1 Szenario

Im letzten Tutorial wurde nicht darauf eingegangen wie man aus einer gegebenen Matrix herausfinden kann, um welche Rotationsachse(normiert) und um welchen Winkel bei dieser Matrix gedreht wird.Sicher funktioniert es auf die Weise,dass man die Matrix zunächst in ein Quaternion umwandelt und schließlich aus dem Quaternion die gewünschten Informationen zieht.Mit Hilfe der Bibliothek könnte dies sehr bequem gelöst werden.

```
Matrix3D M,Id;
Vector3D axis;
double angle;
M = Id.rotateX(M_PI/4);
Quaternion q = M.toQuaternion();
axis = q.getVec()
angle = q.getAngle();
```

Doch es gibt auch eine andere Möglichkeit dieses Ergebnis ohne Wissen über ein Quaternion zu haben zu erlangen. Es lässt sich zeigen, dass man durch geschicktes hintereinanderausführen eine Rotationsmatrix $R(\phi, \vec{u})$ herleiten kann, die genau diese Vorraussetzungen erfüllt. Da eine genaue Herleitung den Rahmen dieses Tutorials sprengen würde, wird hier nur die Matrix angegeben.

$$R(\phi, \vec{u}) =$$

$$\begin{pmatrix} u_x^2(1-\cos\phi)+\cos\phi & u_xu_y(1-\cos\phi)-u_z\sin\phi & u_xu_z(1-\cos\phi)+u_y\sin\phi \\ u_xu_y(1-\cos\phi)+u_z\sin\phi & u_y^2(1-\cos\phi)+\cos\phi & u_yu_z(1-\cos\phi)-u_x\sin\phi \\ u_xu_z(1-\cos\phi)-u_y\sin\phi & u_yu_z(1-\cos\phi)+u_x\sin\phi & u_z^2(1-\cos\phi)+\cos\phi \end{pmatrix}$$

2 Bestimmung des Winkels

Ziel ist es nun aus dieser Matrix den Vektor $\vec{u}$ und den Winkel ϕ zu gewinnen.Für den Winkel betrachtet man die Spur der Matrix (Summe der hauptdiagonalen)

$$spur(R(\phi, \vec{u})) = (u_x^2 + u_y^2 + u_z^2)(1-\cos\phi) + 3\cos\phi = 1 + 2\cos\phi$$

Der obige Schritt folgt sofort, wenn man berücksichtigt, dass $\vec{u}$ normiert ist.Daraus folgt für ϕ:

$$\phi = \arccos\left(\frac{spur(R(\phi, \vec{u}) - 1)}{2}\right)$$

3 Bestimmung der Achse

Was noch fehlt, ist die Bestimmung der Achse aus der Matrix R.Hier fällt jedoch auf:

$$\begin{aligned}
R_{32} - R_{23} &= 2u_x \sin\phi \\
R_{13} - R_{31} &= 2u_y \sin\phi \\
R_{21} - R_{12} &= 2u_z \sin\phi
\end{aligned}$$

Woraus wir nun letztendlich auf die Achse $\vec{u}$ schließen können:

$$\vec{u} = \frac{1}{2\sin\phi} \begin{pmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{pmatrix}$$

natürlich lässt sich auch dies komfortabel mit Hilfe der Bibliothek lösen:

```
Matrix3D Id,R_x;
double angle = M_PI/4;
R_x = Id.rotateX(angle); //Beispielmatrix

Vector3D u = R_x.getVec();
double rotAngle = R_x.getAngle();
```