RigidMotion Documentation

Release 3.1

/ELSA/MU-12016/V3.1

May 28, 2020

CONTENTS

CHAPTER

ONE

PREAMBLE

RigidMotion enables to define or compute rigid motions for arrays (as defined in Converter documentation) or for CGSN/Python trees (pyTrees).

This module is part of Cassiopee, a free open-source pre- and post-processor for CFD simulations.

For use with the array interface, you have to import RigidMotion module:

```
import RigidMotion
```

For use with the pyTree interface:

```
import RigidMotion.PyTree as RigidMotion
```

CHAPTER

TWO

LIST OF FUNCTIONS

– Prescribed motions

RigidMotion.setPrescribedMotion1	
RigidMotion.setPrescribedMotion2	
RigidMotion.setPrescribedMotion3	

– General functions

RigidMotion.evalPosition(array, time, F)	Move the mesh with defined motion function to time t.

CHAPTER

THREE

CONTENTS

RigidMotion.**setPrescribedMotion1**(*a*, *motionName*, *tx="0"*, *ty="0"*, *tz="0"*, *cx="0"*, *cy="0"*, *cz="0"*, *ex="0"*, *ey="0"*, *ez="0"*, *angle="0"*)

Set a precribed motion defined by a translation of the origin (tx,ty,tz), the center of a rotation (cx,cy,cz), the second point of the rotation axis (ex,ey,ez) and the rotation angle in degrees. They can depend on time {t}.

Parameters

- **a** (`[array, list of arrays] or [pyTree, base, zone, list of zones]`) – Input data
- **tx** (`string`) – translation in x motion string
- **ty** (`string`) – translation in y motion string
- **tz** (`string`) – translation in z motion string
- **cx** (`string`) – rotation center x coordinate motion string
- **cy** (`string`) – rotation center y coordinate motion string
- **cz** (`string`) – rotation center z coordinate motion string
- **ex** (`string`) – rotation axis x coordinate motion string
- **ey** (`string`) – rotation axis y coordinate motion string
- **ez** (`string`) – rotation axis z coordinate motion string
- **angle** (`string`) – rotation angle motion string

Example of use:

- Set a prescribed motion of type 1 (pyTree):

```
# - setPrescribedMotion1 (pyTree) -
# Motion defined by time string
import RigidMotion.PyTree as R
```

(continues on next page)

(continued from previous page)

```
import Converter.PyTree as C
import Geom.PyTree as D

a = D.sphere((1.2,0.,0.), 0.2, 30)
a = R.setPrescribedMotion1(a, 'trans', tx="{t}")

C.convertPyTree2File(a, 'out.cgns')
```

RigidMotion.**setPrescribedMotion2**(*a*, *motionName*, *transl_speed*, *psi0*, *pis0_b*, *alp_pnt*, *alp_vct*, *alp0*, *rot_pnt*, *rot_vct*, *rot_omg*, *del_pnt*, *del_vct*, *del0*, *delc*, *dels*, *bet_pnt*, *bet_vct*, *bet0*, *betc*, *bets*, *tet_pnt*, *tet_vct*, *tet0*, *tetc*, *tets*, *span_vct*, *pre_lag_pnt*, *pre_lag_vct*, *pre_lag_ang*, *pre_con_pnt*, *pre_con_vct*, *pre_con_ang*)

Set a precribed motion defined by a elsA rotor motion. Arguments are identical to elsA rotor motion.

Parameters

- **a** (`[array, list of arrays] or [pyTree, base, zone, list of zones]`) – Input data
- **tx** (`string`) – translation in x motion string
- **ty** (`string`) – translation in y motion string
- **tz** (`string`) – translation in z motion string
- **cx** (`string`) – rotation center x coordinate motion string
- **cy** (`string`) – rotation center y coordinate motion string
- **cz** (`string`) – rotation center z coordinate motion string
- **ex** (`string`) – rotation axis x coordinate motion string
- **ey** (`string`) – rotation axis y coordinate motion string
- **ez** (`string`) – rotation axis z coordinate motion string
- **angle** (`string`) – rotation angle motion string

Example of use:

- Set a prescribed motion of type 2 (pyTree):

```
# - setPrescribedMotion2 (pyTree) -
# Motion defined by a Cassiopee Solver rotor motion
```

