LPRES Library 1.3.6
Reference Manual

School of Aeronautical and Space Engineering

Technical University of Madrid

In cooperation with Empresarios Agrupados

Pablo Sierra Heras

Supervisors:

Juan Manuel Tizón Pulido (Technical University of Madrid),
Javier Vilá Vacas (Empresarios Agrupados),
José Francisco Moral Moral (Empresarios Agrupados).

Contents

1 Overview of the Library

1.1 Purpose of this Manual

This document is the reference manual for the LPRES Library (version 1.3.6).

This Reference Manual will provide:

- List of the components of the library.
- Global description of the units of the library.
- Icons associated to every component.
- Description of the data and variables associated to every component and function.
- Limitations of every component and function.
- Description of the physical-mathematical model associated to every component or function.

In this document, the matters referring to the programming nomenclature used to do the library are explained in *Appendix A.*

1.2 Introduction

The LPRES Library, which stands for **Liquid Propellant Rocket Engine Simulation**, contains components to predict the behaviour of the different configurations that a liquid propellant rocket engine can have.

Although the qualities of a thorough and in-depth analysis may be desirable in the study of specific single components and scenarios, the long computational times become insurmountable for complex, multi-component systems.

A good simplification allows the reduction of the 3-D governing partial differential equations to 1-D differential equations which no longer require complex solution methods thus allowing much faster computational times.

As it is a basic library, the components included in it, works in steady state. However, transient models are projected to be included in future developments of it. The library works with simple models but its usefulness is to achieve the analysis of complex situations due to the system number of elements.

The LPRES Library is built to emulate the elements of the professional ESPSS toolkit. The LPRES Library employs simple models but it can address the simulation of real systems with many elements. The advantages that the LPRES Library presents are that it can be used with the Educational version of EcosimPro and, moreover, its learning curve is much faster than the learning curve of the ESPSS toolkit.

At least version 5.2.0 of EcosimPro must be installed in order to run the library.

All files in which the library is written use version 3.1 (at least) of the MATH Library, which is one of the libraries included with EcosimPro.

1.3 Components in the LPRES Library

1.3.1 List of Components

Abstract Components

Component Type Name	Items Represented
FluidInFluidOut	Components with a fluid inlet and a fluid outlet
GasTurbo	Gas turbomachinery components

Fluid Operating Components

Component Type Name	Items Represented
Ambient	Liquid discharge to the atmosphere
Turbine_ch	Turbine with a choked inlet
Turbine	Turbine (the inlet may be choked or not)
Compressor	Compressor
Turbine_liq	Liquid turbine
Pump	Pump
Pipe	Liquid pipe with pressure drop
SplitFrac	Flow splitter with pressure drop
Junction	Junction with pressure drop

Component Type Name	Items Represented
Regulator	Pressure regulator
Inlet	Conditions of a fluid inlet in the system
Tank	Liquid tank
TankOpen	Liquid tank pressurised by the atmosphere
Nozzle	Convergent nozzle with the ambient already connected

Other Operating Components

Component Type Name	Items Represented
GasGen	Gas generator
Injector	Injector
CombCha	Combustion chamber
NozzleConDiv	Choked convergent-divergent nozzle
NozzleExt	Nozzle extension
ThrustMonitor	Monitor to measure the thrust
FlowMeter	Mass flow meter
ControlPanel	Calculations with the measurements done by other components

Heat Operating Components

Component Type Name	Items Represented
CoolingJacket	Cooling jacket

Mechanical Operating Components

Component Type Name	Items Represented
Shaft	Mechanical shaft without acceleration
Gearbox	Mechanical shaft with a gearbox and without acceleration

1.3.2 Hierarchy of the Components

The following figure shows the relationships between inherited components.

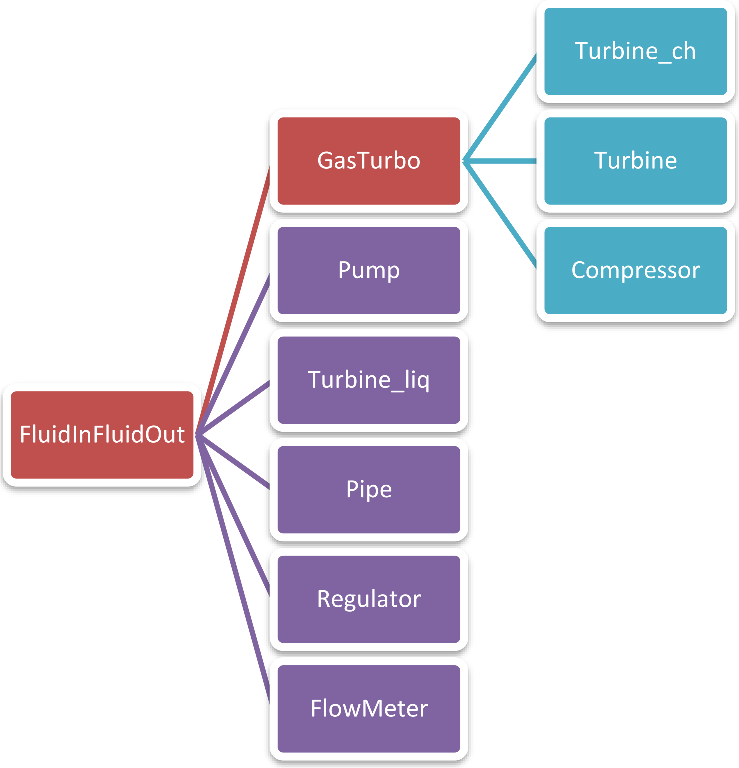


Figure 1 - Hierarchy of the components

2 Global Items

2.1 Global Constants

They are defined in the LPRES_Common.el file.

Name	Type	Value	Description	Units
p_std	CONST REAL	101325	Standard pressure (p_{std})	Pa
T_std	CONST REAL	288.15	Standard temperature (T_{std})	K
g_0	CONST REAL	9.80665	Gravity acceleration at the Earth surface (g_0)	$\frac{m}{s^2}$
R_u	CONST REAL	8314	Universal Gas Constant (R_u)	$\frac{J}{kmol \cdot K}$
T_ref	CONST REAL	298.15	Reference temperature (T_{ref})	K

2.2 Global Variables

They are defined in the LPRES_Common.el file.

Name	Type	Default	Description	Units
Altitude	BOUND REAL	0	Geometric altitude (z)	m

2.3 Global Enumerations

They are defined in the LPRES_Common.el file.

Name	Type	Elements	Description
ChemName	ENUM		Names of available chemicals
Liquids	SET_OF(ChemName)		Names of available liquids
Gases	SET_OF(ChemName)		Names of available gases
LiquidsGases	SET_OF(ChemName)		Names of available chemicals except Comb_prod
LV	SET_OF(ChemName)		Liquids that can be vaporised
ChemState	ENUM	{Liquid, Gas}	State of available chemicals
ConDiv	ENUM	{Convergent, Divergent}	
YesNo	ENUM	{Yes, No}	
Type_Inlet	ENUM	{All, Unknown_W}	
Type_All	ENUM		
OnOffDesign	SET_OF(Type_All)	{Design, Off_design}	
Type_Turbines	SET_OF(Type_All)	{Known_pi, Known_W, Off_design}	
Type_Turbine_liq	SET_OF(Type_All)	{Known_dp, Known_W, Off_design}	
Type_Regulator	SET_OF(Type_All)	{Design, Known_pt_out, Known_dp, Known_dpr}	
Type_cooling	SET_OF(Type_All)	{Darcy, Known_dp}	
AngCoef	ENUM	{Angles, Coefficients}	

The enumerations that have no description are used to set the mode in which the components are acting: if they are on design or off-design, if they are refrigerated, etc.

The “Type_All” enumeration is composed of the “OnOffDesign”, “Type_Turbines”, “Type_Turbine_liq”, “Type_Regulator” and “Type_cooling” elements and allows having the same element (for example “Off_design”) in different enumerations.

The elements of “ChemName”, “Liquids”, “Gases”, “LiquidsGases” and “LV” are described in *Appendix B*.

2.4 Ports

The library uses its own ports, defined in the LPRES_Ports.el file.

Name	Type	Description	Purpose	Symbol
Fluid	SINGLE IN	Fluid port	Port for fluid flow connections	●
FluidInj	SINGLE	Fluid injection port	Port for injection in a combustion chamber	●
GasNozzle	SINGLE	Gas port through a nozzle	Port for gas flow connections through a nozzle	🟣
Mechanical		Mechanical port	Port for mechanical connections	🟤
Heat	SINGLE IN	Heat exchange port	Port for heat exchanges	🔴
Info		Information exchange port	Port for information exchanges between components	○

2.4.1 Port Fluid

Name	Type	Description	Units
Tt	EQUAL REAL	Total temperature (T_t)	K
pt	EQUAL REAL	Total pressure (p_t)	Pa
W	SUM REAL	Mass flow ($\dot{m}$)	$\frac{kg}{s}$
fluid	HIDDEN EQUAL REAL[ChemName]	Working fluid	$-$

The “fluid” variable is an array that is dimensioned with the “ChemName” enumeration. Components of variable “fluid” are the mass fraction of each available chemical (Y_i) plus two auxiliary components. They transmit the molar mass of the products using a stoichiometric mixture ($M_{m,P}$) and the specific heat at constant pressure of the products using a stoichiometric mixture ($c_{P,P}$). They are used if combustion has previously ocurred. If it has not, then the components with indexes “Comb_prod”, “Comb_prod_cp” and “Comb_prod_M” are null.

Note that:

$$Y_{Comb\_prod} + \sum_{i}^{LiquidsGases} Y_i = 1$$

It is not possible to transmit several chemicals through the same port except if there has previously been a combustion.

If there is stoichiometric mixture, $Y_{Comb\_prod} = 1$.

Otherwise, $1 - Y_{CombProd}$ is the excess amount of oxidiser or fuel. As an example, if there is LOX excess, $Y_{LOX} = 1 - Y_{Comb\_prod}$. Hence, despite gases exiting a combustion chamber or a gas generator, the excess amount of LOX is considered a liquid inside the "fluid" array because it is the state of oxygen when entering the chamber. Even though it has no physical sense, it is easier for the library to work with it due to the model used.

Variable "Tt" is the total temperature (T_t) for gases and the temperature (T) for liquids.

