ECOGEN 1.0

Evolutive, Compressible, Open, Genuine, Easy, N-phase

TABLE OF CONTENTS

I. GENERAL INFORMATIONS – LICENSE

1. PROPERTY, WARRANTY

ECOGEN is the legal property of its developers, whose names are listed in the copyright file included with the source distribution. Contributors' names for version 1.0 are listed below:

- Eric Daniel,
- Sébastien Le Martelot,
- Kevin Schmidmayer,
- Fabien Petitpas

ECOGEN is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

ECOGEN is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details available at the following address: http://www.gnu.org/licenses.

2. ECOGEN_V1.0 PACKAGE

This user's guide is for the version 1.0 of ECOGEN package: ECOGEN_V1.0.zip. The package is available on the permanent link: https://github.com/code-mphi/ECOGEN.

The package includes several files or folders organized as described below:

- *ECOGEN/src/* folder including 254 C++ source files in 23 subfolders.
- *ECOGEN/libMeshes/* folder including examples of unstructured meshes in *.geo format (gmsh files[1]).
- *ECOGEN/libEOS/* folder including some possible parameters for Ideal Gas or Stiffened Gas Equation of State in XML files.
- *ECOGEN/libTests* folder including:
    - *ECOGEN/libTests/referenceTestCases/* folder organized in a test cases library according the flow model (Euler Equations ECOGEN solver, Kapila's model for multiphase flow ECOGEN solver, Homogeneous Euler Equation ECOGEN solver, etc.)
    - 4 quick-manual XML files to create a new flow calculation with ECOGEN.
    - 1 reporting file for test cases.
- *ECOGEN.xml* main entry file to select running cases.
- *Makefile*: for compilation in Unix environment. This file may require some adaptation to the user's environment.
- LI*CENSE, COPYRIGHT* and *AUTHORS*: Information files about authors and licensing.
- *README_Developer*: Information files for developers.
- ECOGEN_V1.0_userGuide.pdf: The present user's guide for ECOGEN. Include full description of input and output files to proceed to a run using ECOGEN.

ECOGEN should be compiled using a C++ compiler. It also required a functional system implementation of MPI library (not provided in this package).

[1] Gmsh is a simple and efficient free mesh generator written by Christophe Geuzaine and Jean-François Remacle (see: http://gmsh.info/).

II. USER'S GUIDE

ECOGEN settings are managed via INPUT FILES only. The global INPUT FILES and OUTPUT FILES structure is depicted in Figure II-1.