(continues on next page)

(continued from previous page)

```
import RigidMotion.PyTree as R
import Converter.PyTree as C
import Geom.PyTree as D

a = D.sphere((1.2,0.,0.), 0.2, 30)
a = R.setPrescribedMotion2(a, 'rotor', transl_speed=(0.1,0,0), rot_omg=1.)

C.convertPyTree2File(a, 'out.cgns')
```

RigidMotion.**setPrescribedMotion3**(*a*, *motionName*, *transl_speed*, *axis_pnt*, *axis_vct*, *omega*)

Set a precribed motion defined by a constant speed rotation and translation. omega is in rad/time unit.

Parameters

- **a** ([array, list of arrays] or [pyTree, base, zone, list of zones]) – Input data
- **tx** (string) – translation in x motion string
- **ty** (string) – translation in y motion string
- **tz** (string) – translation in z motion string
- **cx** (string) – rotation center x coordinate motion string
- **cy** (string) – rotation center y coordinate motion string
- **cz** (string) – rotation center z coordinate motion string
- **ex** (string) – rotation axis x coordinate motion string
- **ey** (string) – rotation axis y coordinate motion string
- **ez** (string) – rotation axis z coordinate motion string
- **angle** (string) – rotation angle motion string

Example of use:

- Set a prescribed motion of type 3 (pyTree):

```
# - setPrescribedMotion3 (pyTree) -
# Motion defined by a constant speed and rotation speed
import RigidMotion.PyTree as R
import Converter.PyTree as C
import Geom.PyTree as D

a = D.sphere((1.2,0.,0.), 0.2, 30)
```

(continues on next page)

(continued from previous page)

```
a = R.setPrescribedMotion3(a, 'mot', transl_speed=(1,0,0))

C.convertPyTree2File(a, 'out.cgns')
```

3.1 General functions

RigidMotion.**evalPosition**(*a*, *time*)

Evaluate the position at time t according to a motion. If the motion is defined in a with setPrescribedMotion.

Parameters

- **a** ([pyTree, base, zone, list of zones]) – input data
- **time** (float) – evaluation time

Returns reference copy of a

Return type identical to input

Example of use:

- Evaluate position (pyTree):

```
# - evalPosition (pyTree) -
import RigidMotion.PyTree as R
import Converter.PyTree as C
import Geom.PyTree as D

a = D.sphere((1.2,0.,0.), 0.2, 30)
a = R.setPrescribedMotion1(a, 'trans', tx="{t}")
b = R.evalPosition(a, time=0.1)

C.convertPyTree2File(b, 'out.cgns')
```

Evaluate position at given time, when motion is described by a function. F(t) is a function describing motion. F(t) = (centerAbs(t), centerRel(t), rot(t)), where centerAbs(t) are the coordinates of the rotation center in the absolute frame, centerRel(t) are the coordinates of the rotation center in the relative (that is array's) frame and rot(t), the rotation matrix.

Parameters

- **a** ([pyTree, base, zone, list of zones]) – input data
- **time** (float) – evaluation time

- **F** (python function) – motion function

Returns reference copy of a

Return type identical to input

Example of use:

- Evaluate position with function (pyTree):

```
# - evalPosition (PyTree) -
import RigidMotion.PyTree as R
import Generator.PyTree as G
import Converter.PyTree as C
from math import *

# Coordonnees du centre de rotation dans le repere absolu
def centerAbs(t): return [t, 0, 0]

# Coordonnees du centre de la rotation dans le repere entraine
def centerRel(t): return [5, 5, 0]

# Matrice de rotation
def rot(t):
    omega = 0.1
    m = [[cos(omega*t), -sin(omega*t), 0],
         [sin(omega*t), cos(omega*t),  0],
         [0,            0,             1]]
    return m

# Mouvement complet
def F(t): return (centerAbs(t), centerRel(t), rot(t))

a = G.cart((0,0,0), (1,1,1), (11,11,2))

# Move the mesh
time = 3.
b = R.evalPosition(a, time, F); b[0]='move'
C.convertPyTree2File([a,b], "out.cgns")
```

CHAPTER

FOUR

INDEX

- genindex
- modindex
- search