Procedura 8 e 9 in cascata

```
Maxima 5.44.0 http://maxima.sourceforge.net
using Lisp SBCL 2.0.0
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
```

```
(%i1) skewMatrix(x):=block([res],
                           S:ident(3),
                           for i:1 thru 3 do
                           (
                           for j:1 thru 3 do
                              (
                                if i=j
                                   then S[i][j]:0
                                elseif j>i
                                   then (
                                  temp:(-1)^(j-i)*x[3-remainder(i+j,3)],
                                         S[i][j]:temp,
                                         S[j][i]:-temp

                                          )
                                )
                            ),
                            res:S
                            )
```

(%o1) $\mathrm{skewMatrix}(x) := \mathbf{block}\ ([\mathrm{res}], S: \mathrm{ident}(3), \mathbf{for}\ i\ \mathbf{thru}\ 3\ \mathbf{do}\ \mathbf{for}\ j\ \mathbf{thru}\ 3\ \mathbf{do}\ \mathbf{if}\ i = j\ \mathbf{then}\ (S_i)_j: 0\ \mathbf{elseif}\ j > i\ \mathbf{then}\ (\mathrm{temp}: (-1)^{j-i}\, x_{3-\mathrm{remainder}(i+j,3)}, (S_i)_j: \mathrm{temp}, (S_j)_i: -\mathrm{temp}), \mathrm{res}: S)$

```
(%i2) rodriguez(y):=block([res],
                          I:ident(3),
                          S:skewMatrix(y),
                          res:I+S.S*(1-cos(theta))+S*sin(theta)
                          )
```

(%o2) $\mathrm{rodriguez}(y) := \mathbf{block}\ ([\mathrm{res}], I: \mathrm{ident}(3), S: \mathrm{skewMatrix}(y), \mathrm{res}: I + S \cdot S\, (1 - \cos(\vartheta)) + S \sin(\vartheta))$

```
(%i3) isRotation(M):=block([res],
                           I:ident(3),
                           MMT:trigsimp(expand(M.transpose(M))),
                           detM:trigsimp(expand(determinant(M))),

                           if MMT=I  and detM=1
                              then(

                                   return(res:1)
                                   )

                           else(

                                 res: "R is not rotation matrix"
                                 )
                           )
```

(%o3) $\mathrm{isRotation}(M) := \mathbf{block}\ ([\mathrm{res}], I: \mathrm{ident}(3), \mathrm{MMT}: \mathrm{trigsimp}(\mathrm{expand}(M \cdot \mathrm{transpose}(M))),$

detM: trigsimp(expand(determinant(M))), **if** MMT $= I \wedge$ detM $= 1$ **then** return(res: 1) **else** res: R is not rotation matrix)

```
(%i4) axes(M):=block([res],
                       columns:transpose(M),
                      res:zeromatrix(3,1),
                       for i:1 thru length(columns) do(
                         if(columns[i][1]# 0 or columns[i][2]#0 or
      columns[i][3]#0)
                            then ( return(m: transpose(columns[i])))

                       ),res:m

                       )
```

(%o4) axes(M) := **block** ([res], columns: transpose(M), res: zeromatrix$(3,1)$, **for** i **thru** length(columns) **do if** $(\text{columns}_i)_1 \neq 0 \vee (\text{columns}_i)_2 \neq 0 \vee (\text{columns}_i)_3 \neq 0$ **then** return(m: transpose(columns_i)), res: m)

```
(%i5) skewMatrix(x):=block([res],
                           S:ident(3),
                           for i:1 thru 3 do
                           (
                           for j:1 thru 3 do
                              (
                                if i=j
                                   then S[i][j]:0
                                elseif j>i
                                   then (
                                  temp:(-1)^(j-i)*x[3-remainder(i+j,3)][1],
                                         S[i][j]:temp,
                                         S[j][i]:-temp

                                          )
                                )
                            ),
                            res:S
                            )
```

(%o5) skewMatrix(x) := **block** ([res], S: ident(3), **for** i **thru** 3 **do for** j **thru** 3 **do if** $i = j$ **then** $(S_i)_j$: 0 **elseif** $j > i$ **then** (temp: $(-1)^{j-i}\,(x_{3-\text{remainder}(i+j,3)})_1$, $(S_i)_j$: temp, $(S_j)_i$: $-$temp), res: S)

```
(%i6) sinRotation(skewMat,RRT2):=block([res],

                               for i:1 thru 3 do(
                                  for j:1 thru 3 do(
                                     a:skewMat[i][j],

                                     if a# 0
                                        then (b:RRT2[i][j],

                                              return(

                                         value:b/a
                                         ))
                                    )
                                 ),res:value


                               )
```

(%o6) $$\mathrm{sinRotation}(\mathrm{skewMat},\mathrm{RRT2}):=\mathbf{block}\left([\mathrm{res}],\mathbf{for}\ i\ \mathbf{thru}\ 3\ \mathbf{do}\ \mathbf{for}\ j\ \mathbf{thru}\ 3\ \mathbf{do}\left(a{:}\,(\mathrm{skewMat}_i)_j,\mathbf{if}\ a\neq 0\ \mathbf{then}\left(b{:}\,(\mathrm{RRT2}_i)_j,\mathrm{return}\left(\mathrm{value}{:}\,\frac{b}{a}\right)\right)\right),\mathrm{res}{:}\,\mathrm{value}\right)$$

```
(%i7) cosRotation(x,y):=block([res],

                               for i:1 thru 3 do(
                                  for j:1 thru 3 do(
                                     c:x[i][j],
                                     if(c#0) then(
                                                d:y[i][j],

                                                return(t:(c-d)/c))
                                  )
                               ),
                                res:t

                               )
```