Variable "pt" is the total pressure. For liquids, it is calculated as the sum of the static pressure (p) and the dynamic pressure, where ρ the liquid density and v is its speed:

$$p_t = p + \frac{1}{2}\rho v^2$$

2.4.2 Port FluidInj

Name	Type	Description	Units
Tt	EQUAL REAL	Total temperature (T_t)	K
pt	EQUAL REAL	Total pressure (p_t)	Pa
W	SUM REAL	Mass flow $(\dot{m})$	$\frac{kg}{s}$
fluid	HIDDEN EQUAL REAL[ChemName]	Working fluid	$-$
p_c	EQUAL REAL	Chamber pressure (p_c)	Pa

2.4.3 Port GasNozzle

Name	Type	Description	Units
Tt	EQUAL REAL	Total temperature (T_t)	K
pt	EQUAL REAL	Total pressure (p_t)	Pa
W	SUM REAL	Mass flow $(\dot{m})$	$\frac{kg}{s}$

Name	Type	Description	Units
fluid	HIDDEN EQUAL REAL[ChemName]	Working fluid	–
A_out	EQUAL REAL	Nozzle output area (A_{out})	m^2

2.4.4 Port Mechanical

Name	Type	Description	Units
Power	SUM REAL	Mechanical power (W)	W
N	EQUAL REAL	Rotational speed (ω)	$\frac{rad}{s}$

2.4.5 Port Heat

Name	Type	Description	Units
Q	SUM REAL	Heat flux ($\dot{Q}$)	W
T	EQUAL REAL	Temperature (T)	K
A	SUM REAL	Contact area (A)	m^2

2.4.6 Port Info

Name	Type	Description	Units
Data	EQUAL REAL[n]	Information array	–

This port has a Construction Parameter called “n” to set the number of real variables that are transmitted by the port.

2.5 Functions

They are defined in the LPRES_Funcs.el file.

Name	Type	Purpose	Units
Init_fluid	NO_TYPE	To initialise the value of the variable fluid that goes through the ports	–

Name	Type	Purpose	Units
Know_fluid	ENUM ChemName	To know the chemical name of the fluid by means of the variable fluid that goes through the ports	$-$
State	ENUM ChemState	To know the state of a fluid	$-$
Vaporisation	ENUM Gases	To know the gas into which a liquid vaporises	$-$
M	REAL	To calculate molar mass	
R	REAL	To calculate gas constant	$\frac{J}{kg \cdot K}$
cp	REAL	To calculate specific heat at constant pressure	$\frac{J}{kg \cdot K}$
cv	REAL	To calculate specific heat at constant volume	$\frac{J}{kg \cdot K}$
gamma	REAL	To calculate ratio of specific heats	$-$
FGAMMA	REAL	To calculate the value of the Gamma function	$-$
rho	REAL	To calculate the density of a fluid in liquid state	$\frac{kg}{m^3}$
cond	REAL	To calculate thermal conductivity	$\frac{W}{m \cdot K}$
visc	REAL	To calculate dynamic viscosity	$Pa \cdot s$
hdc_fric	REAL	To calculate the friction factor	$-$
GeopotentialAltitude	REAL	To calculate geopotential altitude	m
ISA_Pressure	REAL	To calculate International Standard Atmosphere (ISA) pressure	Pa
ISA_Temperature	REAL	To calculate International Standard Atmosphere (ISA) temperature	K
ISA_Density	REAL	To calculate International Standard Atmosphere (ISA) density	$\frac{kg}{m^3}$

2.5.1 Function Init_fluid

Inputs

Name	Type	Description	Units
fluid_name	IN ENUM ChemName	Working fluid name	$-$
fluid	OUT REAL[ChemName]	Working fluid	$-$

The “fluid_name” input variable contains the working fluid to be transferred by ports. This function initialises the value of the components of the “fluid” variable that goes through the ports using the chemical set by “fluid_name”. Hence:

$$Y_{fluid\_name} = 1$$

$$Y_i = 0 \quad ; \quad i \neq fluid\_name$$

2.5.2 Function Know_fluid

Inputs

Name	Type	Description	Units
fluid	IN REAL[ChemName]	Working fluid	–

Variables

Name	Type	Inital	Description	Units
fluid_name	ENUM ChemName		Working fluid name	–

This function returns the chemical name from the “fluid” variable that goes through the ports.

For combustion gases, it returns “Comb_prod” even if there is an oxidant or fuel excess.

2.5.3 Function State

Inputs

Name	Type	Description	Units
fluid	IN REAL[ChemName]	Working fluid	–

Variables

Name	Type	Inital	Description	Units
fluid_state	ENUM ChemState		Working fluid state	–

Name	Type	Inital	Description	Units
fluid_name	ENUM ChemName		Working fluid name	–

This function returns the state of a fluid from variable "fluid" that goes through the ports. First of all, it calls up the "Know_fluid" function. This function returns the working fluid name, which contains the state of the working fluid.

The chemicals can only be gases, liquids or both. In this last case, they are defined twice in the enumeration "ChemName". Hence it is possible to know their state by their name.

If they can be either gases or liquids, they may have a phase change at some point but they cannot be a multiphase flow.

2.5.4 Function Vaporisation

Inputs

Name	Type	Description	Units
fluid	IN REAL[ChemName]	Working fluid	–

Variables

Name	Type	Inital	Description	Units
Vapour	ENUM[LV] Gases	{O2, CH4, H2, MMH_vapour}	Working fluid state	–
liquid_name	ENUM Liquids		Working liquid name	–

This function returns the gas to which a liquid vaporises. Only liquids included in the enumeration "LV" can be used in this function because these are the chemicals that can be either in gas or liquid state. The "Vapour" enumeration contains the names of the gases into which each liquid of the "LV" enumeration vaporises.

If the working fluid is not a liquid included in the enumeration "LV", a FATAL ASSERT will appear.

2.5.5 Function M

Inputs

Name	Type	Description	Units
fluid	IN REAL[ChemName]	Working fluid	$-$

Variables

Name	Type	Inital	Description	Units
Chem_M	REAL[LiquidsGases]		Molar mass of each chemical ($M_{m,i}$)	$\frac{g}{mol}$

This function calculates the molar mass of the working fluid from the "fluid" variable that goes through the ports.

If the working fluid is a gas or liquid, it takes the value from the "Chem_M" array, whose values are shown in *Appendix B*.

If the working fluid is a combustion gas, it calculates the average molar mass of the mixture of gases by using this expression:

$$\frac{1}{M_m} = \frac{Y_{Comb\_prod}}{M_{m,P}} + \sum_{i}^{LiquidsGases} \frac{Y_i}{M_{m,i}}$$

Note that it is not possible to transmit several chemicals through the same port except if there has been combustion beforehead. Hence, the equation above is applicable in any case.

It is also important to keep in mind that the same chemical has the same value of $M_{m,i}$ regardless of its state. Therefore, for example, $M_{m,LOX} = M_{m,O2}$.

2.5.6 Function R

Inputs

Name	Type	Description	Units
fluid	IN REAL[ChemName]	Working fluid	–

This function calculates the gas constant of the working fluid from the "fluid" variable that goes through the ports. It performs:

$$R_g = \frac{R_u}{M_m}$$

If the working fluid is not in a gas state, a FATAL ASSERT will appear.

2.5.7 Function cp

Inputs

Name	Type	Description	Units
fluid	IN REAL[ChemName]	Working fluid	–

Variables

Name	Type	Inital	Description	Units
Chem_cp	REAL[LiquidsGases]		Specific heat at constant pressure of each chemical ($c_{P,i}$)	$\frac{J}{kg \cdot K}$

This function calculates the specific heat at constant pressure of the working fluid from variable "fluid" that goes through the ports.

The **calorically perfect gas model** is used in the definition of all components (the ones related to gases) and functions of the library. Therefore, the specific heat at a constant pressure (c_P) is treated as if it were constant: $c_P \neq c_P(p,T)$

If the working fluid is a gas or liquid, it takes the value from the array "Chem_cp", whose values are shown in *Appendix B*.

For liquids, no distinction is made between specific heats. The specific heat of the liquid will be called c. Therefore, $c = c_P = c_V$, where c_V is the specific heat at constant volume.

If the working fluid is a combustion gas, it calculates the average specific heat at constant pressure of the mixture of gases by using this expression:

$$c_P = c_{P,P} \cdot Y_{Comb\_prod} + \sum_{i}^{LiquidsGases} c_{P,i} \cdot Y_i$$

Note that it is not possible to transmit several chemicals through the same port except if there has been combustion beforehead. Hence, the equation above is applicable in any case.

2.5.8 Function cv

Inputs

Name	Type	Description	Units
fluid	IN REAL[ChemName]	Working fluid	–

This function calculates the specific heat at constant volume of the working fluid from variable “fluid” that goes through the ports. It performs:

$$c_V = c_P - R_g$$

If the working fluid is not in a gas state, a FATAL ASSERT will appear.

2.5.9 Function gamma

Inputs

Name	Type	Description	Units
fluid	IN REAL[ChemName]	Working fluid	–

This function calculates the ratio of specific heats of the working fluid from variable “fluid” that goes through the ports. It performs:

$$\gamma = \frac{c_P}{c_V}$$

If the working fluid is not in a gas state, a FATAL ASSERT will appear.

2.5.10 Function FGAMMA

Inputs

Name	Type	Description	Units
fluid	IN REAL[ChemName]	Working fluid	−

This function is used to calculate the value of a constant, Γ(γ), which is recursively used in many equations along the program. The function is the following:

$$\Gamma(\gamma) = \sqrt{\gamma} \cdot \left(\frac{2}{\gamma + 1}\right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

If the working fluid is not in a gas state, a FATAL ASSERT will appear.

2.5.11 Function rho

Inputs

Name	Type	Description	Units
fluid	IN REAL[ChemName]	Working fluid	−

Variables

Name	Type	Inital	Description	Units

Name	Type	Inital	Description	Units
Chem_rho	REAL[Liquids]		Liquid state density of each liquid (ρ_i)	$\frac{kg}{m^3}$

This function sets the density of the working fluid from variable "fluid" that goes through the ports. It can only be used for liquids. If the working fluid is not in a liquid state, a FATAL ASSERT will appear.

Liquid properties (ρ, c, k, μ) are assumed as constant in the definition of all functions and components related to liquids of the library.

It takes the value of the density from the "Chem_rho" array, whose values are shown in *Appendix B*.

It is not possible to transmit different liquids through the same port. Hence, to set the density:

$$\rho = \sum_{i}^{Liquids} \rho_i \cdot Y_i$$

2.5.12 Function cond

Inputs

Name	Type	Description	Units
fluid	IN REAL[ChemName]	Working fluid	$-$

Variables

Name	Type	Inital	Description	Units
Chem_cond	REAL[Liquids]		Thermal conductivity of each liquid (k_i)	$\frac{W}{m \cdot K}$

This function sets the thermal conductivity of the working fluid from the “fluid” variable that goes through the ports. It can only be used for liquids. If the working fluid is not in a liquid state, a FATAL ASSERT will appear.

It takes the value of the thermal conductivity from the array “Chem_cond”, whose values are shown in *Appendix B*.