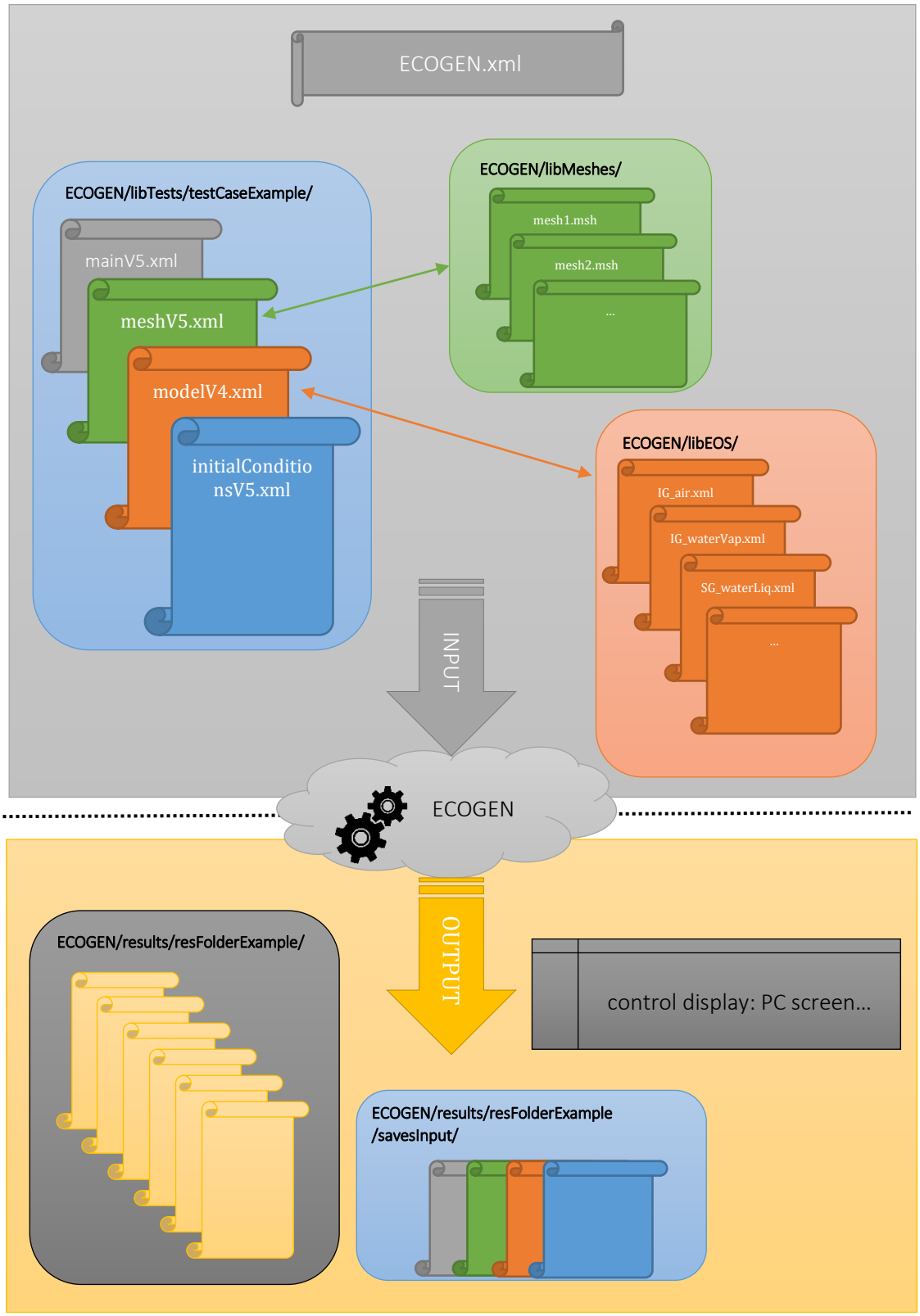

Figure II-1 : Structure of input and output files in ECOGEN.

This user's guide describes the INPUT FILES organization and OUTPOUT FILES created by ECOGEN

2. INPUT FILES

The standard XML format is used for ECOGEN input files. This data format is using markups that gives to the input files some interesting flexibility[1]. ECOGEN is using the TinyXML-2[2] parser to read/write xml files.

ECOGEN package includes a sample of test cases. Each of the them is independent, ready to use and can also be adapted and enriched to develop a new configuration. The test case to be run is chosen in the main input file: *ECOGEN.xml*. In Figure II-2 the minima structure of this file is reported.

```
1  <?xml version = "1.0" encoding = "UTF-8" standalone = "yes"?>
2  <ecogen>
3      <testCase>./libTests/casTestExemple/</testCase>
4  </ecogen>
```

Figure II-2 : Minima structure of the main input file: ECOGEN.xml

In this file, the <testCase> markup indicates the folder containing the test case to be run. It is then possible to run successively several cases by adding as many <testCase> markup as necessary.
Each folder indicated in a <testCase> markup must contain 4 input files:

- *mainV5.xml*
- *meshV5.xml*
- *modelV4.xml*
- *initialConditionsV4.xml*

Additional input files depending on the test case are necessary. They are placed in the following folders:

- *ECOGEN/libEOS/*: contains the files defining the material used in the test case.
- *ECOGEN/libMeshes/*: contains the files defining the mesh used in the test case.

In this section the structure of each input file is detailed. This information is also provided in a condensed form in the "handbook" files at the root folder of test cases library *ECOGEN/libTests/*. These files are named and constitute quick-reference manuals:

- *manualMainV5.xml*
- *manualMeshV5.xml*
- *manualModelV4.xml*
- *manualInitialConditionsV4.xml*

2.1. MAINV5.XML INPUT FILE

mainV5.xml is the main input file needed to define the test case. It **must** be in the current test case folder. Its minimal structure is shown in Figure II-3.

[1] Learning XML at https://www.w3schools.com/xml/.
[2] TinyXML-2 is a simple and efficient open source parser written by Lee Thomason (see: http://www.grinninglizard.com/).

```xml
<?xml version = "1.0" encoding = "UTF-8" standalone = "yes"?>
<computationParam>
    <run>folderNameResults</run>
    <OutputMode format="GNU" binary="false"/>
    <timeControlMode iterations="false">
        <!-- <iterations number="100" iterFreq="10"/> -->
        <physicalTime totalTime="8.e-3" timeFreq="8.e-4"/>
    </timeControlMode>
    <computationControl CFL="0.8"/>
</computationParam>
```

Figure II-3 : Minimal structure of mainV5.xml input file.

It contains general parameters for the computation listed below. Some are mandatory, others optional.

NAME OF THE RUN

The <run> markup is **mandatory**. It will be used to create the folder containing the test case results in *ECOGEN/results/*.

OUTPUT FORMAT

The <outputMode> markup is **mandatory**. The user can choose the writing output format. The attributes are:

- format: can take the value GNU (standard writing in column) or XML (XML VTK format[1]).
- binary: can take the value true or false. Binary (true) or ASCII (false) format can be chosen.

Output results files will be placed in the folder *ECOGEN/results/* into a specific subfolder with the name of the run.

TIME EVOLUTION CONTROL

ECOGEN is a CFD tool based on an explicit integration scheme in time. The <timeControlMode> markup is **mandatory** and defines the temporal evolution of the current simulation. it contains the iterations attribute that can take the two values:

- true: the time control is done thanks to the total number of timesteps and the <iterations> node must be present.
- false: the time control is done thanks to the physical final time and the <physicalTime> node must be present.

The <iterations> markup:

ECOGEN automatically computes the timestep value thanks to a numerical stability criterion (CFL[2] criterion). This markup is defined with the attributes:

- number: Integer equals to the total number of temporal timesteps.
- iterFreq: Integer equal to the frequency of results writing (results are written every iterFeq timestep)

The < physicalTime > markup:

If this markup is used ECOGEN automatically determines the total amount of timestep to compute to reach the chosen physical time. The attributes are:

- totalTime: Real number equals to the physical final time of the simulation. (unit: s (SI)).

[1] VTK format is used in PARAVIEW visualization software (see: https://www.vtk.org/wp-content/uploads/2015/04/file-formats.pdf)

[2] CFL criterion is based on the minimal length size of the mesh and the speediest acoustic wave.

