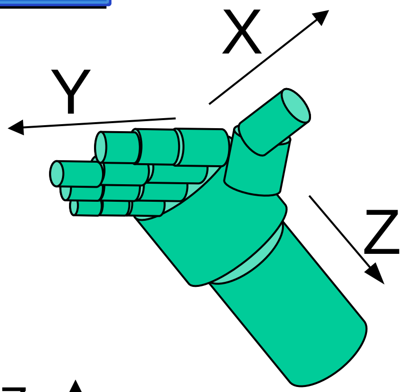
Mouvements 3-D

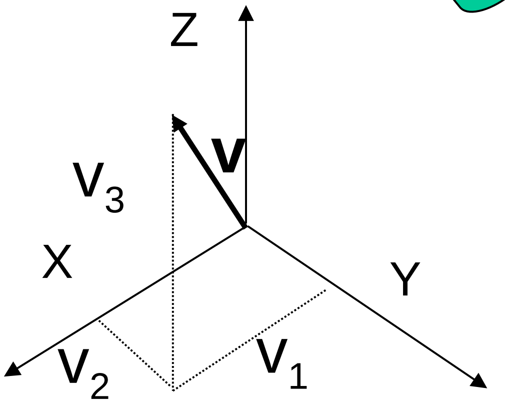
Christian Gentil / Neveu Marc

Points et Repères 3-D

- Axes xyz
  - Règle main droite $X \times Y = Z$

$$v = v_1 x_j + v_2 y_j + v_3 z_j = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$



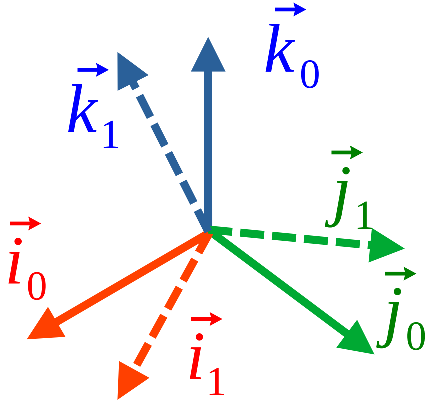


2 repères en rotation selon une origine commune

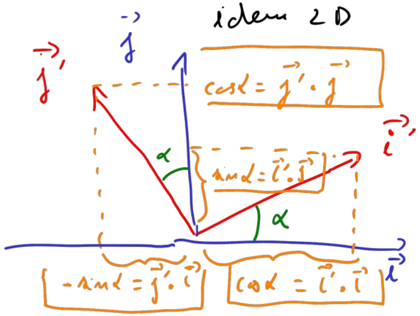
$$\vec{i}_1 = x_{11}\,\vec{i}_0 + x_{12}\,\vec{j}_0 + x_{13}\,\vec{k}_0$$
$$\vec{j}_1 = y_{11}\,\vec{i}_0 + y_{12}\,\vec{j}_0 + y_{13}\,\vec{k}_0$$
$$\vec{k}_1 = z_{11}\,\vec{i}_0 + z_{12}\,\vec{j}_0 + z_{13}\,\vec{k}_0$$

Expression de (x1,y1,z1) en fonction de (x0,y0,z0)

$$\begin{bmatrix} \vec{i}_1 \\ \vec{j}_1 \\ \vec{k}_1 \end{bmatrix} = \begin{bmatrix} {}^0 x_{11} & {}^0 y_{11} & {}^0 z_{11} \\ {}^0 x_{12} & {}^0 y_{12} & {}^0 z_{12} \\ {}^0 x_{13} & {}^0 y_{13} & {}^0 z_{13} \end{bmatrix} \begin{bmatrix} \vec{i}_0 \\ \vec{j}_0 \\ \vec{k}_0 \end{bmatrix}$$



idem 2D



$$\boxed{\begin{aligned} \vec{i}\,' &= \cos\alpha\,\vec{i} + \sin\alpha\,\vec{j} \\ \vec{j}\,' &= -\sin\alpha\,\vec{i} + \cos\alpha\,\vec{j} \end{aligned}}$$

${}^iR_j$: matrice de rotation du repère j dans i

Forme en Cosinus

rotation considérée comme produit scalaire

pour ${}^0x_1$, ${}^0y_1$, et ${}^0z_1$ et les axes ${}^0x_0$, ${}^0y_0$, ${}^0z_0$

$$\vec{i}_1 \cdot \vec{i}_0 = \vec{i}_1^T \vec{i}_0 = x_{11}$$

$$\vec{i}_1 \cdot \vec{j}_0 = \vec{i}_1^T \vec{j}_0 = x_{12}$$

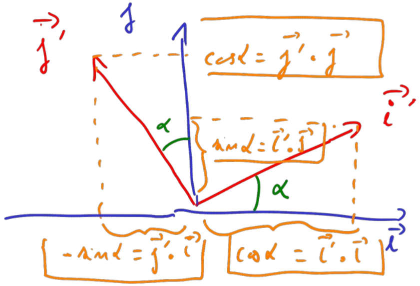
$$\vec{i}_1 = x_{11} \vec{i}_0 + x_{12} \vec{j}_0 + x_{13} \vec{k}_0$$

$$\vec{j}_1 = y_{11} \vec{i}_0 + y_{12} \vec{j}_0 + y_{13} \vec{k}_0$$

$$\vec{k}_1 = z_{11} \vec{i}_0 + z_{12} \vec{j}_0 + z_{13} \vec{k}_0$$

$$ {}^0R_1 = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ y_{11} & y_{12} & y_{13} \\ z_{11} & z_{12} & z_{13} \end{bmatrix} = \begin{bmatrix} \vec{i}_1^T \vec{i}_0 & \vec{i}_1^T \vec{j}_0 & \vec{i}_1^T \vec{k}_0 \\ \vec{j}_1^T \vec{i}_0 & \vec{j}_1^T \vec{j}_0 & \vec{j}_1^T \vec{k}_0 \\ \vec{k}_1^T \vec{i}_0 & \vec{k}_1^T \vec{j}_0 & \vec{k}_1^T \vec{k}_0 \end{bmatrix}$$

idem 2D



$$\boxed{\begin{aligned} \vec{i}' &= \cos\alpha\, \vec{i} + \sin\alpha\, \vec{j} \\ \vec{j}' &= -\sin\alpha\, \vec{i} + \cos\alpha\, \vec{j} \end{aligned}}$$