It is not possible to transmit different liquids through the same port. Hence, to set the thermal conductivity:

$$k = \sum_{i}^{Liquids} k_i \cdot Y_i$$

2.5.13 Function visc

Inputs

Name	Type	Description	Units
fluid	IN REAL[ChemName]	Working fluid	–

Variables

Name	Type	Inital	Description	Units
Chem_visc	REAL[Liquids]		Dynamic viscosity of each liquid (μ_i)	$Pa \cdot s$

This function sets the dynamic viscosity of the working fluid from the “fluid” variable that goes through the ports. It can only be used for liquids. If the working fluid is not in a liquid state, a FATAL ASSERT will appear.

It takes the value of the dynamic viscosity from the “Chem_visc” array, whose values are shown in *Appendix B*.

It is not possible to transmit different liquids through the same port. Hence, to set the dynamic viscosity:

$$\mu = \sum_{i}^{Liquids} \mu_i \cdot Y_i$$

2.5.14 Function hdc_fric

Inputs

Name	Type	Description	Units
D	IN REAL	Hydraulic diameter (D)	m
rug	IN REAL	Absolute rugosity (ε)	m
Re	IN REAL	Reynolds number (Re)	$-$

Variables

Name	Type	Inital	Description	Units
rey	REAL		Auxiliar variable	$-$
fric	REAL		Friction factor (f)	$-$
a	REAL		Auxiliar variable (a)	$-$
b	REAL		Auxiliar variable (b)	$-$

The “hdc_fric” function incorporates the evolution of the friction factor as a function of the local Reynolds number and the relative roughness.

The friction factor (f) is calculated by means of a simple correlation valid for laminar, turbulent and transient flow (Churchill, 1977):

$$f = 8 \cdot \left(\left(\frac{8}{Re} \right)^{12} + \frac{1}{(a+b)^{3/2}} \right)^{\frac{1}{12}}$$

where:

$$a = \left(2.457 \ln \frac{1}{\left(\frac{7}{\mathrm{Re}}\right)^{9} + 0.27\frac{\varepsilon}{D}}\right)^{16}$$

$$b = \left(\frac{37530}{Re}\right)^{16}$$

2.5.15 Function fluidP

Inputs

Name	Type	Description	Units
Combustion	IN BOOLEAN	TRUE if there is combustion and FALSE if either oxidant or fuel is lacking	$-$
fluid_O[ChemName]	IN REAL	Oxidant fluid	$-$
fluid_F[ChemName]	IN REAL	Fuel fluid	$-$
phi	IN REAL	Equivalence ratio	$-$
W_O	IN REAL	Oxidant mass flow	$\frac{kg}{s}$
W_F	IN REAL	Fuel mass flow	$\frac{kg}{s}$
W_F_st	IN REAL	Fuel mass flow that provides a stoichiometric mixture when combined with the oxidant mass flow	$\frac{kg}{s}$
M_P	IN REAL	Molar mass of the products using a stoichiometric mixture	$\frac{g}{mol}$
cp_P	IN REAL	Specific heat at constant pressure of the products using a stoichiometric mixture	$\frac{J}{kg \cdot K}$
fluid_P[ChemName]	OUT REAL	Combustion products fluid	$-$

This function obtains the combustion products fluid composition. To calculate variable “fluid_P” (see *Apendix F*), if “Combustion” is equal to TRUE:

$$Y_{P,i} = \frac{\max(1-\phi,0)\cdot Y_{O,i}\dot{m}_O + \max(\phi-1{,}0)\cdot Y_{F,i}\dot{m}_{Fst}}{\dot{m}_O + \dot{m}_F}$$

$$Y_{P,Comb\_prod} = \frac{(1-\max(1-\phi,0))\cdot(\dot{m}_O + \dot{m}_{Fst})}{\dot{m}_O + \dot{m}_F}$$

The last two components of the array (the ones with indexes “Comb_prod_cp” and “Comb_prod_M”) are set to $c_{P,P}$ and $M_{m,P}$, respectively.

Otherwise, if “Combustion” is equal to FALSE:

$$Y_{P,i} = \frac{Y_{O,i}\dot{m}_O + Y_{F,i}\dot{m}_F}{\dot{m}_O + \dot{m}_F}$$

$$Y_{P,Comb\_prod} = 0$$

In this case, the last two components of the array (the ones with indexes “Comb_prod_cp” and “Comb_prod_M”) are set to 0.

2.5.16 Function fluidG

Inputs

Name	Type	Description	Units
fluid_P[ChemName]	IN REAL	Combustion products	$-$
W_O	IN REAL	Oxidant mass flow	$\frac{kg}{s}$
W_IO	IN REAL	Inert mass flow through the oxidant port	$\frac{kg}{s}$
W_F	IN REAL	Fuel mass flow	$\frac{kg}{s}$
W_IF	IN REAL	Inert mass flow through the fuel port	$\frac{kg}{s}$
W	IN REAL	Total mass flow	$\frac{kg}{s}$
fluid[ChemName]	OUT REAL	Fluid variable	$-$

This function obtains the fluid composition through nozzle port. To compute the “fluid” variable of the outlet port:

$$Y_{out,i} = \frac{Y_{P,i}(\dot{m}_O + \dot{m}_F)}{\dot{m}_{inO} + \dot{m}_{inF}}$$

$$Y_{out,Comb\_prod} = \frac{Y_{P,Comb\_prod}(\dot{m}_O + \dot{m}_F) + \dot{m}_{IO} + \dot{m}_{IF}}{\dot{m}_{inO} + \dot{m}_{inF}}$$

The last two components of the array (the ones with indexes "Comb_prod_cp" and "Comb_prod_M") are set to the same value as the same components of variable "fluid_P". This is because, if there are two consecutive "GasGen" components, the introduced values of $c_{P,P}$ and $M_{m,P}$ will be the same because both will work with the same chemicals and $c_{P,P}$ and $M_{m,P}$ are the values for stoichiometric mixture. Therefore, they are properties of the chemicals used and they do not depend on OF.

2.5.17 Function GeopotentialAltitude

Inputs

Name	Type	Description	Units
z	IN REAL	Geometric altitude (z)	m

This function obtains the geopotential altitude (h) from the geometric altitude. The geopotential altitude is a vertical coordinate referenced to Earth's mean sea level (MSL), an adjustment to geometric altitude (elevation above MSL) using the variation of gravity with latitude and elevation. Thus it can be considered a "gravity-adjusted height". One usually speaks of the geopotential height of a certain pressure level, which would correspond to the geopotential height necessary to reach the given pressure. This function is used by the following functions, which calculate the International Standard Atmosphere (ISA).

At an elevation above MSL of z, the geopotential altitude, h, is defined as:

$$h = \frac{\int_0^z g(z)dz}{g_0}$$

Where:

- g_0 is the gravity acceleration at the Earth surface: $g_0 = G \cdot \frac{M_T}{{R_T}^2} = 9.80665 \frac{m}{s^2}$
- R_T is the Earth radius. A mean value has been used: $R_T = 6371\ km = 6371000\ m$
- M_T is the Earth mass: $M_T = 5.972 \cdot 10^{24}\ kg$

- G the gravitational constant: $G = 6.67384 \cdot 10^{-11}\ N \cdot \left(\frac{m}{kg}\right)^2$

The function $g(z)$ is the value of the gravity depending of the elevation above MSL:

$$g(z) = G \cdot \frac{M_T}{(R_T + z)^2}$$

The value of h is:

$$h = \frac{\int_0^z G \cdot \frac{M_T}{(R_T + \tilde{z})^2} d\tilde{z}}{G \cdot \frac{M_T}{{R_T}^2}} = \int_0^z \frac{{R_T}^2}{(R_T + \tilde{z})^2} d\tilde{z} = -\frac{{R_T}^2}{R_T + z}\Bigg]_{\tilde{z}=0}^{\tilde{z}=z} = -\frac{{R_T}^2}{R_T + z} - \left(-\frac{{R_T}^2}{R_T}\right)$$

The final function is the following:

$$h = \frac{z \cdot R_T}{R_T + z}$$

2.5.18 Function ISA_Temperature

Inputs

Name	Type	Description	Units
z	IN REAL	Geometric altitude (z)	m

Variables

Name	Type	Inital	Description	Units
T_amb	REAL	T_std	Ambient temperature (T_{amb})	K
a	REAL [8]	{-6.5, 0, 1, 2.8, 0, -2.8, -2, 0}	Thermal gradient in each atmosphere layer (a)	$\frac{K}{km}$
T_0	REAL	T_std	Base temperature of the atmosphere layer (T_0)	K

Name	Type	Inital	Description	Units
h_0	REAL	0	Base geopotential altitude of the atmosphere layer (h_0)	m
h_max	REAL [7]	{11000, 20000, 32000, 47000, 51000, 71000, 84852}	Maximum geopotential altitude in each atmosphere layer (h_{max})	m
i	INTEGER		Variable used as a index	
j	INTEGER	1	Variable used as a index	
h	REAL		Geopotential altitude (h)	m

The International Standard Atmosphere (ISA) is an atmospheric model of how the pressure, temperature, density, and viscosity of the Earth's atmosphere change over a wide range of altitudes or elevations. It has been established to provide a common reference for temperature and pressure and consists of tables of values at various altitudes, plus some formulas by which those values were derived. The International Organization for Standardization (ISO) publishes the ISA as an international standard (ISO 2533:1975).

The ISA assumptions are the followings:

- It works with calorically perfect gases, so **the state equation is the ideal-gas law**:

  $$\frac{p}{\rho} = R \cdot T$$

  where ρ is the air density, p the atmospheric pressure, T the air temperature and R the air constant, whose value is: $R = 287 \frac{J}{kg \cdot K}$

- Moreover, the pressure is due to the weight of the air layers above the altitude considered, posing a force balance in a differential volume element:

  $$p \cdot A - (p + dp) \cdot A - dW = 0$$

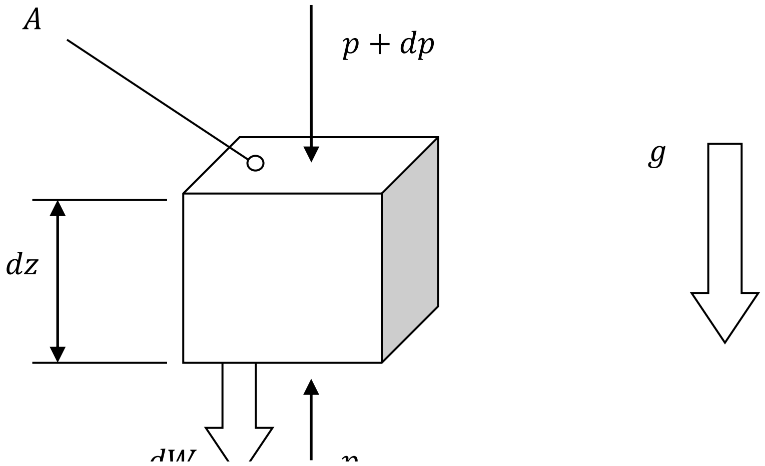


Figure 2- Force balance in an atmospheric fluid differential volume element

$$dW = gdm = g\rho dV = g\rho Adz$$

g is the gravity aceleration, W is the weight, m is the mass and V is the volume.