- timeFreq: Real number equals to the frequency of results writing (results are written every timeFreq seconds).

CFL CRITERION

The <computationControl> markup is **mandatory**. It specifies the value of the attribute CFL which ensures the stability of the temporal integration scheme: the value (real number) must be less than 1.

GLOBAL ACCURACY ORDER OF THE NUMERICAL SCHEME

When it is possible (according to the mesh or to the flow model) ECOGEN can use a second-order scheme (based on MUSCL approach with a TVD slope limiter). In this case the optional markup <secondOrder> can be inserted in the *mainV5.xml* input file as in the following example:

```
<secondOrder>
    <globalLimiter>minmod</globalLimiter>
    <interfaceLimiter>superbee</interfaceLimiter>   <!-- optionnal node -->
</secondOrder>
```

The < secondOrder > markup **must** contain the node <globalLimiter>. The node <interfaceLimiter> is optional and specify a different slope limiter at the flow interface[1].
The slope limiters available in ECOGEN are the following: minmod[2], vanleer[3], vanalbada[4], mc[5], superbee[6].

PROBES

It is possible to record flow variables at given locations in the computational domain against time. This is done by including to the *mainV5.xml* input file the optional <probe> markup.

```
<probe name="probe1">
    <vertex x="0.3" y="0.05" z="0.05"/>
    <timeControl acqFreq="1e-5."/>       <!-- if negative or nul, recording at each time step -->
</probe>
```

The two following nodes **must** be included in the <probe> markup:

- <vertex>: Specifies the location of the probe into the computational domain in each physical direction corresponding to the attributes: x, y and z (unit: m (SI)). Values must be real numbers.
- <timeControl>: permits to specify the probe acquisition frequency (unit: s (SI)). If the value is set to zero or negative, flow values at the probe location are recorded at each time step.

Probes output results files will be placed in the specific subfolder *ECOGEN/results/XXX/probes/* where XXX represent the name of the run.

---

[1] A flow interface is a specific location in the flow: boundary between two phases in multiphase flows – density discontinuity is one-phase flow.
[2] Roe, P. L. (1986). Characteristic-based schemes for the Euler equations. *Annual review of fluid mechanics*, *18*(1), 337-365.
[3] Van Leer, B. (1974). Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme. *Journal of computational physics*, *14*(4), 361-370.
[4] Van Albada, G. D., Van Leer, B., & Roberts Jr, W. W. (1982). A comparative study of computational methods in cosmic gas dynamics.
[5] Van Leer, B. (1977). Towards the ultimate conservative difference scheme III. Upstream-centered finite-difference schemes for ideal compressible flow. *Journal of Computational Physics*, *23*(3), 263-275.
[6] Roe, P. L. (1986). Characteristic-based schemes for the Euler equations. *Annual review of fluid mechanics*, *18*(1), 337-365.

Remarks:

1) Recording probe with a high frequency could have a significant impact on computation performances due to the computer memory time access. To prevent that, one should fix a reasonable acquisition frequency.
2) Several probes can be added simultaneously. For that, place as many as wanted <probe> markups in the *mainV5.xml* input files.

2.2. MESHV5.XML INPUT FILE

meshV5.xml is the input necessary to specify the geometrical characteristics of the computational domain and the kind of mesh used. This file is **mandatory** and must be present in the folder of the current case. The minimalist content of this file in depicted in Figure II-4.

```
<?xml version="1.0" encoding="UTF-8" standalone = "yes"?>
<mesh>
    <type structure="cartesian"/>
    <cartesianMesh>
        <dimensions x="1.e-1" y="5.e-2" z="1."/>
        <numberCells x="50" y ="25" z="1"/>
    </cartesianMesh>
</mesh>
```

Figure II-4 : meshV5.xml input file for a cartesian mesh.

The <type> markup is mandatory and specifies via the attribute structure the kind of mesh used in the current computation:

- cartesian: ECOGEN automatically generates its own Cartesian mesh.
- unstructured: a specific external mesh generator (not provide in ECOGEN) must be used to generate the mesh.

CARTESIAN MESH

Consider the specific data presented in Figure II-4. In this case, ECOGEN is able to automatically generate a cartesian mesh according the markup <cartesianMesh> which **must** contain the two following nodes:

- <dimensions>: Specifies the physical dimensions of the computational domain in each physical direction corresponding to the attributes: x, y and z (unit: m (SI)). Values must be real numbers.
- <numberCells>: Specify the number of cells in each direction corresponding to the x, y and z attributes. The values are integer numbers.

UNSTRUCTURED MESH

The <unstructuredMesh> markup must be present in the *meshV5.xml* input file and contains the following nodes:

- <file>: this **mandatory** node specifies the path of the mesh file via the attribute name. The file must be located in the folder *ECOGEN/libMeshes/*.
- <modeParallele>: This node is required only if the file mesh is a multi-CPU file. The attribute GMSHPretraitement can take the following values:
    - true: ECOGEN automatically splits the given mesh file in as many as necessary files according to the number of available CPU.

- false: do not redo the split of the given mesh (which has already been split in a precedent simulation).

Remark: The attribute GMSHPretraitement must be set as true if it is the first run with the given mesh file.

