Initiator 2.5

Stephanie Peron, Christophe Benoit, Gaelle Jeanfaivre, Pascal Raud
- Onera -

1 Initiator: solution initialization module

1.1 Preamble

Initiator module works on arrays or on pyTrees containing grid information (coordinates must be defined).
For use with the array interface, you have to import Initiator module:

```
import Initiator as I
```

For use with the pyTree interface:

```
import Initiator.PyTree as I
```

1.2 CFD field initialisations on an array or on a pyTree

The input array (or pyTree) a is an array (or a pyTree) defining the grid on which the solution has to be initialized. The output variables are the variables defined in a. If the five conservative variables are not present, then default output variables are coordinates and conservative variables.

I.initConst: initialization by a constant field, given Mach number, incident flow angles and Reynolds number. If no argument are provided, try to initialiaze from reference state:

```
b = I.initConst(a, MInf=None, alphaZ=0., alphaY=0., ReInf=1.e8, loc='nodes')
```

(See: initConst.py) (See: initConstPT.py)

I.initLamb: initialization by a 2D Lamb vortex at position (x0,y0), intensity Gamma and infinite Mach number MInf:

```
b = I.initLamb(a, (x0,y0), Gamma=2., MInf=0.5, loc='nodes')
```

(See: lamb.py) (See: lambPT.py)

I.initVisbal: initialization by a 2D Visbal vortex at position (x0,y0), intensity Gamma and infinite Mach number MInf:

```
b = I.initVisbal(a, (x0,y0), Gamma=2., MInf=0.5, loc='nodes')
```

(See: visbal.py) (See: visbalPT.py)

I.initScully: initialization by a 2D Scully vortex at position (x0,y0), intensity Gamma, core radius coreRadius and infinite Mach number MInf:

```
b = I.initScully(a, (x0,y0), Gamma=2., coreRadius=1., MInf=0.5, model=0, loc='nodes')
```

In this function, the parameter 'model' can be 0 or 1: 0 means constant entropy outward of vortex core and constant temperature in the vortex core, 1 means constant entropy every where.
(See: scully.py) (See: scullyPT.py)

I.initYee: initialization by a 2D Yee isentropic vortex at position (x0,y0), intensity Gamma and infinite Mach number MInf:

```
b = I.initYee(a, (x0,y0), Gamma=2., MInf=0.5, loc='nodes')
```

(See: yee.py) (See: yeePT.py)

I.overlayField: overlay the field of two arrays or pyTrees (b and c). Only density, velocity and energy stagnation density are overlaid. Both fields must use the same adimensioning, which corresponds to the same infinite Mach number MInf. Both arrays must be defined on the same mesh.

```
a = I.overlayField(b, c, MInf=0.5, loc='nodes')
```

(See: overlay.py) (See: overlayPT.py)

1.3 Example files

Example file: initConst.py

```
# - initConst (array) -
import Converter as C
import Generator as G
import Initiator as I

NI = 100; NJ = 100
HI = 50./(NI-1); HJ = 50./(NJ-1)
a = G.cart((0.,0.,0.), (HI,HJ,1.), (NI,NJ,2))
a = I.initConst(a, MInf=0.8)
C.convertArrays2File([a], 'out.plt')
```

Example file: initConstPT.py

```
# - initConst (pyTree) -
import Converter.PyTree as C
import Generator.PyTree as G
import Initiator.PyTree as I

NI = 100; NJ = 100
HI = 50./(NI-1); HJ = 50./(NJ-1)
a = G.cart((0.,0.,0.), (HI,HJ,1.), (NI,NJ,2))
a = I.initConst(a, MInf=0.8, loc='centers')
C.convertPyTree2File(a, 'out.cgns')
```

Example file: lamb.py

```
# - initLamb (array) -
import Converter as C
import Generator as G
import Initiator as I
```

```
NI = 100; NJ = 100
HI = 50./(NI-1); HJ = 50./(NJ-1)
a = G.cart( (0.,0.,0.), (HI,HJ,1.), (NI,NJ,2))
a = I.initLamb(a, position=(25.,25.), Gamma=2., MInf=0.8)
C.convertArrays2File([a], 'out.plt')
```

Example file: lambPT.py

```
# - initLamb (pyTree) -
import Converter.PyTree as C
import Generator.PyTree as G
import Initiator.PyTree as I

NI = 100; NJ = 100
HI = 50./(NI-1); HJ = 50./(NJ-1)
a = G.cart((0.,0.,0.), (HI,HJ,1.), (NI,NJ,2))
a = I.initLamb(a, position=(7.,7.), Gamma=2., MInf=0.8, loc='centers')
C.convertPyTree2File(a, "out.cgns")
```