Substituting and solving:

$$-dp - g\rho dz = 0$$

It is better to use the geopotential altitude, h, instead of the geometrical altitude, z, and consider a constant gravity, g_0, which corresponds to the gravity acceleration at the Earth's surface. The change that it is necessary to do is:

$$gdz = g(z)dz = g_0 dh$$

So that the pressure variation can be expressed as:

$$\frac{dp}{dh} = -\rho \cdot g_0$$

In this function, the objective is to obtain just the temperature, but first, as it has been explained in the previous paragraph, it is necessary to convert the geometric altitude to the geopotential altitude because the input of the function is the first one. To do this, GeopotentialAltitude function is used.

The ISA model divides the atmosphere into layers with linear temperature distributions.

Layer	Level name	Base geometric altitude above MSL (z_0) [m]	Base geopotential altitude above MSL (h_0) [m]	Lapse rate (a) [K/km]
0	Troposphere	0	0	-6.5
1	Tropopause	11019	11000	0
2	Stratosphere	20063	20000	1
3	Stratosphere	32162	32000	2.8
4	Stratopause	47350	47000	0
5	Mesosphere	51413	51000	-2.8
6	Mesosphere	71802	71000	-2
7	Mesopause	86000	84852	

Layers in the ISA Standard Atmosphere 1976

$$h_0 = GeopotentialAltitude(z_0) \Rightarrow z_0 = \frac{h_0 \cdot R_T}{R_T - h_0}$$

The table above shows the lapse rate (or thermal gradient) in each layer to set the linear temperature distributions.Therefore, the ambient temperature (T_{amb}), is calculated with the following equation:

$$T_{amb} = T_0 + a \cdot (h - h_0)$$

Where $T_0 = T_{std}$ in the first layer and T_0 is the temperature when $h = h_{max}$ in the upper layers. The model is valid up to the mesopause (geometric altitude of $86000\ m$).

The result of this equation is shown in the following graph:

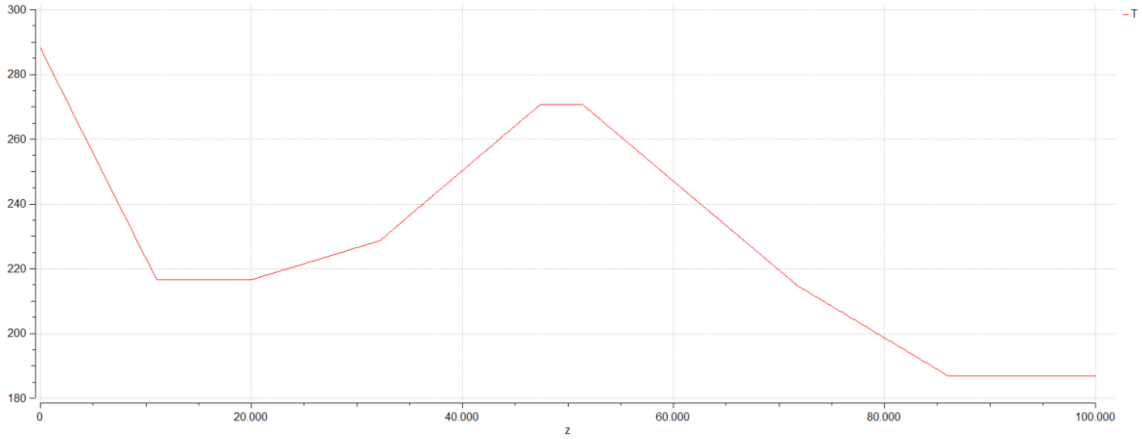


Figure 3 - Ambient temperature v.s. Geometrical altitude

2.5.19 Function ISA_Pressure

Inputs

Name	Type	Description	Units
z	IN REAL	Geometric altitude (z)	m

Variables

Name	Type	Inital	Description	Units
p_amb	REAL	p_std	Ambient pressure (p_{amb})	Pa
a	REAL [8]	{-6.5, 0, 1, 2.8, 0, -2.8, -2, 0}	Thermal gradient in each atmosphere layer (a)	$\frac{K}{km}$
b	BOOLEAN [8]	{FALSE, TRUE, FALSE, FALSE, TRUE, FALSE, FALSE, TRUE}	Variable that shows if the value of the temperature gradient is 0 (TRUE) or not (FALSE) in each atmosphere layer	
T_0	REAL	T_std	Base temperature of the atmosphere layer (T_0)	K
p_0	REAL	p_std	Base pressure of the atmosphere layer (p_0)	Pa
h_0	REAL	0	Base geopotential altitude of the atmosphere layer (h_0)	m
h_max	REAL [7]	{11000, 20000, 32000, 47000, 51000, 71000, 84852}	Maximum geopotential altitude in each atmosphere layer (h_{max})	m
i	INTEGER		Variable used as a index	
j	INTEGER	1	Variable used as a index	
h	REAL		Geopotential altitude (h)	m

This function, obtains the pressure using the International Standard Atmosphere model explained in the previous section.

The mathematical model is obtained from the next differential equation:

$$\frac{dp_{amb}}{dh} = -\frac{p_{amb}}{R \cdot T_{amb}} \cdot g_0$$

which is obtained from the two ISA assumptions. Introducing the T_{amb} linear equation:

$$\frac{dp_{amb}}{p_{amb}} = -\frac{g_0 dh}{R \cdot [T_0 + a \cdot (h - h_0)]}$$

Integrating it between h_0 and h and being the initial conditions $p_{amb} = p_0$ when $h = h_0$, it establishes a different equation to determine the ambient pressure (p_{amb}) according to the value of the thermal gradient (a):

$$p_{amb} = \begin{cases} p_0 \cdot e^{-\frac{g_0}{R \cdot T_0}(h-h_0)} & a = 0 \\ p_0 \cdot \left(\frac{T_0 + a \cdot (h - h_0)}{T_0}\right)^{-\frac{g_0}{R \cdot a}} & a \neq 0 \end{cases}$$

Where $p_0 = p_{std}$ in the first layer and p_0 is the temperature when $h = h_{max}$ in the upper layers.

The result of this equation is shown in the following graph:

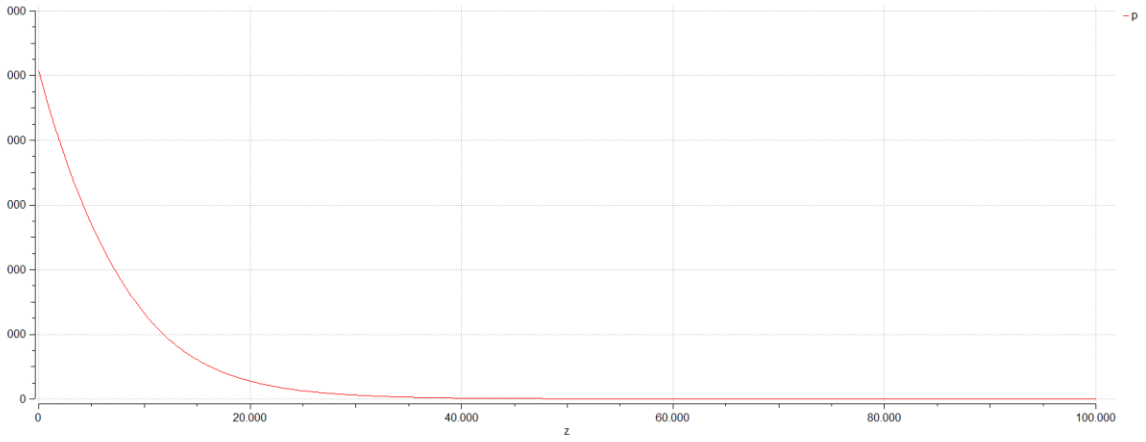


Figure 4 - Ambient pressure v.s. Geometrical altitude

Another possibility that could have been used instead of the ISA model in this function is isothermal atmosphere model, whose equation is:

$$p_{amb} = p_{std} \cdot e^{-\frac{z}{6898}}$$

The result of this equation is shown in the following graph in comparison with the ISA model:

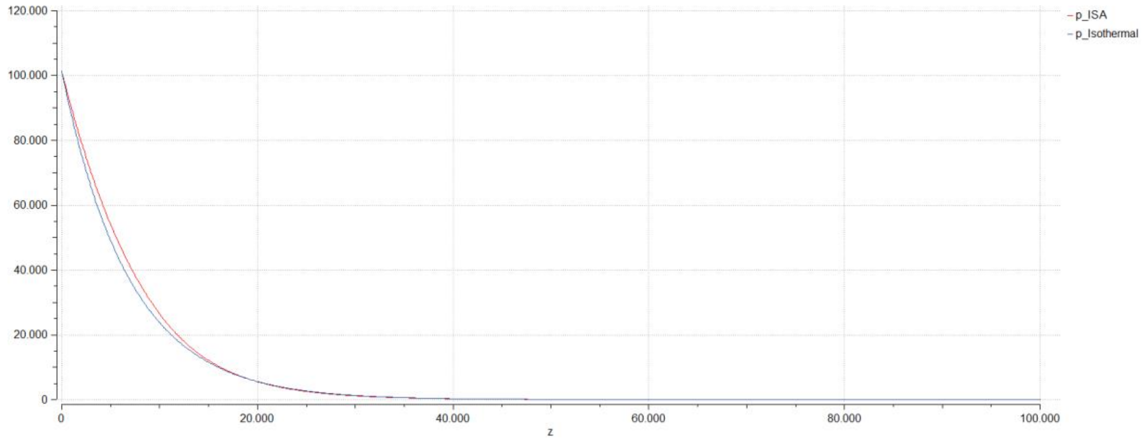


Figure 5 - Comparison between the ISA model and the Isothermal model

And the relative error, in parts per unit, between both models is:

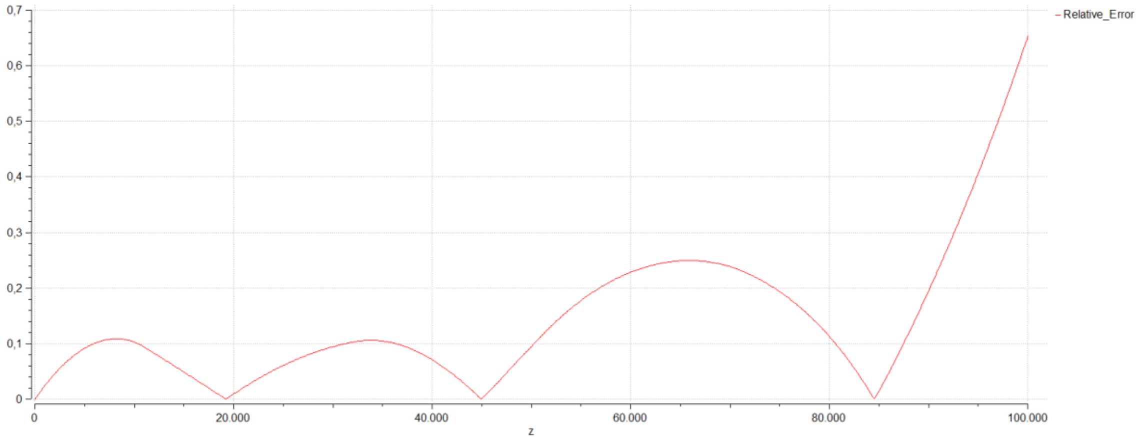


Figure 6 - Relative error committed by the Isothermal model

2.5.20 Function ISA_Density

Inputs

Name	Type	Description	Units
z	IN REAL	Geometric altitude (z)	m

Variables

Name	Type	Inital	Description	Units
rho_amb	REAL	p_std/(T_std*287)	Ambient density (ρ_{amb})	$\frac{kg}{m^3}$
a	REAL [8]	{-6.5, 0, 1, 2.8, 0, -2.8, -2, 0}	Thermal gradient in each atmosphere layer (a)	$\frac{K}{km}$
b	BOOLEAN [8]	{FALSE, TRUE, FALSE, FALSE, TRUE, FALSE, FALSE, TRUE}	Variable that shows if the value of the temperature gradient is 0 (TRUE) or not (FALSE) in each atmosphere layer	
T_0	REAL	T_std	Base temperature of the atmosphere layer (T_0)	K
rho_0	REAL	p_std/(T_std*287)	Base pressure of the atmosphere layer (ρ_0)	$\frac{kg}{m^3}$
h_0	REAL	0	Base geopotential altitude of the atmosphere layer (h_0)	m
h_max	REAL [7]	{11000, 20000, 32000, 47000, 51000, 71000, 84852}	Maximum geopotential altitude in each atmosphere layer (h_{max})	m
i	INTEGER		Variable used as a index	
j	INTEGER	1	Variable used as a index	
h	REAL		Geopotential altitude (h)	m

This function, obtains the density using the International Standard Atmosphere model explained in the previous section.

The mathematical model calculates the value of the air density from p_{amb} and T_{amb} using the ideal-gas law equation:

$$\rho_{amb} = \frac{p_{amb}}{R \cdot T_{amb}}$$

There is a different equation for determining the ambient density (ρ_{amb}) according to the value of the thermal gradient (a):

$$\rho_{amb} = \begin{cases} \rho_0 \cdot e^{-\frac{g_0}{R \cdot T_0} \cdot (h - h_0)} & a = 0 \\ \rho_0 \cdot \left(\frac{T_0 + a \cdot (h - h_0)}{T_0} \right)^{-\frac{g_0}{R \cdot a} - 1} & a \neq 0 \end{cases}$$

Where $\rho_0 = \frac{p_{std}}{R \cdot T_{std}}$ in the first layer and ρ_0 is the temperature when $h = h_{max}$ in the upper layers.

The result of this equation is shown in the following graph:

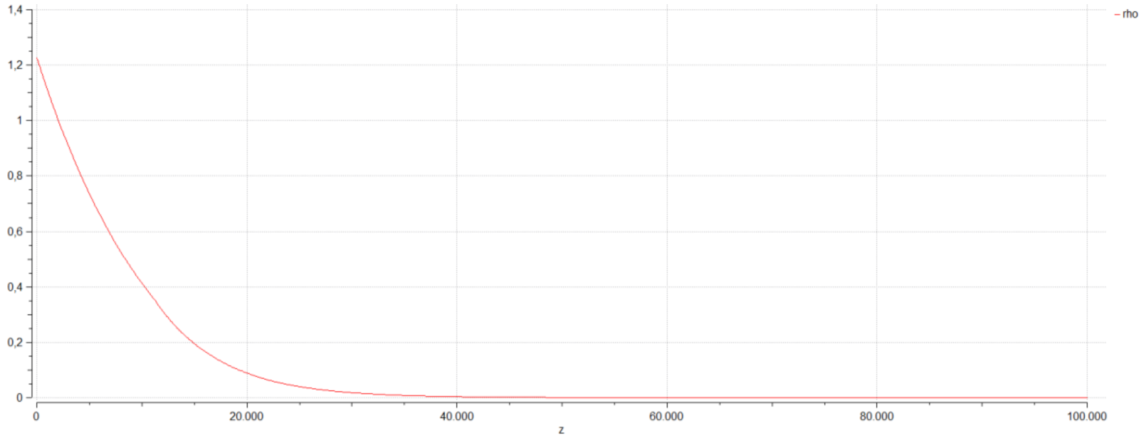


Figure 7 - Ambient density v.s. Geometrical altitude

3 Abstract Components

The library uses these kinds of components to simplify the definition of almost all library components because, some equations and ports are the same in most of them. Therefore, it decreases the number of lines written in the code.

3.1 Abstract Component FluidInFluidOut

It is defined in the LPRES_CompBasic.el file.

3.1.1 Description

It is used for the definition of components with a fluid inlet and a fluid outlet.

3.1.2 Ports

Name	Type	Direction	Description
f_in	Fluid	IN	Inlet fluid port
f_out	Fluid	OUT	Outlet fluid port

3.1.3 Formulation

Components defined as a “FluidInFluidOut” fulfil this equation:

$$\dot{m}_{in} = \dot{m}_{out}$$

which establishes that mass flow is constant in all their sections. The theoretical development can be followed in *Appendix C*.

This component also establishes that:

$$Y_{i,in} = Y_{i,out}$$

because it is assumed that there is no phase change in it. To establish it, the equation included is:

$$fluid_{in} = fluid_{out}$$

Components defined as “FluidInFluidOut” transmit the auxiliary variables of the “fluid” array between their ports.

3.2 Abstract Component GasTurbo

It is inherited from abstract component “GasInGasOut”.

It is defined in the LPRES_CompBasic.el file.

3.2.1 Description

It is used for the definition of gas turbomachinery components.

3.2.2 Ports

Name	Type	Direction	Description
m	Mechanical	IN	Inlet mechanical port

3.2.3 Variables

Name	Type	Inital	Description	Units
Power	REAL		Mechanical power (W)	W

3.2.4 Formulation

As shown in *Appendix D*:

$$W = \dot{m} \cdot c_P \cdot (T_{t,in} - T_{t,out})$$

In this way, W will be positive in turbines and negative in compressors.

The abstract component also sets the power of the inlet mechanical port to W:

$$W = W_{in}$$

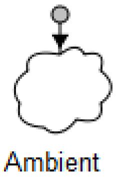
4 Components

4.1 Component Ambient

4.1.1 Description

This component type simulates a liquid discharge to the atmosphere.

It is defined in the LPRES_CompBasic.el file.



4.1.2 Construction Parameters

Name	Type	Default	Units
Type	ENUM OnOffDesign	Design	–

4.1.3 Ports

Name	Type	Direction	Description
1	Fluid	IN	Inlet fluid port

4.1.4 Data

Name	Type	Default	Description	Units	Condition
A	REAL	0.01	Discharge area (A)	m^2	Type=Off_Design

4.1.5 Variables

Name	Type	Inital	Description	Units
p_amb	REAL		Ambient pressure (p_{amb})	Pa
A_d	REAL		Design discharge area (A)	m^2

4.1.6 Formulation

This component establishes the outlet static pressure in a liquid discharge to the atmosphere.

This component uses the "ISA_Pressure" function to set the ambient pressure:

$$p_{amb} = ISA\_Pressure(z)$$

The port sends the total pressure, so it is necessary to calculate the static pressure in order to equalise it with the ambient pressure:

$$p_{t,in} = p_{amb} + \frac{1}{2 \cdot \rho}\left(\frac{\dot{m}}{A}\right)^2$$

For obtaining this equation: $\dot{m} = \rho v A$ as *Appendix C* shows.

On-off design mode

If Type=Off_Design, the discharge area (A) will be set as data and the component will calculate the mass flow. Otherwise, if Type=Design, the component will calculate the discharge area (A) and the mass flow will be set by another component.

To have a general overview, see *Appendix G*.

Error Messages

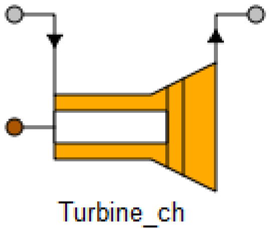
Only liquids can go through this component. If the working fluid is not in a liquid state, a FATAL ASSERT will appear.

4.2 Component Turbine_ch

4.2.1 Description

This component type represents a turbine with a choked inlet.

It is inherited from abstract component "GasTurbo". It is defined in the LPRES_CompBasic.el file.



4.2.2 Construction Parameters

Name	Type	Default	Units
Type	ENUM Type_Turbines	Known_pi	–

4.2.3 Data

Name	Type	Default	Description	Units	Condition
eta_d	REAL	0.5	Design efficiency (η_d)	–	
alpha_2	REAL	45	Flow angle in section 2 (α_2)	º	
A_in	REAL	0.001	Input area (A_{in})	m^2	Type=Off_Design
rpm	REAL	30000	Design rotational speed (rpm)	rpm	Type=Known_pi or Type=Known_W
pi	REAL	10	Design expansion ratio (π)	–	Type=Known_pi

Name	Type	Default	Description	Units	Condition
W	REAL	5	Design mass flow ($\dot{m}$)	$\frac{kg}{s}$	Type=Known_W

4.2.4 Variables

Name	Type	Inital	Description	Units
A_in_d	REAL		Design input area (A_{in})	m^2
eta	REAL		Efficiency (η)	$-$
alpha	REAL		Total temperature ratio (α)	$-$

4.2.5 Formulation

This component represents an axial turbine with a choked inlet. Its model is detailed below. An outline of the first stage turbine can be seen in the following figure:

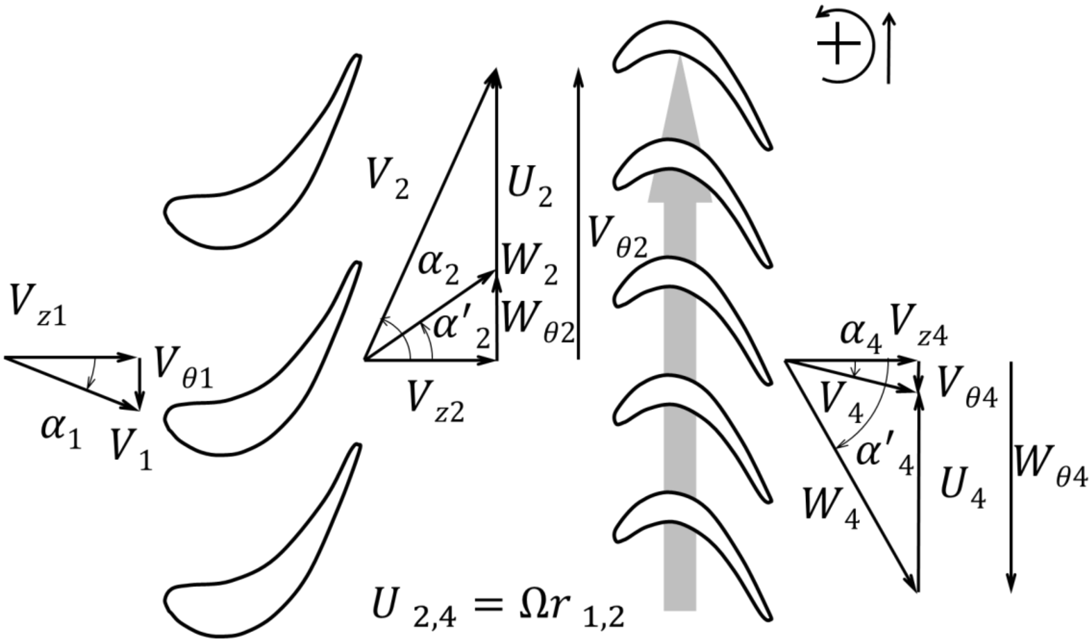


Figure 8 - Nomenclature used for the velocity triangles in the midline analysis of an axial turbine stage, comprised by a stator between stations 1 and 2 and by a rotor between 3 (= 2) and 4.