Inverse d'une matrice de rotation

Repère 0 / repère 1

$$\vec{i}_0 = x'_{11}\vec{i}_1 + x'_{12}\vec{j}_1 + x'_{13}\vec{k}_1$$
$$\vec{j}_0 = y'_{11}\vec{i}_1 + y'_{12}\vec{j}_1 + y'_{13}\vec{k}_1$$
$$\vec{k}_0 = z'_{11}\vec{i}_1 + z'_{12}\vec{j}_1 + z'_{13}\vec{k}_1$$

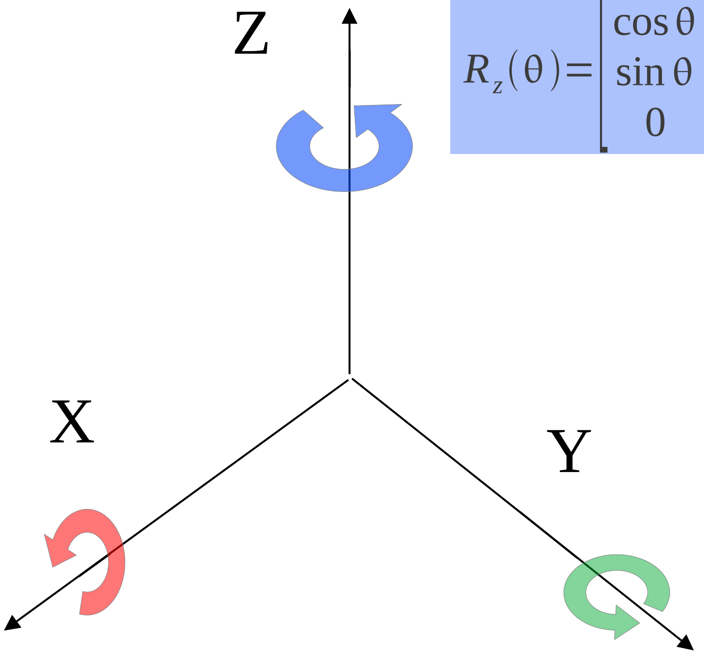
$$^0R_1 = \begin{bmatrix} \vec{i}_1^T\vec{i}_0 & \vec{i}_1^T\vec{j}_0 & \vec{i}_1^T\vec{k}_0 \\ \vec{j}_1^T\vec{i}_0 & \vec{j}_1^T\vec{j}_0 & \vec{j}_1^T\vec{k}_0 \\ \vec{k}_1^T\vec{i}_0 & \vec{k}_1^T\vec{j}_0 & \vec{k}_1^T\vec{k}_0 \end{bmatrix}$$

$$^1R_0 = \begin{bmatrix} \vec{i}_0^T\vec{i}_1 & \vec{i}_0^T\vec{j}_1 & \vec{i}_0^T\vec{k}_1 \\ \vec{j}_0^T\vec{i}_1 & \vec{j}_0^T\vec{j}_1 & \vec{j}_0^T\vec{k}_1 \\ \vec{k}_0^T\vec{i}_1 & \vec{k}_0^T\vec{j}_1 & \vec{k}_0^T\vec{k}_1 \end{bmatrix} = {}^0R_1^T$$

L'inverse d'une matrice de rotation est sa transposée, car repère orthonormé :

$$\vec{V}_i \cdot \vec{V}_j = \delta_{i,j}$$

Rotation autour des axes



Rotation θ *selon* z

$$R_z(\theta)=\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}$$

Rotation θ *selon* x

$$R_x(\theta)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\theta & -\sin\theta\\ 0 & \sin\theta & \cos\theta\end{bmatrix}$$

Rotation θ *selon* y

$$R_y(\theta)=\begin{bmatrix}\cos\theta & 0 & \sin\theta\\ 0 & 1 & 0\\ -\sin\theta & 0 & \cos\theta\end{bmatrix}$$

Transformations Spatiales

Représentation de l'Orientation

- Une matrice de rotation est redondante (9 termes)
  - seulement 3 termes indépendants
- Représentations à 3 ou 4 termes
  - angles d'Euler
  - vecteurs de Rodrigues

Représentations à 4 termes
Quaternions

Comme en 2-D
- matrice => 4 termes
- utilisation des complexes => 2 termes

Angles d'Euler

Tout ensemble non-redondant de 3 rotations successives selon les axes principaux

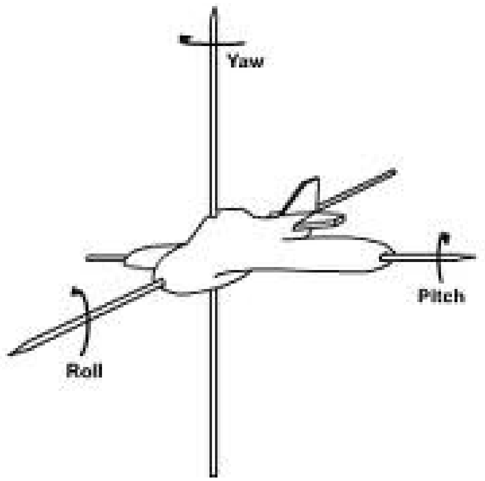
12 systèmes (parmi les plus communs : ZYZ)