IMPORTANT INFORMATION ABOUT THE MESH FILE FORMATS

ECOGEN is built to use mesh files for mono- or multiprocessors computations generated with the opensource Gmsh[1] software with some specific precaution when editing the geometry file (**.geo*). Only the ***MSH file format version 2*** can be used in the ECOGEN_V1.0 version. In Figure II-5 is depicted a mesh file that corresponds to a nozzle. This geometry file corresponds to that available in *ECOGEN/libMeshes/nozzles/nozzle2D_simple2.geo*.

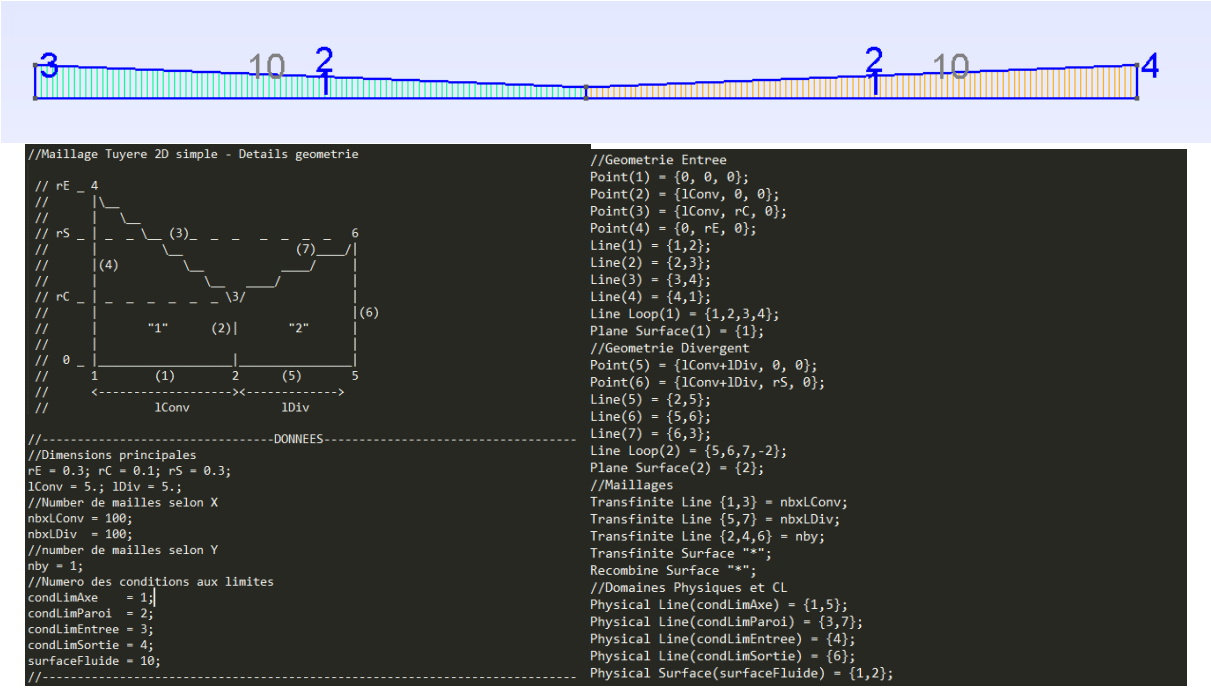

Figure II-5 : Example of geometrical data file - tuyere2D_simple2.geo – for generating a mesh file at .msh format using Gmsh software and usable with ECOGEN.

The computational domain is a nozzle, the mesh is unstructured with quadrangles. To define a computation, the rules are the following:

- Each part of the domain occupied by the fluid should correspond to a *physical surface* or a *physical volume*[2] which is attribute to the value 10.
- Each boundary condition must correspond to a *physical line* or a *physical surface*[3]. The values are successively taken from 1 to maximum 9. This is an important point that will be used to define boundary condition physical treatment in the *initialConditionsV4.xml* input file described in section 2.4.

In the current example of figure Figure II-5, 4 boundaries conditions are set:

- Symetrical axis: 1 (condLimAxe =1)

[1] Gmsh is a simple and efficient free mesh generator written by Christophe Geuzaine and Jean-François Remacle (see: http://gmsh.info/).

[2] The choice between *physical surface* or *physical volume* for fluid parts depends on 2D or 3D computation.

[3] The choice between *physical line* or *physical surface* for boundary conditions depends on 2D or 3D computation.

- Wall: 2 (condLimParoi =2)
- Inflow: 3 (condLimEntree =3)
- Outflow: 4 (condLimSortie =3)

CARTESIAN MESH WITH ADAPTIVE MESH REFINEMENT (AMR) OPTIOON

An efficient Adaptive Mesh refinement (AMR) technology[1] is embedded in ECOGEN. To do that *meshV5.xml* file must content the optional node <AMR> of the <cartesianMesh> markup to define the following attributes:

- lvlMax: integer to define the maximal number of refinements.
- criteriaVar: real number that controls the detection of gradients for the location of the refinement.
- varRho, varP, varU, varAlpha: boolean (true or false) to select the flow quantity on which the gradient operator is applied to detect large gradient.
- xiSplit, xiJoin: normalized real numbers to control if a computational cell, selected by its high gradient value, must be refined or un-refined (values are in the range 0-1.).