The total temperature ratio is defined as:

$$\alpha = \frac{T_{t,out}}{T_{t,in}}$$

and the specific work as:

$$\tau = \frac{W}{\dot{m}}$$

because the power is defined as positive in turbines since they provide energy to work.

The mass flow, $\dot{m}$, at the stator outlet (station 2) is:

$$\dot{m} = \rho_2 V_2 A_{in} \cos(\alpha_2) = \rho_2 M_2 a_2 A_{in} \cos(\alpha_2) =$$

$$= A_{in} \cos(\alpha_2) \cdot \rho_{t,in} a_{t,in} \cdot M_2 \left(\frac{1}{1 + \frac{\gamma - 1}{2} M_2^2} \right)^{\frac{\gamma+1}{2(\gamma-1)}} =$$

$$= A_{in} \cos(\alpha_2) \cdot \frac{p_{t,in}}{\sqrt{R_g T_{t,in}}} \cdot \sqrt{\gamma} M_2 \left(\frac{1}{1 + \frac{\gamma - 1}{2} M_2^2} \right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

Note that A_{in} is the inlet area, which is perpendicular to the axis of the turbine. Angle α_2 is constant regardless of the operating regime. Hence, it is a parameter of the turbine. The ideal-gas law has been used. **The ideal-gas law will be always used throughout the library when dealing with gases.**

To arrive at this equation, some equations which relate total and static variables for gases by means of the Mach number of the gas at that section have been used. They can be deducted from the energy equation with isentropic flux:

$$s = s_t \; ; \; ds = c_P \frac{dT}{T} - R_g \frac{dp}{p} \;\Rightarrow\; 0 = c_P ln \frac{T_t}{T} - R_g ln \frac{p_t}{p}$$

which leads to:

$$\frac{h}{h_t} = \frac{T}{T_t} = \left(\frac{a}{a_t} \right)^2 = \left(\frac{\rho}{\rho_t} \right)^{\gamma - 1} = \left(\frac{p}{p_t} \right)^{\frac{\gamma - 1}{\gamma}}$$

and from:

$$\frac{T}{T_t} = \frac{1}{1 + \frac{\gamma - 1}{2} M^2}$$

which comes from:

$$h = h_t - \frac{v^2}{2} \quad \Rightarrow \quad T = T_t - \frac{v^2}{2 \cdot c_P} \quad \Rightarrow$$

$$1 = \frac{T_t}{T} - \frac{v^2}{2c_P T} \quad \Rightarrow \quad \frac{T_t}{T} = 1 + \frac{v^2}{2 \cdot T}\frac{(\gamma - 1)}{\gamma R_g} \quad ; \quad M^2 = \frac{v^2}{\gamma R_g T}$$

The critical section of the turbine is located at the stator outlet. The model of this turbine imposes that $M_2 = 1$ which means that **the turbine is choked**. Hence:

$$\dot{m} = A_{in}\cos(\alpha_2) \cdot \frac{p_{t,in}}{\sqrt{R_g T_{t,in}}} \cdot \sqrt{\gamma}\left(\frac{2}{\gamma + 1}\right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

And taking into account the definition of function $\Gamma(\gamma)$:

$$\frac{\dot{m}\sqrt{R_g T_{t,in}}}{p_{t,in}A_{in}} = \cos(\alpha_2) \cdot \Gamma(\gamma)$$

The turbine evolution is considered adiabatic and stationary (the residence time is much lower than the characteristic heat transfer time). Therefore, a description of losses can be made as the adiabatic efficiency, which is defined as the ratio between the power obtained and the power that would be obtained in an ideal process (constant entropy) expanding the fluid between the same pressures:

$$\eta = \frac{(\tau)_{real}}{(\tau)_{ideal}} = \frac{h_{t,out} - h_{t,in}}{\left(h_{t,out}\right)_{ideal} - h_{t,in}} \approx \frac{T_{t,out} - T_{t,in}}{\left(T_{t,out}\right)_{ideal} - T_{t,in}} = \frac{1 - \frac{T_{t,out}}{T_{t,in}}}{1 - \frac{\left(T_{t,out}\right)_{ideal}}{T_{t,in}}}$$

In an ideal situation, the entropy is constant:

$$\frac{\left(T_{t,out}\right)_{ideal}}{T_{t,in}} = \left(\frac{p_{t,out}}{p_{t,in}}\right)^{\frac{\gamma-1}{\gamma}}$$

Hence:

$$\eta = \frac{1 - \alpha}{1 - \left(\frac{p_{t,out}}{p_{t,in}}\right)^{\frac{\gamma-1}{\gamma}}}$$

Finally, $\eta = \eta_d$ is the performance equation for the efficiency. This kind of equations, in which the variable has the same value as the design variable, is very common throughout the library because they are thought to be improved in future developments of the library.

Be careful not to use this component if the turbine is not choked because the results will be wrong.

On-off design mode

If Type=Known_pi, the component will set the rotational speed (rpm) and the expansion ratio as data, where:

$$\pi = \frac{p_{t,in}}{p_{t,out}}$$

and it will calculate the mass flow and the inlet area (A_{in}). The power will be set by the component connected to the mechanical port.

If Type=Known_W, the component will set the rotational speed (rpm) and the mass flow as data, and it will calculate the expansion ratio and the inlet area (A_{in}). Once again, the power will be set by the component connected to the mechanical port.

Otherwise, if Type=Off_Design, the inlet area (A_{in}) will be set as data and the component will calculate the power and the expansion ratio. The mass flow will be imposed by another component. In this case, the rotational speed will be set by the component connected to the mechanical port.

To have a general overview, see *Appendix G*.

Error Messages

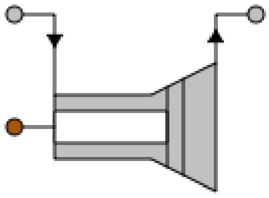
Only gases can go through this component. If the working fluid is not in a gas state, a FATAL ASSERT will appear.

4.3 Component Turbine

4.3.1 Description

This component type represents a turbine (the inlet may be choked or not).

It is inherited from abstract component "GasTurbo". It is defined in the LPRES_CompBasic.el file.



Turbine

4.3.2 Construction Parameters

Name	Type	Default	Units
Type	ENUM Type_Turbines	Known_pi	–
Type_AC	ENUM AngCoef	Coefficients	–

4.3.3 Data

Name	Type	Default	Description	Units	Condition
eta_d	REAL	0.8	Design efficiency (η_d)	–	
phi_d	REAL	0.05	Design flow coefficient if Type_AC=Coefficients (ϕ_d)	–	Type_AC=Coefficients
psi_d	REAL	0.7	Design loading coefficient if Type_AC=Coefficients (ψ_d)	–	Type_AC=Coefficients
alpha_2	REAL	45	Flow angle in section 2 (α_2)	º	
alpha_4r	REAL	-30	Relative flow angle in section 4 if Type_AC=Angles (α'_4)	º	Type_AC=Angles
M	REAL	1	Design Mach number in section 2 if Type_AC=Angles and Type=Design or initial Mach number in section 2 for iterative calculations (M_2)	–	
A_in	REAL	0.005	Input area (A_{in})	m^2	Type=Off_Design
r_m	REAL	0.01	Average radius (r_m)	m	Type=Off_Design
rpm	REAL	30000	Design rotational speed (rpm)	rpm	Type=Known_pi or Type=Known_W
pi	REAL	1.5	Design expansion ratio (π)	–	Type=Known_pi
W	REAL	5	Design mass flow ($\dot{m}$)	$\frac{kg}{s}$	Type=Known_W
U_0	REAL	10000	Initial blade speed for iterative calculations	$\frac{m}{s}$	

4.3.4 Variables

Name	Type	Inital	Description	Units
A_in_d	REAL		Design input area (A_{in})	m^2
r_m_d	REAL		Design average radius (r_m)	m
eta	REAL		Efficiency (η)	$-$
psi	REAL		Loading coefficient (ψ)	$-$
phi	REAL		Flow coefficient (ϕ)	$-$
U	ALG REAL	U_0	Blade speed (U)	$\frac{m}{s}$
alpha	REAL		Total temperature ratio (α)	$-$
tau	REAL		Specific work (τ)	$\frac{J}{kg}$
M_2	REAL	M	Mach number in section 2 (M_2)	$-$
V_z2	REAL		Axial speed in section 2 (V_{z2})	$\frac{m}{s}$
V_2	REAL		Speed in section 2 (V_2)	$\frac{m}{s}$

4.3.5 Formulation

This component represents an axial turbine. Its inlet may be choked or not. Its model is detailed below. The outline of the turbine is the same as that represented for component “Turbine_ch”.

The total temperature ratio is defined as:

$$\alpha = \frac{T_{t,out}}{T_{t,in}}$$

and the specific work as:

$$\tau = \frac{W}{\dot{m}}$$

because the power is defined as positive in turbines since they provide energy to work.