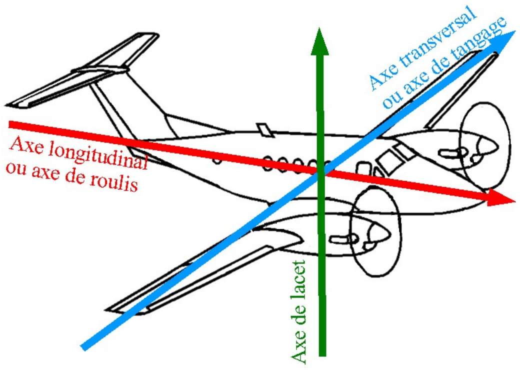
$$R_{zyz}(\phi\theta\psi)=R_z(\phi)R_y(\theta)R_z(\psi)$$

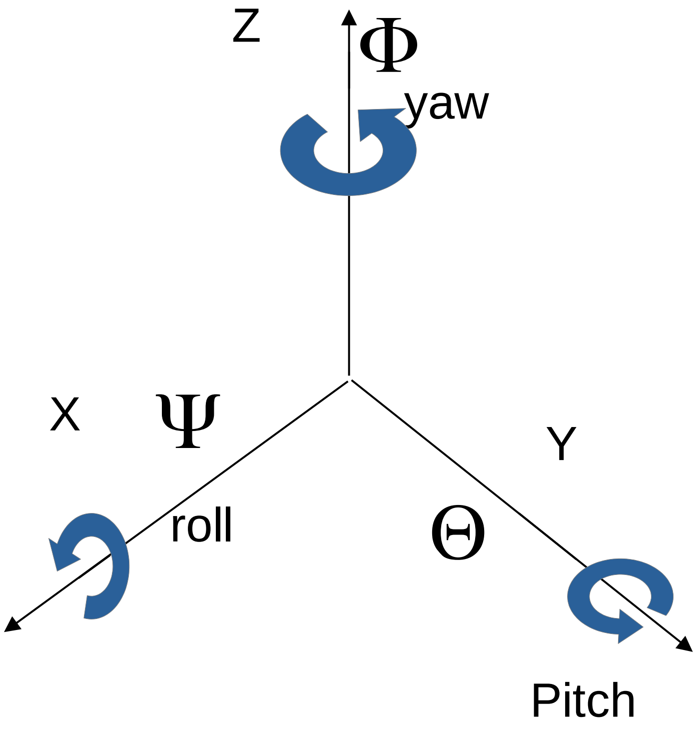
$$R_{zyz}(\phi\theta\psi)=\begin{bmatrix} c\phi & -s\phi & 0 \\ s\phi & c\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}\begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

angles Roll, Pitch Yaw

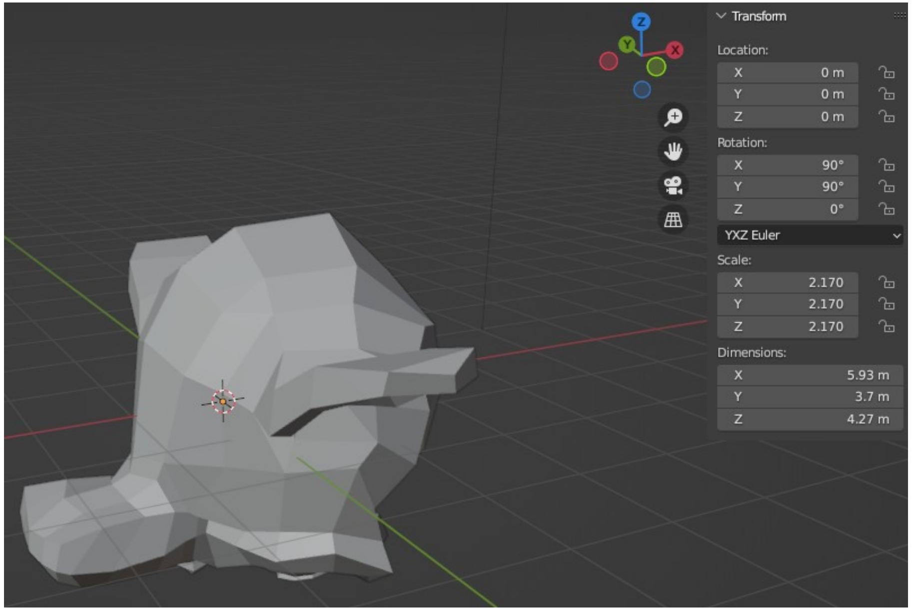
$$R_{xyz}(\Phi\,\Theta\,\Psi)=R_z(\Phi)\,R_y(\Theta)\,R_x(\Psi)$$