The global efficiency of the method is greatly depending on the chosen values for the criteriaVar, xiSplit and xiJoin attributes. These values depend on the physical problem and required a real know-how. More details about these criterion values can be found in (Schmidmayer, Petitpas, & Daniel, 2018). Here is presented an example for the <AMR> node in *meshV5.xml*:

```
<AMR lvlMax="2" criteriaVar="0.2" varRho="true" varP="true" varU="false" varAlpha="false"
xiSplit="0.1" xiJoin="0.1"/> <!-- Noeud optionnel -->
```

2.3. MODELV4.XML INPUT FILE

The flow model of the computation is specified in *modelv4.xml* file. It is **mandatory** located in the folder of the current case. A typical form of this file is reported in Figure II-6.

```
1  <?xml version = "1.0" encoding = "UTF-8" standalone = "yes"?>
2  <model>
3      <flowModel name="Kapila" numberPhases="2"/>
4      <EOS name="GP_air.xml"/>
5      <EOS name="SG_eau.xml"/>
6  </model>
```

Figure II-6 : modelV4.xml input file.

FLOW MODEL

The <model> markup is **mandatory** to specify the mathematical model to solve during the computation. This markup contains the following attributes:

- name: name of the mathematical flow model. This attribute can take the values: Euler, Kapila, ThermalEq or EulerHomogeneous.
- numberPhases: Integer number corresponding to the number of phases present in the simulations. The total amount of equations is related to this number. This attribute is not necessary for the values of name: Euler, EulerHomogeneous.

[1] Schmidmayer, K., Petitpas, F., & Daniel, E. (2018). Adaptive Mesh Refinement algorithm based on dual trees for cells and faces for multiphase compressible flows.

Remark: if EulerHomogeneous is chosen, two additional attributes may be used: liquid and vapor to specify the number corresponding to the liquid phase and the vapor phase. It is phase 0 (for the first) or 1 (for the second).

EQUATIONS OF STATE

The *modelV4.xml* input fle **must** contain as many <EOS> markups as many number of phases. Each phase is described thanks to relations and parameters. The values of these parameters are specified in a separate file: the attribute name contains the name of this file that must be placed in the folder *ECOGEN/libEOS/*. Some fluid files are already present in the ECOGEN package.

RELAXATION PROCEDURES

An additional attribute <relaxation> may be used to impose some specific equilibrium between the phases depending on the flow model used. The attribute type specifies the kind of equilibrium:

- P: a pressure equilibrium is imposed at every location of the flow. It does not require additional attributes.
- PT: Both pressure and thermal equilibrium are imposed at every location of the flow. It does not require additional attributes.
- PTMu: a thermodynamical equilibrium is imposed at every location of the flow. It must be associated to the node <dataPTMu> with attributes liquid and vapor to specify the name of the EOS of the liquid and the vapor phase.

Hereafter the complete node when PTMu is used:

```
<relaxation type="PTMu">
    <dataPTMu liquid="SG_waterLiq.xml" vapor="IG_waterVap.xml"/>
</relaxation>
```

SOURCE TERMS

The additional <sourceTerms> markup can be used to numerically integrate some source terms in the equations. The attribute type selects the source term:

- heating: related to a thermal energy heating/cooling. This attribute requires the <dataHeating> node with the attribute volumeHeatPower: a real number corresponding to the power by volume unit added to the flow (unit :W/m^3 (SI)).
- gravity: if the gravity is considered. The node <dataGravity> with the following attributes must be present:
    - g: real number for the value of the gravity (unit m/s^2(SI)).
    - axe: can take the value x, y or z according the direction of the gravity in the mesh.
    - direction: can take the value positive or negative according to the considered axis.

Hereafter the complete node when gravity is used:

```
<sourceTerms type="gravity">
    <dataGravity g="9.81" axe="Y" direction="positive"/>
</sourceTerms>
```

2.4. INITIALCONDITIONSV4.XML INPUT FILE

initialConditionsV4.xml input file includes the initial conditions and the boundary conditions of the flow simulation. It is **mandatory** located in the folder of the current case. The typical structure of this file is

depicted in Figure II-7.

```
<?xml version = "1.0" encoding = "UTF-8" standalone = "yes"?>
<CI>
  <!-- LIST OF GEOMETRICAL DOMAINS  -->
  <physicalDomains>
    <domain name="base" state="leftChamber" type="entireDomain"/>
  </physicalDomains>

  <!-- LIST OF BOUNDARY CONDITIONS -->
  <boundaryConditions>
    <boundCond name="CLXm" type="abs" number="1"/>
    <boundCond name="CLXp" type="abs" number="2"/>
  </boundaryConditions>

  <!--  LIST OF STATES  -->
  <state name="leftChamber">
    <material type="fluide" EOS="IG_air.xml">
      <dataFluid alpha="0.5" density="1.0"/>
    </material>
    <material type="fluide" EOS="SG_water.xml">
      <dataFluid alpha="0.5" density="1000.0"/>
    </material>
    <mixture>
      <dataMix pressure = "1.e5"/>
      <velocity x="0." y="0." z="0."/>
    </mixture>
    <transport name="color" value="32."/>
  </state>
</CI>
```

Figure II-7 : initialConditionsV4.xml input file.

PHYSICAL DOMAINS

The <physicalDomains> markup is **mandatory**. Some different initial conditions can be specified at different zones of the computational domain. This markup must contain as many nodes <domain> as necessary to correctly initialize the computational domain (overlaps are possible). A <domain> node contains the following attributes:

- name: a name for the domain. This name has no influence on the choices remaining in this file.
- state: This is the state of the fluid which will be specified with the <state> markup further in the file.
- type: to specify the kind of geometrical domain on which the state of the fluid must be attribute: entireDomain, halfSpace, disc, rectangle, pavement or sphere.