(%o7) $$\mathrm{cosRotation}(x,y):=\mathbf{block}\left([\mathrm{res}],\mathbf{for}\ i\ \mathbf{thru}\ 3\ \mathbf{do}\ \mathbf{for}\ j\ \mathbf{thru}\ 3\ \mathbf{do}\left(c{:}\,(x_i)_j,\mathbf{if}\ c\neq 0\ \mathbf{then}\left(d{:}\,(y_i)_j,\mathrm{return}\left(t{:}\,\frac{c-d}{c}\right)\right)\right),\mathrm{res}{:}\,t\right)$$

```
(%i8) degree(v,M):=block([sinR,cosR,res],
                           S:skewMatrix(v),
                           I:ident(3),
                           RRsin:trigsimp((M-transpose(M))*1/2),
                           RRcos:trigsimp(((M+transpose(M))*1/2)-I),
                           sinR:sinRotation(S,RRsin),

                           SS:S.S,
                           cosR:cosRotation(SS,RRcos),
                           res:atan2(expand(sinR),expand(cosR))

                             )
```

(%o8) $\mathrm{degree}(v,M):=\mathbf{block}\Big([\mathrm{sinR},\mathrm{cosR},\mathrm{res}],S\colon\mathrm{skewMatrix}(v),I\colon\mathrm{ident}(3),\mathrm{RRsin}\colon \mathrm{trigsimp}\Big(\frac{(M-\mathrm{transpose}(M))\,1}{2}\Big),\mathrm{RRcos}\colon\mathrm{trigsimp}\Big(\frac{(M+\mathrm{transpose}(M))\,1}{2}-I\Big),\mathrm{sinR}\colon \mathrm{sinRotation}(S,\mathrm{RRsin}),\mathrm{SS}\colon S\cdot S,\mathrm{cosR}\colon\mathrm{cosRotation}(\mathrm{SS},\mathrm{RRcos}),\mathrm{res}\colon\mathrm{atan2}(\mathrm{expand}(\mathrm{sinR}), \mathrm{expand}(\mathrm{cosR}))\Big)$

```
(%i9) axesDegree(R):=block([v,theta,res],
                           isRot:isRotation(R),
                           if isRot=1 then (
                              I:ident(3),
                              adjR:adjoint(I-R),
                              v:axes(adjR),
                              vNorm:v/sqrt(v.v),
                              theta:degree(vNorm,R),
                              print("Axe, degree"),
                              res:[vNorm,theta]



                           )
                           else res:"R is not rotation matrix"



                          )
```

(%o9) $\mathrm{axesDegree}(R):=\mathbf{block}\Big([v,\vartheta,\mathrm{res}],\mathrm{isRot}\colon\mathrm{isRotation}(R),\ \mathbf{if}\ \mathrm{isRot}=1\ \mathbf{then}\ \Big(I\colon\mathrm{ident}(3), \mathrm{adjR}\colon\mathrm{adjoint}(I-R),v\colon\mathrm{axes}(\mathrm{adjR}),\mathrm{vNorm}\colon\frac{v}{\sqrt{v\cdot v}},\vartheta\colon\mathrm{degree}(\mathrm{vNorm},R),\mathrm{print}(\mathrm{Axe,\ \ degree\ \ }),\mathrm{res}\colon [\mathrm{vNorm},\vartheta]\Big)\ \mathbf{else}\ \mathrm{res}\colon \text{R is not rotation matrix}\Big)$

Ruoto attorno all'asse x di $\frac{2\pi}{3}$. La corrispettiva matrice di rotazione viene calcolata tramite la procedura di Rodriguez. (procedura 8)

```
(%i10) v:1/sqrt(3)*matrix([1,-1,1]);
```

(%o10) $\begin{pmatrix} \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{pmatrix}$

$R_x(\theta)$ descrive una matrice di rotazione dipendente dal parametro ϑ lungo l'asse di rotazione x.

```
(%i11) R[x](theta):=rodriguez(transpose(v))
```

(%o11) $R_x(\vartheta):=\mathrm{rodriguez}(\mathrm{transpose}(v))$

Assegno l'angolo di rotazione di $\frac{2\,\pi}{3}$ alla matrice di rotazione descritta sopra.

```
(%i12) R:R[x](2*%pi/3)
```

(%o12) $\begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{pmatrix}$

A questo punto si verifica che la matrice di rotazione ottenuta abbia lo stesso asse e angolo tramite la procedura 9.

```
(%i13) axesDegree(R);
```

Axe, degree

(%o13) $$\left[\begin{pmatrix} \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{pmatrix}, \frac{2\,\pi}{3}\right]$$

(%i14)

La procedura axesDegree ritorna effettivamente l'asse di rotazione $v = \frac{1}{\sqrt{3}}\begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$ e l'angolo di rotazione $\frac{2\pi}{3}$.