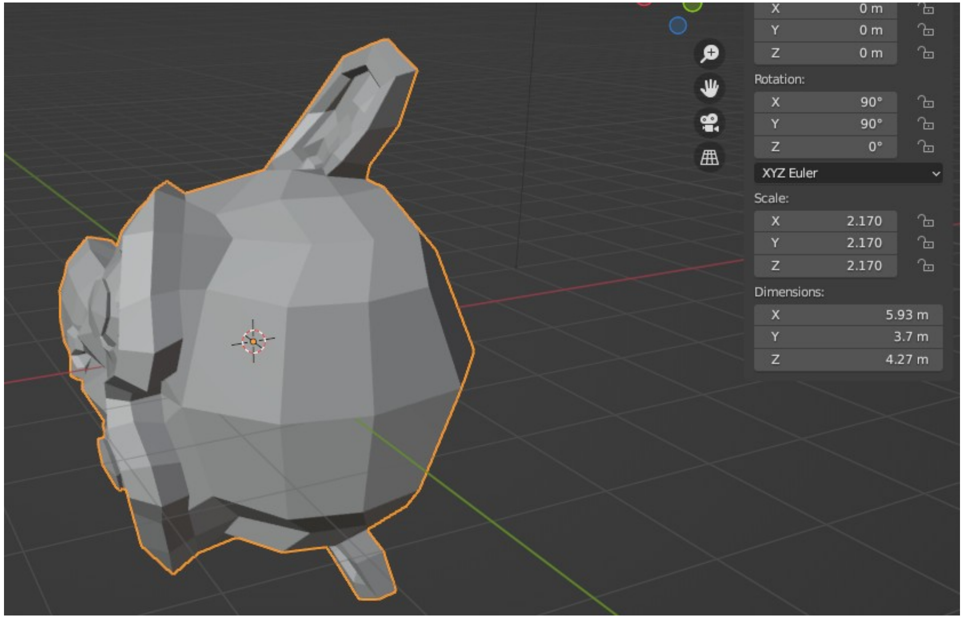






Attention : Euler Rxyz <> Ryxz





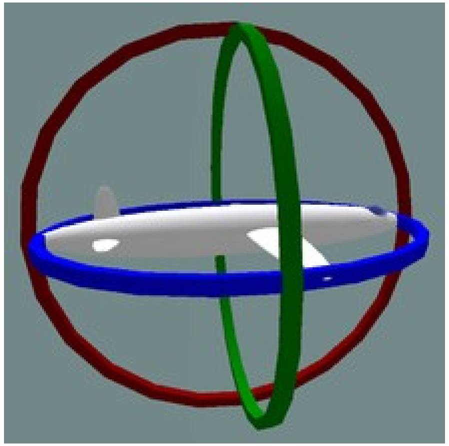
$$R_Y(90)\,R_X(90)$$

$$R_X(90)\,R_Y(90)$$

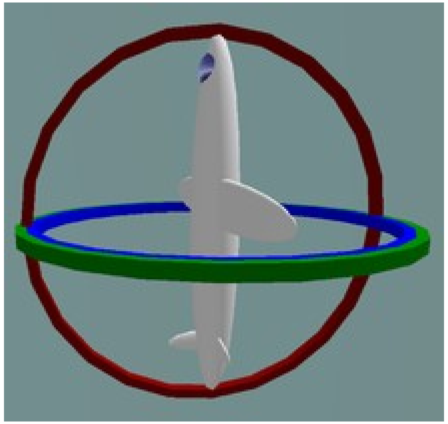
Gimbal Lock

- Blocage de cardan

Monter exemple
dans blender
de la roue (cylindre)
qui tourne dans blender



Situation normale : les trois cardans sont indépendants



Blocage de cardan : deux des trois cardans sont coplanaires, un degré de liberté est perdu

Gimbal Lock : exemple

$$R_{zyz}(\phi\theta\psi)=\begin{bmatrix} c\phi & -s\phi & 0 \\ s\phi & c\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}\begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$avec \quad \phi\in[-\pi,\pi],\theta\in[0,\pi] \quad et \quad \psi\in[-\pi,\pi]$$

Supposons que $\theta = 0$

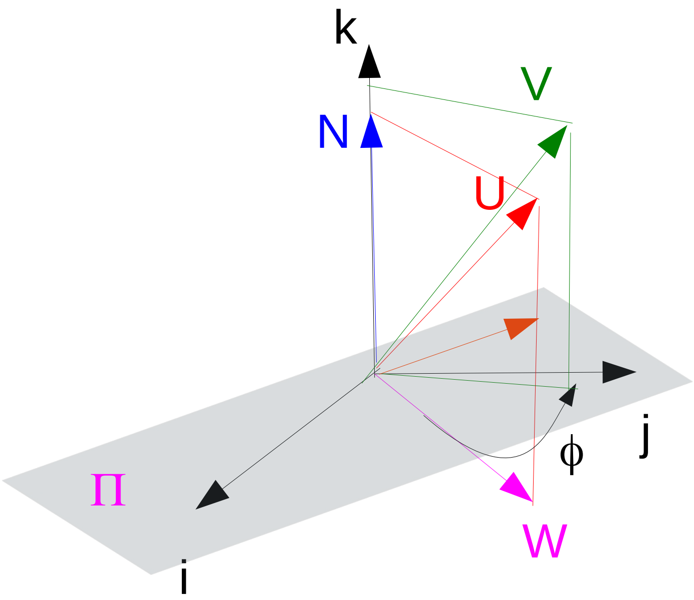
$$R_{zyz}(\phi 0\psi)=\begin{bmatrix} c\phi & -s\phi & 0 \\ s\phi & c\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}=\begin{bmatrix} c(\phi+\psi) & -s(\phi+\psi) & 0 \\ s(\phi+\psi) & c(\phi+\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

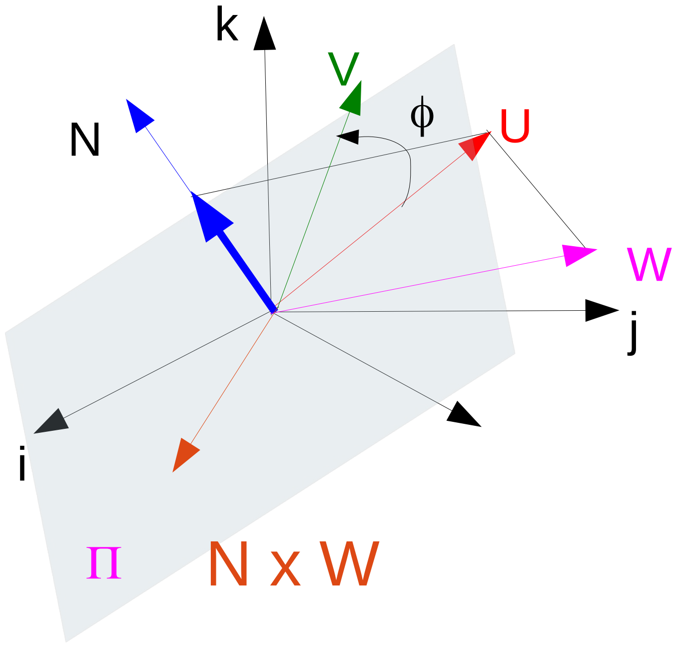
Changer les valeurs de ϕ et de ψ a le même effet : l'angle de rotation $\phi+\psi$ change, mais l'axe de rotation reste dans la direction Z. Un degré de liberté a été perdu.