Important remark: The initial conditions are attributed on each domain by using a superposition principle. The order is important: In the case of overlapping, the last attributed data are considered in the flow computation. Hence, it is important to attribute at least the entire domain at the first-place thanks to the value entireDomain.

According to the geometrical shape, additional information is required thanks to the use of the nodes among the list:

- entireDomain: Set the initial condition on the entire domain. No more information required.
- halfSpace: Set the initial condition on a half-domain. The node <dataHalfSpace> must be included with the following attributes:
    - axe: can take the value x, y or z.
    - origin: real number, indicates the location of the edge between the two subdomains on the specified axis.
    - direction: can take the value positive or negative on the specified axis.
- disc: In 2D allows to define a disc on a plane, in 3D a cylinder with an infinite length is defined. The node <dataDisc> must be added with the following attributes:

    - Attributes axe1, axe2: The name of 2 axes to define the plane on which the disc is defined. Can take two different values among x, y or z.
    - Attribute radius: Real number of the radius disc (unit: m (SI)).
    - Node : requires the attributes x, y et z giving the location of the center of the disc in the plan (axe1, axe2) in real numbers (unit: m (SI)).
- Rectangle: In 2D allows to define a rectangle on a plane, in 3D a rectangular beam with an infinite length is defined. The node < dataRectangle > must be added with the following attributes:
    - Attributes axe1, axe2: The name of 2 axes to define the plane on which the disc is defined. Can take two different values among x, y or z.
    - Attributes lAxe1, lAxe2: Length of both sides along (axe1,axe2).
    - Node <posInferiorVertex>: equipped with the attributes x, y and z, real numbers giving the location of the inferior corner in the plane (axe1, axe2).
- Pavement: Set the initial condition in a pavement. The additional node <dataPavement> must be added with the attributes:
    - Attributes lAxeX, lAxeY, lAxeZ: Real numbers for length of each side of the pavement along axes (unit: m (SI)).
    - Node <posInferiorVertex>: with the des attributes x, y and z, real numbers corresponding to the location of the inferior corner (unit: m (SI)).
- sphere: Set the initial condition in a sphere. The additional node <dataSphere> is required with the attributes or nodes:
    - Attribute radius: real number giving the radius of the sphere (unit: m (SI)).
    - Node : with the attributes x, y et z real numbers giving the ocation on the ceter of the sphere (unit: m (SI)).

As example, hereafter the complete node do define a rectangle domain in *(x,y)* plane :

```
<domain name="HP"  state="chambreDroite" type="rectangle">
    <dataRectangle axe1="x" axe2="y" lAxe1="0.3" lAxe2="0.2">
        <posInferiorVertex x="0.2" y="0." z="0."/>
    </dataRectangle>
</domain>
```

BOUNDARY CONDITIONS

The <boundaryConditions> markup is **mandatory**. The boundary conditions are specified at the boundary of the computational domain. This markup must contain as many nodes <boundCond> as necessary to recover the entire boundary. Each <boundCond> node contains the following attributes:

- name: a name for the boundary condition. This name has no influence on the choices remaining in this file.
- type: the kind of boundary condition, to choose among abs, wall, tank, outflow.
- numero: integer number that correspond to the number of the boundary.

According <type>, additional information is required thanks the use of the nodes among the list:

- abs: The numerical treatment corresponds to an outgoing flow with no wave reflection. No more information required.
- wall: The numerical treatment corresponds to a wall boundary condition. No more information required.
- tank: The numerical treatment corresponds to the link between the boundary with an infinite tank. An infinite tank is characterized by a null velocity while pressure and temperature are constant). <tank> requires the <dataTank> node with the following attributes:
    - p0: Stagnation pressure, real number (unit: Pa(SI)).
    - T0: Stagnation pressure, real number (unit: K (SI)).

An additional <fluidsProp> node is necessary to define the presence of each phase in the tank. It must contain as many nodes <dataFluid> as the number of phases in the flow simulation and contains the attributes:

- EOS: the name of the file corresponding to the choice of the EOS for the phase in the tank. This file must correspond to the one specified in *modelV4.xml* input file for every fluid.
- alpha: The volume fraction of the fluid in the tank, real number in the range $]0.,1.[$.

- outflow: In the case of a subsonic flow, the pressure is set equal to the ambient pressure at the boundary. The additional <dataOutflow> node is required with the attributes:
  - p0: outside pressure, real number (unit: Pa(SI)).

Important: The choice of the boundary condition number is made according to the kind of mesh given in *meshV5.xml* input file according to the following rules:

cartesian: The boundaries are ordered and labeled from 1 to 6 (in 3D) according to:

- 1: boundary condition at the minimal *x* location
- 2: boundary condition at the maximal *x* location
- 3: boundary condition at the minimal *y* location
- 4: boundary condition at the maximal *y* location
- 5: boundary condition at the minimal *z* location
- 6: boundary condition at the maximal *z* location

unStructured: When an unstructured mesh is used, the number of the boundary condition must correspond to the number specified in the mesh file **.geo*, presented in section 2.2.