Example file: visbal.py

```
# - initVisbal (array) -
import Converter as C
import Generator as G
import Initiator as I

NI = 100; NJ = 100
HI = 50./(NI-1); HJ = 50./(NJ-1)
a = G.cart( (0.,0.,0.), (HI,HJ,1.), (NI,NJ,2))
a = I.initVisbal(a,position=(25.,25.), Gamma=2., MInf=0.8)
C.convertArrays2File([a], 'out.plt')
```

Example file: visbalPT.py

```
# - initVisbal (pyTree) -
import Converter.PyTree as C
import Generator.PyTree as G
import Initiator.PyTree as I

NI = 100; NJ = 100
HI = 50./(NI-1); HJ = 50./(NJ-1)
MachInf = 0.8

a = G.cart((0.,0.,0.), (HI,HJ,1.), (NI,NJ,2))
z = I.initVisbal(a, (3.,3.), 2., MachInf, loc='centers')
C.convertPyTree2File(z, 'out.cgns')
```

Example file: scully.py

```
# - initScully (array) -
import Generator as G
import Converter as C
import Initiator as I

NI = 200; NJ = 200
HI = 1./(NI-1); HJ = 1./(NJ-1)
a = G.cart((0.,0.,0.), (HI,HJ,1.), (NI,NJ,2))
a = I.initScully(a, (0.5,0.5), -0.2, 0.05, 0.8, 0)
C.convertArrays2File([a], "out.plt")
```

Example file: scullyPT.py

```
# - initScully (pyTree) -
import Generator.PyTree as G
import Converter.PyTree as C
import Initiator.PyTree as I

NI = 200; NJ = 200
HI = 1./(NI-1); HJ = 1./(NJ-1)
a = G.cart((0.,0.,0.), (HI,HJ,1.), (NI,NJ,2))
a = I.initScully(a, (0.5,0.5), -0.2, 0.05, 0.8, 0, loc='centers')
C.convertPyTree2File(a, 'out.cgns')
```

Example file: yee.py

```
# - initYee (array) -
import Converter as C
import Generator as G
import Initiator as I

NI = 100; NJ = 100
HI = 50./(NI-1); HJ = 50./(NJ-1)
a = G.cart( (0.,0.,0.), (HI,HJ,1.), (NI,NJ,2))
a = I.initYee(a, position=(25.,25.), Gamma=2., MInf=0.8)
C.convertArrays2File([a], 'out.plt')
```

Example file: yeePT.py

```
# - initYee (pyTree) -
import Converter.PyTree as C
import Generator.PyTree as G
import Initiator.PyTree as I

NI = 100; NJ = 100
HI = 50./(NI-1); HJ = 50./(NJ-1)
MachInf = 0.8

a = G.cart((0.,0.,0.), (HI,HJ,1.), (NI,NJ,2))
z = I.initYee(a, (3.,3.), 2., MachInf, loc='centers')
C.convertPyTree2File(z, 'out.cgns')
```

Example file: overlay.py

```
# - overlayField (array) -
import Converter as C
import Generator as G
import Initiator as I

NI = 100; NJ = 100
HI = 50./(NI-1); HJ = 50./(NJ-1)
MachInf = 0.8

a = G.cart((0.,0.,0.), (HI,HJ,1.), (NI,NJ,2))
ac = I.initVisbal(a, (3.,3.), 2., MachInf)
ac2 = I.initVisbal(a, (20.,3.), 2., MachInf)
an = I.overlayField(ac, ac2, MachInf)
C.convertArrays2File([an], "out.plt")
```

Example file: overlayPT.py

```
# - overlayField (pyTree) -
import Converter.PyTree as C
import Generator.PyTree as G
import Initiator.PyTree as I
```

```
NI = 100; NJ = 100
HI = 50./(NI-1); HJ = 50./(NJ-1)
MachInf = 0.8
a = G.cart((0.,0.,0.), (HI,HJ,1.), (NI,NJ,2))
z1 = I.initVisbal(a, (3.,3.), 2., MachInf, loc='centers'); z1[0]='cart1'
z2 = I.initVisbal(a, (20.,3.), 2., MachInf, loc='centers'); z2[0]='cart2'
zn = I.overlayField(z1, z2, MachInf, loc='centers')
C.convertPyTree2File(zn, 'out.cgns')
```