Unicité ?

- Non :
  - $R(\Theta,\mathbf{u})= R(\Theta+2k\pi, \mathbf{u})$, $k \in N$
  - $R(-\Theta,-\mathbf{u})=R(\Theta,\mathbf{u})$
- R matrice orthogonale
  - $RR^t=I$ et $|R|=1$
  - $R(v)= R\mathbf{v}$
- L'ensemble des matrices de rotation (de taille fixée) :
  - forme le groupe des rotations ou groupe spécial orthogonal. Ici SO(3).
- $\forall R \in SO(3)$, rotation de Θ autour de $\mathbf{u}$ unitaire? Trouver Θ et $\mathbf{u}$

Formule de Rodrigues





$$\vec{V}=R\vec{U}=\begin{pmatrix}x'\\y'\\z'\end{pmatrix}=\begin{pmatrix}\cos\phi & -\sin\phi & 0\\ \sin\phi & \cos\phi & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}x\\y\\z\end{pmatrix}$$

$$\vec{V}'=(\vec{U}\cdot\vec{N})\vec{N}+(\cos\phi)\vec{W}+(\sin\phi)\vec{N}\times\vec{W}$$
$$\vec{V}'=(\vec{U}\cdot\vec{N})\vec{N}+(\cos\phi)(\vec{U}-(\vec{U}\cdot\vec{N})\vec{N})+(\sin\phi)\vec{N}\times\vec{U}$$
$$\vec{V}'=(\cos\phi)\vec{U}+(1-\cos\phi)(\vec{U}\cdot\vec{N})\vec{N}+(\sin\phi)(\vec{N}\times\vec{U})$$

Formule de Rodrigues

Dans le repère $(\vec{U}, \vec{N}, \vec{W})$

$$R = \cos\phi \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + (1 - \cos\phi) \begin{pmatrix} n_x^2 & n_x n_y & n_x n_z \\ n_y n_x & n_y^2 & n_y n_z \\ n_z n_x & n_z n_y & n_z^2 \end{pmatrix} + \sin\phi \begin{pmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{pmatrix}$$

Formule de Rodrigues

- R(Θ,**u**) = rotation de Θ autour de l'axe **u**
- Formule de Rodrigues

$$R(\theta,\mathbf{u})=I+sU+(1-c)U^2$$

$$U=\begin{pmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{pmatrix}$$

$$U\,\mathbf{v}=\mathbf{u}\times\mathbf{v}=\begin{bmatrix} u_y v_z-u_z v_y \\ u_z v_x-u_x v_z \\ u_x v_y-u_y v_x \end{bmatrix} \qquad et \quad U^2=\mathbf{u}\,\mathbf{u}^t-I$$

avec s = $\sin(\theta)$ et c = $\cos(\theta)$

Rq : $U^t=-U$

NB : U représente le produit vectoriel

Rodrigues (interprétation)

on note indifféremment $\boldsymbol{v}$ et $\vec{v}$

$\vec{v} = P(\vec{v}) + (I - P)(\vec{v})$

avec P projection de $\vec{v}$ sur $\vec{u}$: $P(\vec{v}) = \boldsymbol{u}(\boldsymbol{u}^t\boldsymbol{v}) = (\boldsymbol{u}\boldsymbol{u}^t)\boldsymbol{v}$

$$R(\theta,\boldsymbol{u})(\vec{v}) = R(\theta,\boldsymbol{u})(P(\vec{v})) + R(\theta,\boldsymbol{u})((I-P)(\vec{v}))$$

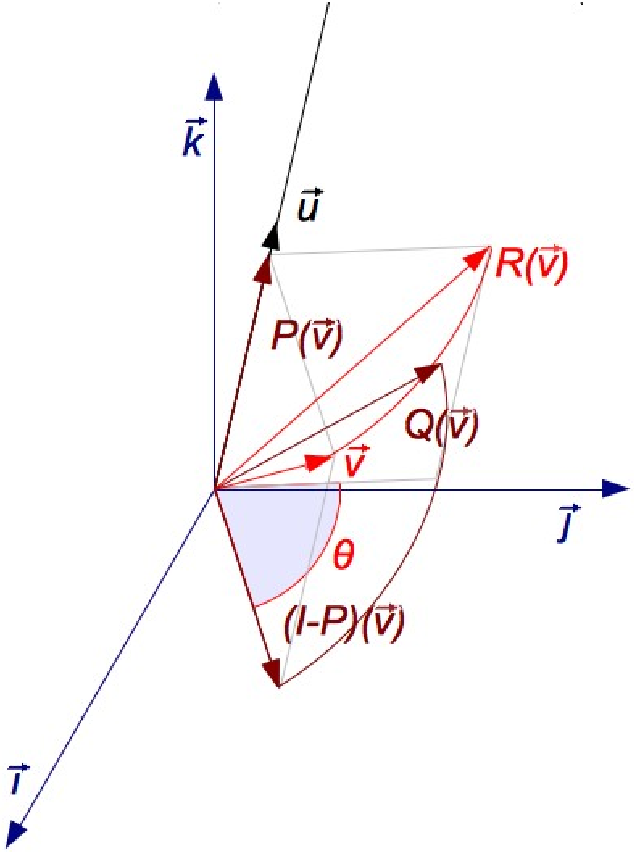
$$\mathit{soit}\quad Q(\vec{v}) = \vec{u}\times\vec{v} = \vec{u}\times(I-P)(\vec{v})$$

