Procedura 6: calcolo matrici di rotazioni tramite autovalori/autovettori

Scrivere una procedura che utilizzando gli autovalori/autovettori calcoli la matrice di rotazione intorno all'asse rappresentata dal versore v, di un angolo ϑ:

$$S(v) = V.\Lambda.V^{-1}$$

$$R_v(\theta) = V.e^{\Lambda\theta}.V^{-1}$$

In particolare, i passi da seguire sono i seguenti:

1) $S(v) = V.\Lambda.V^{-1}$ con V:= matrice degli auto vettori come colonne

Λ:= matrice diagonale con gli autovali nella diagonale principale

2) $e^{S(v)\theta} = Ve^{\Lambda\theta}V^{-1}$ con $e^{\Lambda\theta} := \text{matrice diagonale}$

3) $R_v(\theta) = V.e^{\Lambda\theta}.V^{-1}$

La funzione matRot prende in input una matrice S(v) e ne calcola autovalori, disposti nella diagonale principale della matrice Λ, e autovettori disposti come colonne nella matrice V.
In seguito, la variabile matExp salva la matrice esponenziale del punto 2).
Infine restituisce in output la matrice di rotazione del punto 3):

$$R_v(\theta) = V.e^{\Lambda\theta}.V^{-1}$$

```
(%i1) matRot(M):=block([V,matExp,eigVect,Lambda,res],
                  eigVect:eigenvectors(M),
                  V:transpose(matrix(eigVect[2][1][1],eigVect[2][2][1],
      eigVect[2][3][1])),
                  Lambda:matrix([eigVect[1][1][1],0,0],[0,eigVect[1][1][2],0],[0,
      0,eigVect[1][1][3]]),
                  matExp:matrixexp(Lambda*theta),
                  res:expand(demoivre(expand(V.matExp.invert(V))))
                  )
```

(%o1) $\mathrm{matRot}(M) := \mathbf{block}\left([V, \mathrm{matExp}, \mathrm{eigVect}, \Lambda, \mathrm{res}], \mathrm{eigVect}: \mathrm{eigenvectors}(M), V: \mathrm{transpose}\left(\begin{pmatrix} (\mathrm{eigVect}_2)_1 & 1 \\ (\mathrm{eigVect}_2)_2 & 1 \\ (\mathrm{eigVect}_2)_3 & 1 \end{pmatrix}\right), \Lambda: \begin{pmatrix} ((\mathrm{eigVect}_1)_1)_1 & 0 & 0 \\ 0 & ((\mathrm{eigVect}_1)_1)_2 & 0 \\ 0 & 0 & ((\mathrm{eigVect}_1)_1)_3 \end{pmatrix}, \mathrm{matExp}: \mathrm{matrixexp}(\Lambda\,\vartheta), \mathrm{res}: \mathrm{expand}(\mathrm{demoivre}(\mathrm{expand}(V\cdot\mathrm{matExp}\cdot\mathrm{invert}(V))))\right)$

La funzione rotEigen(k) riceve in input il versore e_x, e_y, e_z ed invoca matRot(M) per il calcolo della matrice di rotazione corrispondente. Restituisce in output l'effettiva matrice di rotazione corrispondente al versoe in input.
Altrimenti, se il versore inserito è diverso da e_x, e_y, e_z restituisce errore.

```
(%i2) rotEigen(k):=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)*k[3-remainder(i+j,3)],
                                       S[i][j]:temp,
                                       S[j][i]:-temp

                                        )
                              )
                          ),
                     res:matRot(S)
                     )
```

(%o2) $\mathrm{rotEigen}(k) := \textbf{block}\ ([\mathrm{res}], S\colon \mathrm{ident}(3), \textbf{for}\ i\ \textbf{thru}\ 3\ \textbf{do}\ \textbf{for}\ j\ \textbf{thru}\ 3\ \textbf{do}\ \textbf{if}\ i = j\ \textbf{then}\ (S_i)_j\colon 0\ \textbf{elseif}\ j > i\ \textbf{then}\ (\mathrm{temp}\colon (-1)^{j-i} k_{3-\mathrm{remainder}(i+j,3)}, (S_i)_j\colon \mathrm{temp}, (S_j)_i\colon -\mathrm{temp}), \mathrm{res}\colon \mathrm{matRot}(S))$

Matrice di rotazione $R_x(\theta)$:

`(%i3) R[x](theta):=rotEigen([1,0,0]);`

(%o3) $R_x(\vartheta) := \mathrm{rotEigen}([1,0,0])$

`(%i4) R[x](theta);`

`Proviso: assuming 4*theta # 0`(%o4) $\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\vartheta) & -\sin(\vartheta) \\ 0 & \sin(\vartheta) & \cos(\vartheta) \end{pmatrix}$

Matice di rotazione $R_y(\theta)$:

`(%i5) R[y](theta):=rotEigen([0,1,0]);`

(%o5) $R_y(\vartheta) := \mathrm{rotEigen}([0,1,0])$

`(%i6) R[y](theta);`

`Proviso: assuming 4*theta # 0`

(%o6) $\begin{pmatrix} \cos(\vartheta) & 0 & \sin(\vartheta) \\ 0 & 1 & 0 \\ -\sin(\vartheta) & 0 & \cos(\vartheta) \end{pmatrix}$

Matrice di rotazione $R_z(\theta)$:

`(%i7) R[z](theta):=rotEigen([0,0,1]);`

(%o7) $R_z(\vartheta) := \mathrm{rotEigen}([0,0,1])$

`(%i8) R[z](theta);`

`Proviso: assuming 4*theta # 0`(%o8) $\begin{pmatrix} \cos(\vartheta) & -\sin(\vartheta) & 0 \\ \sin(\vartheta) & \cos(\vartheta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$

(%i9)