As example, hereafter the complete node <boundaryConditions> corresponding to the boundary conditions for the mesh depicted in Figure II-5 :

```
<boundaryConditions>
    <boundCond name="axe" type="wall" numero="1" />
    <boundCond name="paroi" type="wall" numero="2" />
    <boundCond name="entree" type="tank" numero="3">
        <dataTank p0="10.e5" T0="700."/>
        <fluidsProp>
            <dataFluid EOS="GP_air.xml" alpha="0.5"/>
            <dataFluid EOS="GP_air2.xml" alpha="0.5"/>
        </fluidsProp>
    </boundCond>
    <boundCond name="sortie" type="outflow" numero="4">
        <dataOutflow p0="5.82e5"/>
    </boundCond>
</boundaryConditions>
```

Remark: The boundary conditions are dependent on the flow model specified in *modelV4.xml* input file. Some boundary conditions may be not available for the flow model considered.

MECHANICAL AND THERMODYNAMICAL STATES OF THE FLUID

For each physical domain in the <physicalDomains> markup, a fluid state must correspond. It implies an additional <state> markup for each state of fluid. This <state> markup contains:

- as many <material> nodes as the number of phases involved in the simulation.
- A <mixture> node is required if a multiphase model is used.

Each <material> node corresponds to a phase and contains the following attributes or nodes:

- Attribute type: Only the value fluide is available in the current ECOGEN version.
- Attribute EOS: the name of the file corresponding to the fluid Equation of State parameters. This file must correspond to the one specified in *modelV4.xml* input file for each phase.
- Node <dataFluid>: contains data related to the considered state of the fluid in the current phase.

This last node <dataFluid> as well as the <mixture> node are dependent on the flow model according to:

- Euler: Single phase flow. In this case, the <mixture> node is absent and the <dataFluid> node contains the following attributes or nodes:
    - Attribute temperature: Initial temperature of the fluid, real number (unit: K (SI)).
    - Attribute pressure: Initial pressure of the fluid, real number (unit: Pa(SI)).
    - Node <velocity>: with x, y and z attributes setting the initial values for the components of the velocity vector, real numbers (unit: m/s (SI)).
- Kapila: Multiphase flow at pressure and velocity equilibrium (same velocity and same pressure for every phase). Each <dataFluid> node corresponds to a phase with the following attributes:
    - alpha: Volume fraction of the phase, real number in the range $]0.,1.[$.
    - density: Initial specific mass of the fluid, real number (unit: kg/m^3 (SI)) or temperature: Initial temperature (unit: K).

  Moreover, in this case, the <mixture> node contains:
    - the <dataMix> node with pressure attribute for initial pressure of the fluid, real number (unit: Pa(SI)).
    - the <velocity> node with x, y and z attributes setting the initial values for the components of the velocity vector, real number (unit: m/s (SI)).
- ThermalEq: Multiphase flow at pressure, velocity and thermal equilibrium (same velocity, same pressure and same temperature for every phase). In this case, every <dataFluid> node corresponds to a phase with only one attribute alpha setting the volume fraction real number in the range $]0.,1.[$.

  The <mixture> node contains the following attributes and nodes:
    - the <dataMix> node with temperature and pressure attributes for initial temperature of the fluid, real number (unit: K (SI)) and initial pressure, real number (unit: Pa).
    - Attribute pressure: Initial pressure of the mixture, real number (unit: Pa(SI)).
    - Node <velocity>: with x, y and z attributes setting the initial values for the components of the velocity vector of the mixture, real numbers (unit : m/s (SI)).
- EulerHomogeneous: Multiphase flow at mechanical and thermodynamical equilibrium. In this case, every <dataFluid> node corresponds to a phase with only one attribute alpha setting the volume fraction real number in the range $]0.,1.[$.

  Moreover, in this case, the <mixture> contains the following attributes and nodes:
    - the <dataMix> node with pressure attribute for initial pressure of the mixture, real number (unit: Pa(SI)).
    - Node <velocity>: with x, y and z attributes setting the initial values for the components of the velocity vector of the mixture, real numbers (unit: m/s (SI)).

As example, hereafter the complete node corresponding to the definition of a complete state with the EulerHomogeneous model:

```
<state name="chamberHP">
    <material type="fluide" EOS="SG_waterLiq.xml">
        <dataFluid alpha="0.99"/>
    </material>
    <material type="fluide" EOS="IG_waterVap.xml">
        <dataFluid alpha="0.01"/>
    </material>
    <mixture>
        <dataMix pressure = "1.e6"/>
        <velocity x="0." y="0." z="0."/>
    </mixture>
</state>
```

Remark: Be careful to set the volume fraction in the range $]0,.1.[$ as well as the sum over the phases

equal to 1.

2.5. MATERIALS INPUT FILES

For each phase, an Equation of State is required. The data for a given phase are gathered in a file. Every file must be in the folder *ECOGEN/libEOS/* and must follow rules depending on the EOS. Two equations of states are developed in ECOGEN:

-Ideal gas: for gaseous phase only.

-Stiffened gas: for condensed matter (liquid, solid) in a pressure range where the compressible assumption is reasonable.