$$= (P(\vec{v}))$$

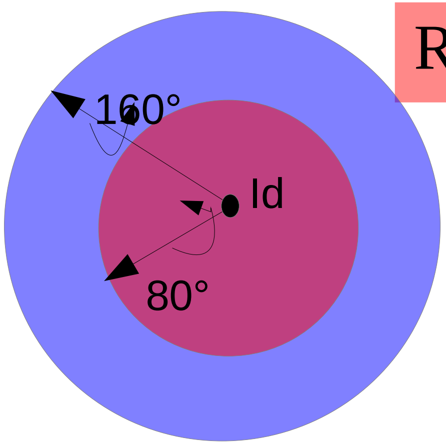
$$= c(I-P)(\vec{v}) + s\,\vec{Q}(v)$$

$$R(\theta,\boldsymbol{u})(\vec{v}) = (\boldsymbol{u}\boldsymbol{u}^t)\boldsymbol{v} + c(\boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^t)\boldsymbol{v} + s\,\boldsymbol{u}\times\boldsymbol{v}$$

$$R(\theta,\boldsymbol{u})(\vec{v}) = \boldsymbol{v} + s\,\boldsymbol{u}\times\boldsymbol{v} + (1-c)(\boldsymbol{u}\boldsymbol{u}^t - I)\boldsymbol{v}$$
$$= (I + sU + (1-c)U^2)\boldsymbol{v}$$



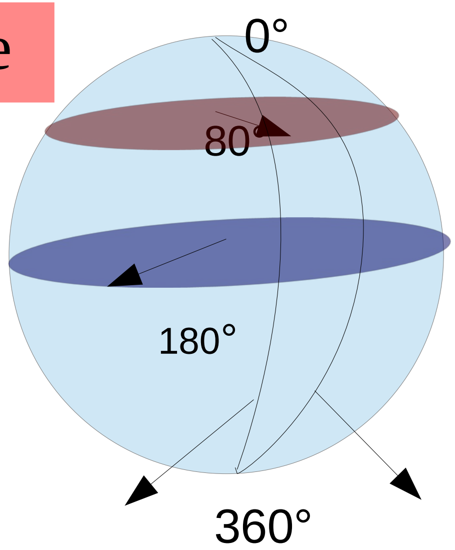
Espace des rotations : idée intuitive



Rotation en 3D = angle + axe

Axe = direction du vecteur

Angle = norme du vecteur



Angle=0 → rotation=Id → sphère = point
Angle + → sphères concentriques, R+
Angle > 180, R-
Angle = 360 → sphère = point

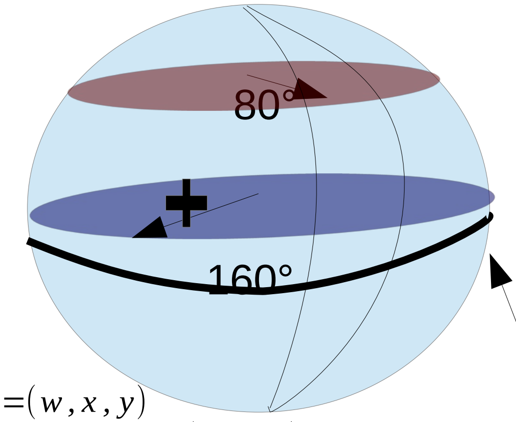
pôle Nord
les méridiens divergent
puis convergent
pôle Sud.

Pôle Nord
cercles concentriques R+
Parallèles R-
pôle Sud u

Méridiens ~ axes de rotations
distances au pôle Nord ~ angles

Paramétrisation

$(1,0,0)$



$P=(L,l)=(w,x,y)$

et $w^2+x^2+y^2=1$

$(-1,0,0)$

équateur : $w=0$ et $x^2+y^2=1$

Espace des rotations ~ sphère de dimension 3 dans un espace à 4 dimensions (hypersphère).

Une rotation 3D ~ un point de la sphère 4D
Sphère 3D ~ section de l'hypersphère
~ représenter les rotations d'axes dans un plan

Point sur l'hypersphère P = (Ɵ,φ,ψ) = angles d'Euler

NB : angle de la rotation = 2 * latitude (les points de l'équateur ~ rotations de 180°)
pôle Sud ~ rotation identité de 360°

Point sur la sphère P = (L,l)
Pôles dégénérés
=> P= (w,x,y)

Paramètres : 2 coordonnées (latitude,Longitude). Mais dégénérescence

$$P \approx \text{ rotation d'axe } \vec{V} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \text{ d'angle } \theta = 2\cos^{-1} w$$

$$w^2+x^2+y^2+z^2=1$$

Quaternions

Idée = analogie rotation dans le plan et complexes

Proposés par Hamilton

Les parametres d'Euler sont les composants du quaternion

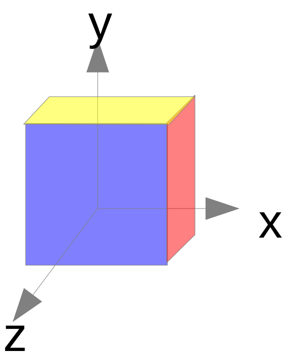
Quaternion Q : partie scalaire et partie vectorielle