The mass flow, $\dot{m}$, at the stator outlet (station 2) is:

$$\dot{m} = \rho_2 V_2 A_{in}\cos(\alpha_2) = \rho_2 M_2 a_2 A_{in}\cos(\alpha_2) =$$

$$= A_{in}\cos(\alpha_2)\cdot \rho_{t,in} a_{t,in}\cdot M_2\left(\frac{1}{1+\frac{\gamma-1}{2}M_2^2}\right)^{\frac{\gamma+1}{2(\gamma-1)}} =$$

$$= A_{in}\cos(\alpha_2)\cdot \frac{p_{t,in}}{\sqrt{R_g T_{t,in}}}\cdot \sqrt{\gamma} M_2\left(\frac{1}{1+\frac{\gamma-1}{2}M_2^2}\right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

The relationship between V_2 and M_2 is:

$$V_2 = M_2 \cdot \sqrt{\gamma R_g \frac{T_{t,in}}{1+\frac{\gamma-1}{2}M_2^2}}$$

Besides, a geometrical relation can be set between V_2 and V_{z2}:

$$V_{z2} = V_2 \cdot \cos(\alpha_2)$$

Coefficients ψ and ϕ are:

$$\psi = \frac{\tau}{U^2}$$

$$\phi = \frac{V_{z2}}{U}$$

where

$$U = \omega \cdot r_m$$

The inlet does not have to be choked. Hence, it needs a broader equation to substitute the equation that appeared in component “Turbine_ch” which sets M_2 to 1. This equation is the performance equation which comes from the Euler equation of turbomachines. For a single stage-turbine:

$$-\tau = U_4 V_{\theta 4} - U_2 V_{\theta 2}$$

Assuming $U = U_2 = U_4$ (which is true if $r_m = r_2 = r_4$) and $V_{z2} = V_{z4}$ and with the definitions of ψ and ϕ, the Euler equation of turbomachines is as follows:

$$\psi = \phi(\tan\alpha_2 - \tan\alpha_4)$$

Besides, some geometrical relations can be applied:

$$V_{\theta 4} = W_{\theta 4} + U_4 \quad \Rightarrow \quad V_{z2} \tan\alpha_4 = V_{z2}\tan\alpha_4' + U \quad \Rightarrow \quad 1 = \phi(\tan\alpha_4 - \tan\alpha_4')$$

This relation can be checked in the following figure:

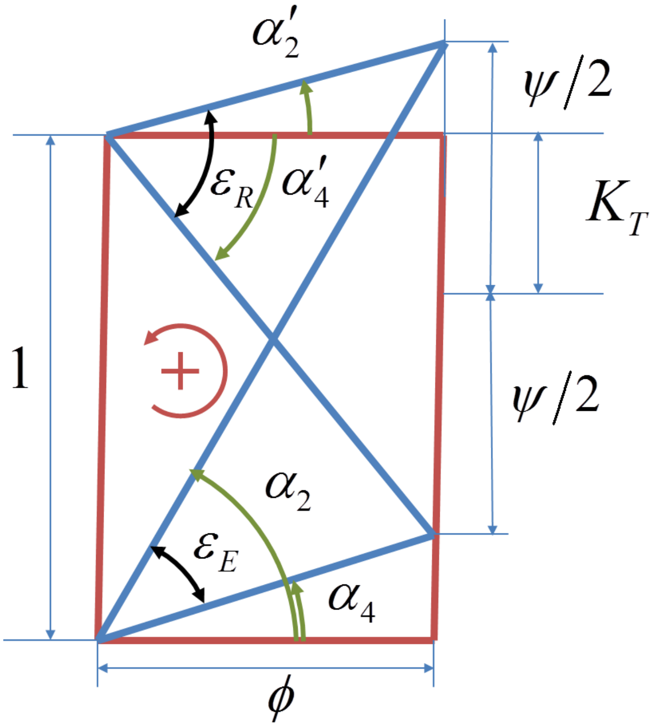


Figure 9 - Geometric representation of the relationship between the load parameter, the flow parameter and the degree of reaction when defining velocity triangles.

Finally, the performance equation is:

$$\psi = \phi\left(\tan\alpha_2 - \tan\alpha_4'\right) - 1$$

Angles α_2 and α_4' are constant regardless of the operation regime. Hence, they are parameters of the turbine and that was why they were selected. The previous performance equation is used if Type_AC=Angles.

This component has the possibility, if Type_AC=Coefficients, of working with the performance equation written in the following way, which is also valid for a multi-stage turbine:

$$\psi = \frac{1-\psi_d}{\phi_d}\phi - 1$$

where ϕ_d the design flow coefficient and ψ_d the design loading coefficient given as data.

And for the efficiency, the performance equation is:

$$\eta = \eta_d$$

The turbine evolution is considered adiabatic and stationary (the residence time is much lower than the characteristic heat transfer time). Therefore, a description of losses can be made as the adiabatic efficiency, which is defined as the ratio between the power obtained and the power that would be obtained in an ideal process (constant entropy) expanding the fluid between the same pressures:

$$\eta = \frac{(\tau)_{real}}{(\tau)_{ideal}} = \frac{h_{t,in} - h_{t,out}}{h_{t,in} - \left(h_{t,out}\right)_{ideal}} \approx \frac{T_{t,in} - T_{t,out}}{T_{t,in} - \left(T_{t,out}\right)_{ideal}} = \frac{1 - \frac{T_{t,out}}{T_{t,in}}}{1 - \frac{\left(T_{t,out}\right)_{ideal}}{T_{t,in}}}$$

In an ideal situation, the entropy is constant:

$$\frac{\left(T_{t,out}\right)_{ideal}}{T_{t,in}} = \left(\frac{p_{t,out}}{p_{t,in}}\right)^{\frac{\gamma-1}{\gamma}}$$

Hence:

$$\eta = \frac{1 - \alpha}{1 - \left(\frac{p_{t,out}}{p_{t,in}}\right)^{\frac{\gamma-1}{\gamma}}}$$

If it is known that the inlet of this component will be choked, it would be preferable to use component "Turbine_ch" because it is simpler. Besides, if the inlet is choked, the calculation of r_m in design mode is not representative because the performance equation is not being accomplished in that case; and if the mode is off-design, r_m will not be used.

If data "M", which is used to set M_2 in design mode, is bigger than 1, the value taken will be 1.

On-off design mode

If Type=Known_pi, the component will set the rotational speed (rpm) and the expansion ratio as data, where:

$$\pi = \frac{p_{t,in}}{p_{t,out}}$$

and it will calculate the mass flow, the inlet area (A_{in}) and the average radius (r_m). The power will be set by the component connected to the mechanical port.

If Type=Known_W, the component will set the rotational speed (rpm) and the mass flow as data, and it will calculate the expansion ratio, the inlet area (A_{in}) and the average radius (r_m). Once again, the power will be set by the component connected to the mechanical port.

Moreover, in design mode (Type=Known_pi or Type=Known_W) if Type_AC=Angles, the component will set M_2 as data. Otherwise, if Type_AC=Coefficients, there is an extra equation which matches the flow coefficient to the design flow coefficient given as data in order to fix the design condition:

$$\phi = \phi_d$$

If Type=Off_Design, the inlet area (A_{in}) and the average radius (r_m) will be set as data and the component will calculate the power, the expansion ratio and M_2. The mass flow will be imposed by another component. In this case, the rotational speed will be set by the component connected to the mechanical port.

To have a general overview, see *Appendix G*.

Error Messages

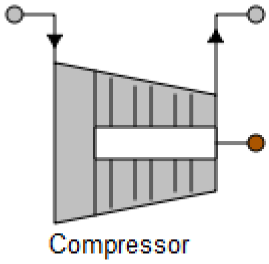
Only gases can go through this component. If the working fluid is not in a gas state, a FATAL ASSERT will appear.

4.4 Component Compressor

4.4.1 Description

This component type represents a compressor.

It is inherited from abstract component “GasTurbo”. It is defined in the LPRES_CompBasic.el file.



4.4.2 Construction Parameters

Name	Type	Default	Units
Type	ENUM OnOffDesign	Design	–
Type_AC	ENUM AngCoef	Coefficients	–

4.4.3 Data

Name	Type	Default	Description	Units	Condition
eta_d	REAL	0.8	Design efficiency (η_d)	–	
phi_d	REAL	0.05	Design flow coefficient (ϕ_d)	–	
psi_d	REAL	0.7	Design loading coefficient if Type_AC=Coefficients (ψ_d)	–	Type_AC=Coefficients
alpha_2r	REAL	-30	Relative flow angle in section 2 if Type_AC=Angles (α'_2)	º	Type_AC=Angles
r_m	REAL	0.3	Average radius (r_m)	m	Type=Off_Design
A_in	REAL	0.35	Input area (A_{in})	m^2	Type=Off_Design
pi	REAL	4	Design compression ratio (π)	–	Type=Design
U_0	REAL	500	Initial blade speed for iterative calculations	$\frac{m}{s}$	

4.4.4 Variables

Name	Type	Inital	Description	Units
A_in_d	REAL		Design inlet area (A_{in})	m^2
r_m_d	REAL		Design average radius (r_m)	m
eta	REAL		Efficiency (η)	–
psi	REAL		Loading coefficient (ψ)	–
phi	REAL		Flow coefficient (ϕ)	–
U	ALG REAL	U_0	Blade speed (U)	$\frac{m}{s}$
tau	REAL		Specific work (τ)	$\frac{J}{kg}$
rho_in	REAL		Inlet density (ρ_{in})	$\frac{kg}{m^3}$

Name	Type	Inital	Description	Units
M_in	ALG REAL	0.001	Inlet Mach number (M_{in})	–

4.4.5 Formulation

This component represents an axial compressor. Its model is detailed below. An outline of the compressor can be seen in the following figure:

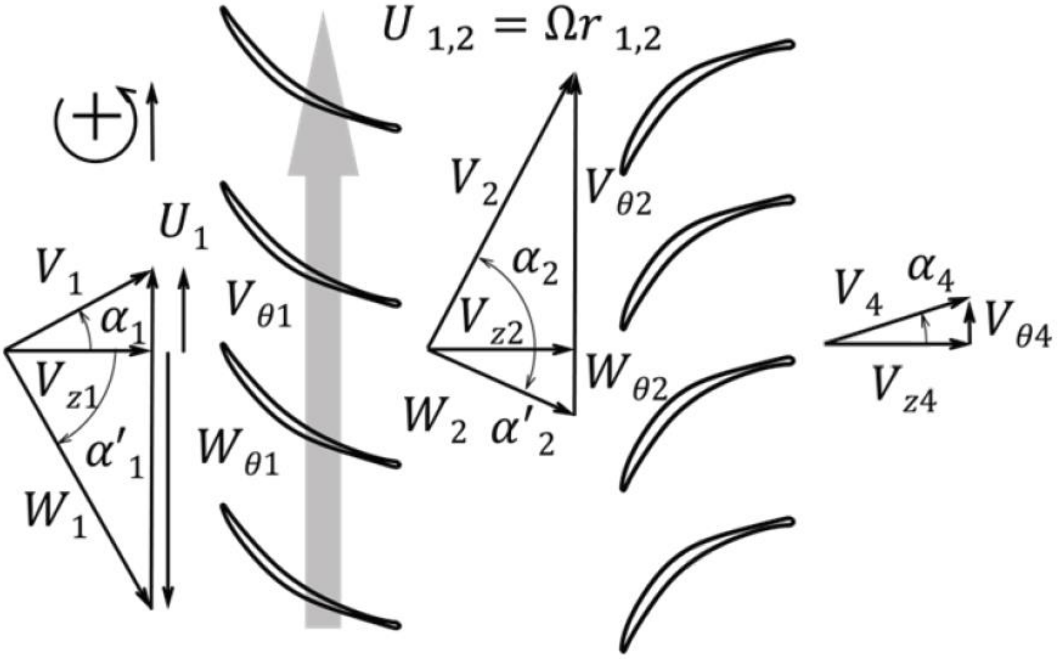


Figure 10 - Nomenclature used for the velocity triangles in the midline analysis of an axial compressor stage, comprised by a rotor between stations 1 and 2 and by a stator between 3 (= 2) and 4.