IDEAL GAS

It is a simple relation linking the pressure (p), the temperature (T) and the density (ρ) or the specific volume ($v = \frac{1}{\rho}$):

$$pv = rT$$

r is the universal gas constant divided by the molar weight of the gas (unit: J/(K.kg) (SI)). The Gibbs relation as well as Maxwell relations yield the definition of the internal energy:

$$e(p,v) = \frac{p}{\gamma - 1} v + e_{ref}$$

One can easily obtain the entropy:

$$s(p,T) = c_v ln\left(\frac{T^{\gamma}}{p^{\gamma-1}}\right) + s_{ref}$$

The following parameters are assumed constant and $\gamma, c_v, e_{ref}\ et\ s_{ref}$ must be specified in the material file, as in the following example in Figure II-8:

```
<?xml version = "1.0" encoding = "UTF-8" standalone = "yes"?>
<parametersEOS>
    <EOS fichier="IG_air.xml" type="IG"/>
    <parameters
        gamma="1.4"
        cv="800"
        energyRef="0."
        entropyRef="0.">
    </parameters>
</parametersEOS>
```

Figure II-8 :example of material file for air, assumed as an ideal gas.

STIFFENED GAS

This Equation of state was first presented in 1971 by Harlow & Amdsen[1]:

$$e = \frac{p + \gamma p_{\infty}}{(\gamma - 1)} v + e_{ref}$$

This EOS takes into account for the molecular attraction within condensed matter. This quite simple EOS is largely used in numerical simulation based on diffuse interface methods because it is able to

[1] Harlow, F. H., & Amsden, A. A. (1968). Numerical calculation of almost incompressible flow. *Journal of Computational Physics*, *3*(1), 80-93.

reproduce shock relation as well as saturation curve[1] for a liquid-vapor mixture when the phase transition is modeled.

Hereafter some useful relations for the specific volume, the enthalpy and the entropy:

$$v(p,T) = \frac{(\gamma - 1)c_v T}{p + p_\infty}$$

$$h(T) = \gamma c_v T + e_{ref}$$

$$s(p,T) = c_v ln\left(\frac{T^\gamma}{(p + p_\infty)^{\gamma-1}}\right) + s_{ref}$$

The ideal gas law is recovered if $p_\infty = 0$. The following parameters are assumed constant and $\gamma, p_\infty, c_v, e_{ref}\ et\ s_{ref}$ must be specified in the material file, as in the following example:

```
<?xml version = "1.0" encoding = "UTF-8" standalone = "yes"?>
<parametersEOS>
    <EOS fichier="SG_water.xml" type="SG"/>
    <parameters
        gamma="4.4"
        pInf="6.e8"
        cv="1000.0"
        energyRef="0."
        entropyRef="0.">
    </parameters>
</parametersEOS>
```

Figure II-9 : example of material file for liquid water, assumed a stiffened gas.

III. RESULTS FILES

Writing the result files is done according the user's choice in *mainV5.xml* INPUT FILE of the current test (see: section II.2.1). For each run, the results are recorded in the specified folder in *ECOGEN/results/XXX/* where XX is the test case name.

One can select the following format:

- GNU: Format in ASCII, results are given in column.
- XML: VTK file format (ASCII or binary).

The name of the results files follows the rules:

result(*format*)_CPU(*proc*)_AMR(*niveau*)_TIME(*instant*). (*ext*)

that can select results files according:

- (*format*) : data format (empty for ASCII, B64 for binary).
- (*instant*) : time of writing results (depends on the selected frequency for writing.
- (*niveau*) : AMR level (in the case of an AMR simulation).
- (*proc*) : the number of the processor where the results are from (in the case of a

[1] Le Métayer, O., Massoni, J., & Saurel, R. (2004). Élaboration des lois d'état d'un liquide et de sa vapeur pour les modčles d'écoulements diphasiques. *International journal of thermal sciences*, *43*(3), 265-276.

parallel simulation).

- (*ext*) : kind of mesh.

1.1. USING GNU FORMAT

This format lead to results files with the (ext)=*out* extension. This format in colon is very useful for a quick visualization of the results when the freeware *gnuplot*[1] or any other tool. When this format is selected, ECOGEN automatically creates a script file *visualisation.gnu* at the root of the result folder. This allows a very quick and efficient use for 1D runs.

1.2. XML VTK FILE FORMAT

ECOGEN can provide output using XML files in VTK file format (ASCII or BINARY). Writing files according VTK file format leads to files with the extension:

- (*ext*)=vtr : cartesian mesh.
- (*ext*)=vtu : unstructured mesh.

ECOGEN also produces a file named *collection.pvd* in order to load only one pack of files when the software PARAVIEW[2].

1.3. SAVING INPUT FILES

In addition to the results, a copy of the INPUT FILES of the current simulation is done in the subfolder *ECOGEN/results/dossierResExemple/savesEntrees/* to ensure a safe reproduction of the results.

1.4. SCREEN OUTPUT

In real time, some data are available during the simulation directly at the terminal screen as soon as ECOGEN starts. hereafter an example of a screenshot:

```
T2 | ----------------------------------------
T2 | RESULTS FILE NUMBER : 1,  ITERATION 8370
T2 |     Physical time       = 0.100009 s
T2 |     Last time step      = 1.15732e-05 s
T2 |     Elapsed time     = 0 s
T2 | ----------------------------------------
T2 | printing file number : 1...  ...OK
```

One reads:

- number of the results files: here number 1.
- number of the last timestep: here 8370th.
- Real time in the simulation: 0.1s.
- Value of the last timestep: here1.15732 10^{-5} s or 0.15732µs
- CPU time since the beginning of the simulation: here 0s.

[1] Gnuplot is a free simple tool to visualize results (see: http://www.gnuplot.info/)
[2] Paraview is a powerful open source visualization application (see: https://www.paraview.org/).