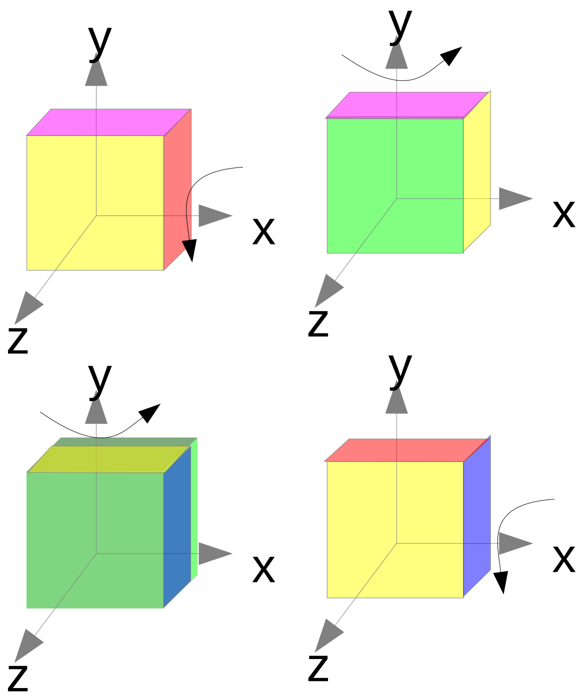
$$q = [w.1 + x.i + y.j + z.k] = [w + \vec{\mathbf{v}}] \in \mathbb{R} \times \mathbb{R}^3$$

$$avec \quad i^2 = j^2 = k^2 = -1$$

Non commutativité

partiellement anticommutative : 1 · i = i · 1 = i mais i · j = k et j · i = -k.





Opérations sur Quaternions

Addition: $q_1+q_2=[w_1+w_2,\ (v_1+v_2)]$

Multiplication: $q_1q_2=[w_1w_2-(v_1\cdot v_2),\ (w_1v_2+w_2v_1+v_1\times v_2)]$

Conjugué: $q^*=[w,\ v]^*=[w,\ -v]$

Norme: $\|q\|=\sqrt{(q.q^*)}=\sqrt{([w^2+(v\cdot v)])}$

Inverse: $q^{-1}=\dfrac{q^*}{\|q\|^2}$

Quaternions = Rotations

la représentation d'un point cartésien dans l'espace des quaternions :

=**un quaternion pur** (pas de partie réelle) :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow p = [0, (x, y, z)]$$

Soit **q** un quaternion *unitaire* (i.e. ||**q**|| = *1*).

La *conjugaison* de **p** par **q** est:

$$p' = \mathbf{q}p\mathbf{q}^{-1} = \mathbf{q}p\mathbf{q}^{*}$$

Correspondance entre quaternion unitaire et rotation vectorielle

On peut démontrer que le vecteur V'

$$\vec{V}\,' = R_{[2\phi,\vec{N}]}(\vec{V})$$

s'écrit

$$[O,\vec{V}\,'] = [0, R_{[2\phi,\vec{N}]}(\vec{V})] = [\cos\phi, \sin\phi\vec{N}] \times [0,\vec{V}] \times [\cos\phi, -\sin\phi\vec{N}]$$

=> la rotation est représentée par le quaternion

$$R_{[2\phi,\vec{N}]} \Leftrightarrow [\cos\phi, \sin\phi\vec{N}]$$

Quaternions = Rotations

D'où $\mathbf{p}' = [0, \mathbf{v}']$ $\quad \|\mathbf{p}\| = \|\mathbf{p}'\|$

Et si $q = \left[\cos\frac{\theta}{2},\ v \sin\frac{\theta}{2}\right]$

$\mathbf{p}' = \mathbf{q}\mathbf{p}\mathbf{q}^*$

alors l'effet de la conjugaison sur **p** est de le tourner dans le sens trigonométrique selon l'axe v de θ degres.

=> tout quaternion unitaire représente une rotation selon un axe.

Note : la rotation **q** est la même que **-q** ⇒ *redondance*

Quaternions - Rotations

Q quaternion qcq

$$Soit\ Q=[a,\vec{V}],\quad q=\|Q\|\ \ et\ \ v=\|\vec{v}\|$$

$$Q=q.\left[\frac{1}{q}a,\frac{1}{q}\vec{V}\right]=q.\left[\frac{1}{q}a,\frac{v}{q}\frac{1}{v}\vec{V}\right]\quad si\ \ v\neq 0\ \ et\ \ donc\ \ q\neq 0$$

$$comme\ \ \frac{1}{v}\vec{V}\ \ \ est\ \ normé\ \ q^2=a^2+v^2\equiv\left(\frac{a}{q}\right)^2+\left(\frac{v}{q}\right)^2=1$$

Donc il existe Φ tel que

$$\cos\phi=\frac{a}{q}\ et\ \sin\phi=\frac{v}{q}$$

$$donc\quad Q=[a,\vec{V}]=\left[q\cos\phi+q\sin\phi\,\vec{U}\right]$$

$$\vec{U}=\frac{1}{v}\vec{V}$$

On peut lui associé une rotation

Quaternions = Rotations

Exemple: rotation du point P = [x, y, z] selon le vecteur **v** de θ degres.

