Thermolib Documentation

Release 3.5

/THERMOLIB/MU-01/V3.5

Nov 21, 2022

CONTENTS

CHAPTER

ONE

PREAMBLE

This is a python module interfacing CEDRE Thermolib, that enables to compute complex variables from a solution.

Fast is only available for use with the pyTree interface. You must import the module:

```
import Thermolib.PyTree as Thermolib
```

CHAPTER

TWO

LIST OF FUNCTIONS

– Actions

Thermolib.PyTree.Thermo(filename[, workdir, ...])	
Thermolib.PyTree.Thermo._computeFlowVars(t, ...)	Compute flow variables.
Thermolib.PyTree.Thermo._computeFluidVars(t, ...)	Compute fluid variables.
Thermolib.PyTree.Thermo._computeFluidVsp(t, ...)	Compute variable for each species.

CHAPTER

THREE

CONTENTS

3.1 Actions

`Thermolib.PyTree.`**`Thermo`**(*file*, *workdir='.'*))

Create a handle for computing variables with thermolib.

Parameters

- **`file`** (`string`) – xml file used by epicea
- **`workdir`** (`string`) – directory where xml files are

`Thermolib.PyTree.Thermo.`**`_computeFlowVars`**(*a*, *variables*, *cont=None*)

Compute flow variables in place. If cont is specified, compute on specific container. Possible variables are:

- Mach number: "M", "Mach"
- Total enthalpy: "Htot", "Total Enthalpy", "Enthalpie totale"
- Total energy: "Etot", "Total Energy", "Energie totale"
- Total pressure: "Ptot", "Total Pressure", "Pression totale"
- Total temperature: "Ttot", "Total Temperature", "Temperature totale"
- Density: "Rho", "Density", "Masse volumique"
- Enthalpy: "H", "Enthalpy", "Enthalpie statique"
- Internal energy: "E", "Internal Energy", "Energie interne"
- Specific heat Cp: "Cp", "Specific Heat CP", "Chaleur specifique CP"
- Specific heat cv: "Cv", "Specific Heat CV", "Chaleur specifique CV"
- Gamma: "Gamma", "Specific Heat Ratio"
- Sound speed: "Vson", "Sound Speed", "Vitesse du son"

- Entropy: "S", "Entropy", "Entropie"
- Molmass: "Mm", "Molmass", "Masse molaire"
- Viscosity: "Mu", "Viscosity", "Viscosite"
- Conductivity: "Lambda", "Conductivity", "Conductivite"
- Electric conductivity: "Sigma elec", "Electric conductivity", "Conductivite elec"

Parameters

- **a** (`Zone, list of Zones, Base, pyTree`) – input data
- **variables** (`list of strings`) – list of variables to be computed
- **cont** (`string`) – container name (optional)

Example of use:

- Compute flow variables using thermolib (pyTree):

```
# - computeFlowVars (PyTree) -
import Converter.PyTree as C
import Converter.Internal as Internal
import Thermolib.PyTree as Thermolib

Internal.__FlowSolutionCenters__ = 'SolutionFlow'

t = C.convertFile2PyTree("Simple/archive_CHARME.hdf")

h = Thermolib.Thermo('epicea.xml', workdir='Simple')

#h._computeFlowVars(t, ['centers:Mach', 'centers:Etot'])
h._computeFlowVars(t, ['Mach', 'Etot'], 'SolutionFlow')

C.convertPyTree2File(t, 'out.hdf')
```

`Thermolib.PyTree.Thermo.`**`_computeFluidVars`**(*a*, *variables*, *cont=None*)

Compute fluid variables in place. If cont is specified, compute on specific container. Possible variables are:

- Density: "Rho", "Density", "Masse volumique"
- Enthalpy: "H", "Enthalpy", "Enthalpie statique"
- Internal energy: "E", "Internal Energy", "Energie interne"
- Specific heat Cp: "Cp", "Specific Heat CP", "Chaleur specifique CP"
- Specific heat cv: "Cv", "Specific Heat CV", "Chaleur specifique CV"

- Gamma: "Gamma", "Specific Heat Ratio"
- Sound speed: "Vson", "Sound Speed", "Vitesse du son"
- Entropy: "S", "Entropy", "Entropie"
- Molmass: "Mm", "Molmass", "Masse molaire"
- Viscosity: "Mu", "Viscosity", "Viscosite"
- Conductivity: "Lambda", "Conductivity", "Conductivite"
- Electric conductivity: "Sigma elec", "Electric conductivity", "Conductivite elec"

Parameters

- **a** (Zone, list of Zones, Base, pyTree) – input data
- **variables** (list of strings) – list of variables to be computed
- **cont** (string) – container name (optional)

Example of use:

- Compute fluid variables using thermolib (pyTree):

```
# - computeFluidVars (PyTree) -
import Converter.PyTree as C
import Converter.Internal as Internal
import Thermolib.PyTree as Thermolib

Internal.__FlowSolutionCenters__ = 'SolutionFlow'

t = C.convertFile2PyTree("Simple/archive_CHARME.hdf")

h = Thermolib.Thermo('epicea.xml', workdir='Simple')

#h._computeFluidVars(t, ['centers:Rho', 'centers:H'])
h._computeFluidVars(t, ['Rho', 'H'], 'SolutionFlow')

C.convertPyTree2File(t, 'out.hdf')
```

Thermolib.PyTree.Thermo.**_computeFluidVsp**(*a*, *variables*, *cont=None*)

Compute fluid variables for species in place. If cont is specified, compute on specific container. Possible variables are:

- Enthalpy: "H", "Enthalpy", "Enthalpie (esp)"
- Internal energy: "E", "Internal Energy", "Energie interne (esp)"
- Specific heat Cp: "Cp", "Specific Heat CP", "Chaleur specifique CP (esp)"

- Specific heat Cv: "Cv", "Specific Heat CV", "Chaleur specifique CV (esp)"
- Entropy: "S", "Entropy", "Entropie (esp)"
- Viscosity: "Mu", "Viscosity", "Viscosite (esp)"
- Conductivity: "Lambda", "Conductivity", "Conductivite (esp)"
- Diffusivity: "Cdif", "Diffusivity", "Coeff diffusion (esp)"
- Mole fractions: "Xj", "Fractions molaires (esp)"
- Chemical potentials: "Gj", "Potentiels chimiques (esp)"

  Parameters

  - **a** (Zone, list of Zones, Base, pyTree) – input data
  - **variables** (list of strings) – list of variables to be computed
  - **cont** (string) – container name (optional)

Example of use:

- Compute fluid variables for species using thermolib (pyTree):

```
# - computeFluidVsp (PyTree) -
import Converter.PyTree as C
import Converter.Internal as Internal
import Thermolib.PyTree as Thermolib

Internal.__FlowSolutionCenters__ = 'SolutionFlow'

t = C.convertFile2PyTree("Simple/archive_CHARME.hdf")

h = Thermolib.Thermo('epicea.xml', workdir='Simple')

#h._computeFluidVsp(t, ['centers:H', 'centers:Cp'])
h._computeFluidVsp(t, ['H', 'Cp'], 'SolutionFlow')

C.convertPyTree2File(t, 'out.hdf')
```

CHAPTER FOUR

INDEX

- genindex
- modindex
- search