação de quatérnios

- Utilizaremos interpolação para determinar quatérnios intermediários
- Lembre-se que existem dois vetores redundantes no espaço dos quatérnios para cada orientação única em 3D
  - Dois caminhos a seguir: ($< 180^\circ$) e ($> 180^\circ$)
- Neste caso, negamos um deles caso o produto interno seja < 0.

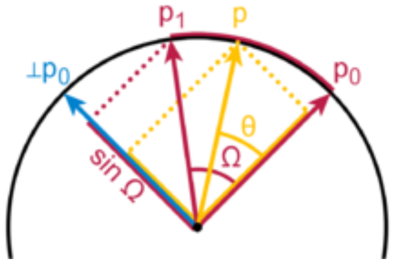
Interpolação na esfera

- Relembrando a interpolação linear (Lerp)
  - Equação parametrizada da reta

$$P_L(t) = (1\text{-}t)P + tP' = P + t(P'\text{-}P)$$

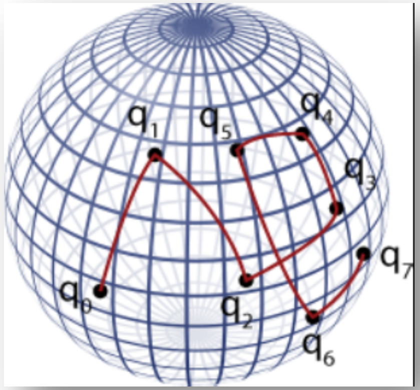
- Interpolação esférica (Slerp)
  - Precisamos levar em conta o ângulo relativo ao arco correspondente ao menor caminho

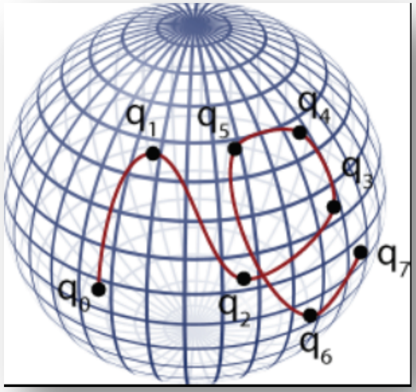
$$P_S(t) = \frac{\sin[(1-t)\Omega]}{\sin(\Omega)}P + \frac{\sin[t\Omega]}{\sin(\Omega)}P'$$



Interpolando mais que duas rotações

- Abordagem trivial, interpolamos quatérnios consecutivos utilizando a operação *Slerp*
  - Rotações contínuas
- Velocidade angular não é suave nas junções
  - Splines!





Tarefa

- Leitura suplementar:
  - Erik Dam, Martin Koch, Martin Lillholm,"Quaternions, Interpolation and Animation", Technical Report DIKU-TR-98/5
  - http://web.mit.edu/2.998/www/QuaternionReport1.pdf
- Escreva um vertex shader que receba como parâmetro um quatérnio de rotação e rotacione os vértices de entrada usando o quatérnio especificado.