1) representation en quaternion **p** du point P :

$$p=[0,(x,y,z)]$$

2) Création du quaternion **q** pour la rotation:

$$q=\left[\cos\frac{\theta}{2},\ \frac{v}{\|v\|}\cdot\sin\frac{\theta}{2}\right]$$

3) Conjugaison de **p** par **q**:

$$p'=\boldsymbol{qpq}^{*}$$

Matrice de Rotation ↔ Conversion en Quaternion

$$q=(w\,,\ x\,,\ y\,,\ z)\rightarrow R=\begin{bmatrix} w^2+x^2-y^2-z^2 & 2\,xy-2\,wz & 2\,xz+2\,wy \\ 2\,xy+2\,wz & w^2-x^2+y^2-z^2 & 2\,yz-2\,wx \\ 2\,xz-2\,wy & 2\,yz+2\,wx & w^2-x^2-y^2+z^2 \end{bmatrix}$$

$$R=\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}\rightarrow q=[w\,,\ (x\,,y\,,z)]\ \text{where}\ \begin{cases} w=\sqrt{\dfrac{1+m_{11}^2+m_{22}^2+m_{33}^2}{4}} \\ x=\dfrac{m_{32}-m_{23}}{4\,w} \\ y=\dfrac{m_{13}-m_{31}}{4\,w} \\ z=\dfrac{m_{21}-m_{12}}{4\,w} \end{cases}$$

“Deconstruction” d’un Quaternion

Etant donné un quaternion unitaire arbitraire on peut retrouver l’axe et l’angle de la rotation associée :

$$q=\left(w\,,\ \ v\right)\Rightarrow\begin{cases}\text{axe de rotation}\quad u=\dfrac{v}{\|v\|}\\[2ex]\text{angle de rotation}\quad \theta=\cos^{-1}\left[w^2-\|v\|^2\right]\end{cases}$$

Normalisation

Avec des rotations successives, accumulation d’erreurs

→ quaternions non-unitaires

→ on normalise

$$q' = \frac{q}{\|q\|}$$

Normalisation de Matrices : plus difficile

$$R' = R\left(R^T R\right)^{\frac{1}{2}}$$

Composition de Rotations par Quaternions

Comme avec les matrices, les rotations par quaternion sont composées de rotations de base :

$$\text{soit } \boldsymbol{p}' = \boldsymbol{q}_1 \boldsymbol{p} \boldsymbol{q}_1^* \text{ et } \boldsymbol{p}'' = \boldsymbol{q}_2 \boldsymbol{p}' \boldsymbol{q}_2^*$$

$$\Rightarrow \boldsymbol{p}'' = \boldsymbol{q}_2 (\boldsymbol{q}_1 \boldsymbol{p} \boldsymbol{q}_1^*) \boldsymbol{q}_2^*$$

$$= (\boldsymbol{q}_2 \boldsymbol{q}_1) \boldsymbol{p} (\boldsymbol{q}_1^* \boldsymbol{q}_2^*)$$

propriété de la conjugaison

$$\text{mais } (\boldsymbol{ab})^* = \boldsymbol{b}^* \boldsymbol{a}^*$$

$$\text{donc } \boldsymbol{p}'' = (\boldsymbol{q}_2 \boldsymbol{q}_1) \boldsymbol{p} (\boldsymbol{q}_2 \boldsymbol{q}_1)^*$$

$$\text{soit } \boldsymbol{p}'' = \boldsymbol{r} \boldsymbol{p} \boldsymbol{r}^* \quad \text{avec } \boldsymbol{r} = \boldsymbol{q}_2 \boldsymbol{q}_1$$

Noter l'ordre ($\mathbf{q}_2$ après $\mathbf{q}_1$)

Avantages des quaternions

- Représentation compacte d'une rotation en 3D
- Permet d'éviter le blocage de Cardan
- Permet de définir facilement
  - l'interpolation linéaire sphérique:

$$\mathrm{SLERP}(q_0, q_1, t) = q_0 (q_0^{-1} q_1)^t$$

  - où la vitesse est uniforme sur l'arc de cercle, contrairement à une interpolation linéaire
  - Interpolations sphériques progressives et contrôlées.

Avantages des Quaternions pour les Rotations

- Moins de *stockage* (4 réels vs. 16 réels)
- Construction de *rotations arbitraires* selon des vecteurs plus facile.
- *Interpolation* facile pour des animations.

Operation	Matrices	Quaternions
Stockage	**9**	**4**
Transformation	**9M 6A**	**15M 15A**
Composition	**27M 18A**	**16M 12A**
Normalisation	**compliquée**	**8M 3A 1sqrt**

Inconvénients

- Rotation autour d’un axe passant par (0,0,0)
- Pour un axe de direction u passant par C
  - R(p) =c+q(p-c)q*
- Composition de transformations translation rotation : pas pratique
- **Solution : les quaternions duaux :**

Kavan, L., Collins, S., O’Sullivan, C., & Zara, J. (2006). Dual quaternions for rigid transformation blending. Trinity College Dublin, Tech. Rep. TCD-CS-2006-46.

Kavan, L., Collins, S., Žára, J., & O'Sullivan, C. (2007, April). ***Skinning with dual quaternions.*** *In Proceedings of the 2007 symposium on Interactive 3D graphics and games (pp. 39-46).*

Les quaternions pour les rotations avec GLM

```
#include <glm/gtc/quaternion.hpp>
```

```
using namespace glm;

quat q0, q1; // déclaration de quaternions
// Création d’un quaternion unitaire représentant une rotation
// d’axe (1,1,0) et d’angle 90 degrés
q0=angleAxis((float)radians(90.), vec3(1.,1.,0.));
...
// composition rotations = multiplication des quaternions
quad  q= q0*q1 ; // = R0 o R1
```

Les quaternions pour les rotations avec GLM

Utilisation **sans** shader

```
// transformation du quaternion unitaire
// en matrice de rotation

// Conversion en matrice homogène Glm
mat4 myMatriceRotation ;
myMatriceRotation  = mat4_cast(q0);

// application de la matrice de rotation
glMultMatrixf(&myMatriceRotation[0][0]);
```

Les quaternions pour les rotations avec GLM

```
Utilisation avec shader
```

```
// GLSL pas de quaternion ;(
// 1) on doit faire à la mimine avec des vec4 + implémentation
// des opérations


// 2) en C++ on convertit le quaternion en matrice MODEL
// et on fait comme d'hab.
```

Compléments / précisions

- https://www.3dgep.com/understanding-quaternions/
- https://fr.wikipedia.org/wiki/Quaternions_et_rotation_dans_l%27espace