The the specific work is defined as:

$$\tau = -\frac{W}{\dot{m}}$$

because the power is defined as negative in compressors since they require external energy to work.

The mass flow equation is:

$$\dot{m} = \rho_{in} V_{in} A_{in} = \rho_{in} M_{in} a_{in} A_{in} =$$

$$= A_{in} \rho_{t,in} a_{t,in} M_{in} \left(\frac{1}{1 + \frac{\gamma - 1}{2} M_{in}^2} \right)^{\frac{\gamma+1}{2(\gamma-1)}} =$$

$$= \frac{A_{in} p_{t,in}}{\sqrt{R_g T_{t,in}}} \cdot \sqrt{\gamma} M_{in} \left(\frac{1}{1 + \frac{\gamma - 1}{2} M_{in}^2} \right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

From this equation M_{in} is calculated by means of iterations. Knowing M_{in}, the inlet density can be calculated as:

$$\left(\frac{\rho_{t,in}}{\rho_{in}} \right)^{\gamma-1} = 1 + \frac{\gamma - 1}{2} M_{in}^2 \quad ; \quad \rho_{t,in} = \frac{p_{t,in}}{R_g T_{t,in}}$$

The inlet density is used to the definition of the coefficients ψ and ϕ, which are:

$$\psi = \frac{\tau}{U^2}$$

$$\phi = \frac{\dot{m}}{\rho_{in} A_{in} U} = \frac{V_{in}}{U}$$

where

$$U = \omega \cdot r_m$$

Note that $V_{in} = V_1 = \frac{\dot{m}}{\rho_{in} A_{in}}$.

To obtain the performance equation, the Euler equation of turbomachines is considered for the rotor of a single stage-turbine:

$$\tau = U_2 V_{\theta 2} - U_1 V_{\theta 1}$$

One can assume $U = U_1 = U_2$ (which is true if $r_m = r_1 = r_2$), $V_{z1} = V_{z2}$ and $\alpha_1 = 0$, which implies $V_{in} = V_{z1}$. These assumptions together with the definitions of ψ and ϕ make the Euler equation of turbomachines to be:

$$\psi = \phi(\tan \alpha_2 - \tan \alpha_1) = \phi \tan \alpha_2$$

Besides, some geometrical relations can be applied:

$$V_{\theta 2} = W_{\theta 2} + U_2 \;\; \Rightarrow \;\; V_{z1} \tan \alpha_2 = V_{z1} \tan \alpha_2' + U_2 \;\; \Rightarrow \;\; 1 = \phi(\tan \alpha_2 - \tan \alpha_2^{'})$$

Finally, the performance equation is:

$$\psi = \phi \tan \alpha_2^{'} + 1$$

Angle α_2' is constant regardless of the operating regime. Hence, it is a parameter of the compressor and that was why it was selected. The previous performance equation is used if Type_AC=Angles.

This component has the possibility, if Type_AC=Coefficients, of working with the performance equation written in the following way, which is also valid for a multi-stage compressor:

$$\psi = 1 - \frac{1 - \psi_d}{\phi_d} \phi$$

where ϕ_d the design flow coefficient and ψ_d the design loading coefficient given as data.

And for the efficiency, the performance equation is:

$$\eta = \eta_d$$

The compressor evolution is considered adiabatic and stationary (the residence time is much lower than the one characteristic of heat transmission). Therefore, a description of losses can be made as the adiabatic efficiency which is defined as the ratio between the power that would be required in an ideal process (constant entropy) and the power required to compress the fluid between the same pressures:

$$\eta = \frac{(\tau)_{ideal}}{(\tau)_{real}} = \frac{(h_{t,out})_{ideal} - h_{t,in}}{h_{t,out} - h_{t,in}} \approx \frac{(T_{t,out})_{ideal} - T_{t,in}}{T_{t,out} - T_{t,in}} = \frac{\frac{(T_{t,out})_{ideal}}{T_{t,in}} - 1}{\frac{T_{t,out}}{T_{t,in}} - 1}$$

In an ideal situation, the entropy is constant:

$$\frac{(T_{t,out})_{ideal}}{T_{t,in}} = \left(\frac{p_{t,out}}{p_{t,in}}\right)^{\frac{\gamma-1}{\gamma}}$$

Hence:

$$\eta = \frac{\left(\frac{p_{t,out}}{p_{t,in}}\right)^{\frac{\gamma-1}{\gamma}} - 1}{\frac{T_{t,out}}{T_{t,in}} - 1}$$

On-off design mode

If Type=Design, the component will set the compression ratio as data, where:

$$\pi = \frac{p_{t,out}}{p_{t,in}}$$

and it will calculate the inlet area (A_{in}) and the power. The rotational speed will be set by the turbine connected to the mechanical port.

Moreover, in design mode, there is an extra equation which matches the flow coefficient to the design flow coefficient given as data in order to fix the design condition:

$$\phi = \phi_d$$

Otherwise, if Type=Off_Design, the inlet area (A_{in}) will be set as data and the component will calculate the rotational speed and the compression ratio. In this case, the power will be set by the turbine connected to the mechanical port.

The mass flow will be set by another component in both cases.

To have a general overview, see *Appendix G*.

Error Messages

Only gases can go through this component. If the working fluid is not in a gas state, a FATAL ASSERT will appear.

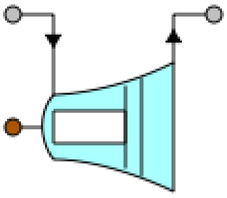
If the compressor inlet is choked, a KILLPOINT ASSERT will appear.

4.5 Component Turbine_liq

4.5.1 Description

This component type represents a liquid turbine.

It is inherited from abstract component “FluidInFluidOut”. It is defined in the LPRES_CompBasic.el file.



Turbine_liq

4.5.2 Construction Parameters

Name	Type	Default	Units
Type	ENUM Type_Turbine_liq	Known_dp	−

4.5.3 Ports

Name	Type	Direction	Description
m	Mechanical	IN	Inlet mechanical port

4.5.4 Data

Name	Type	Default	Description	Units	Condition
eta_d	REAL	0.8	Design efficiency (η_d)	−	
phi_d	REAL	0.05	Design flow coefficient (ϕ_d)	−	
psi_d	REAL	0.7	Design loading coefficient (ψ_d)	−	
A_in	REAL	0.01	Input area (A_{in})	m^2	Type=Off_Design
r_m	REAL	0.05	Average radius (r_m)	m	Type=Off_Design
rpm	REAL	30000	Design rotational speed (rpm)	rpm	Type=Known_dp or Type=Known_W
dp	REAL	5000000	Design pressure decrease (dp)	Pa	Type=Known_dp
W	REAL	5	Design mass flow ($\dot{m}$)	$\frac{kg}{s}$	Type=Known_W
U_0	REAL	500	Initial blade speed for iterative calculations	$\frac{m}{s}$	

4.5.5 Variables

Name	Type	Inital	Description	Units
A_in_d	REAL		Design inlet area (A_{in})	m^2
r_m_d	REAL		Design average radius (r_m)	m
eta	REAL		Efficiency (η)	$-$
psi	REAL		Loading coefficient (ψ)	$-$
phi	REAL		Flow coefficient (ϕ)	$-$
U	ALG REAL	U_0	Blade speed (U)	$\frac{m}{s}$
tau	REAL		Specific work (τ)	$\frac{J}{kg}$
Power	REAL		Mechanical power (W)	W

4.5.6 Formulation

This component represents a liquid turbine. Its model is detailed below.

The model used for pumps is similar to the one used for the “Turbine” component but suitable for liquids. The specific work is:

$$\tau = \frac{W}{\dot{m}}$$

because the power is defined as positive in turbines since they provide energy to work.

Coefficients ψ and ϕ are:

$$\psi = \frac{\tau}{U^2}$$

$$\phi = \frac{\dot{m}}{A_{in} \cdot U \cdot \rho}$$

where

$$U = \omega \cdot r_m$$

With coefficients ψ and ϕ a performance equation can be obtained:

$$\psi = 1 - \frac{1-\psi_d}{\phi_d}\phi$$

where ϕ_d the design flow coefficient and ψ_d the design loading coefficient given as data.

And for the efficiency, the performance equation is:

$$\eta = \eta_d$$

Following the development done in *Appendix D*, but in this case for perfect liquids (**the liquids with which the library works are perfect liquids: ideal liquids and calorically perfect**), the energy equation of a liquid turbine is:

$$W = \dot{m}\left(c(T_{in} - T_{out}) + \frac{p_{t,in} - p_{t,out}}{\rho}\right)$$

And with the definition of the specific work:

$$\tau = -c\Delta T - \frac{\Delta p_t}{\rho}$$

The efficiency is defined as the ratio between the power that would be required in an ideal process (constant entropy) and the power required in a real situation:

$$\eta = \frac{(\tau)_{ideal}}{(\tau)_{real}}$$

In an ideal situation, the entropy is constant and, consequently for perfect liquids, $\Delta T = 0$. Hence:

$$(\tau)_{ideal} = -\frac{\Delta p_t}{\rho}$$

And the efficiency results:

$$\eta = \frac{\tau}{(p_{t,in} - p_{t,out})/\rho}$$

Finally, the energy equation can be rewritten by using the efficiency, yielding:

$$c\Delta T = -\tau - \frac{\Delta p_t}{\rho} = \eta\frac{\Delta p_t}{\rho} - \frac{\Delta p_t}{\rho} = \frac{\Delta p_t}{\rho}(\eta - 1) = (1-\eta)\frac{p_{t,in} - p_{t,out}}{\rho}$$

The component also sets the power of the inlet mechanical port to W:

$$W = W_{in}$$

On-off design mode

If Type=Known_dp, the component will set the rotational speed (rpm) and the pressure decrease as data, where:

$$dp = p_{t,in} - p_{t,out}$$

and it will calculate the mass flow, the inlet area (A_{in}) and the average radius (r_m). The power will be set by the component connected to the mechanical port.

If Type=Known_W, the component will set the rotational speed (rpm) and the mass flow as data, and it will calculate the pressure decrease, the inlet area (A_{in}) and the average radius (r_m). Once again, the power will be set by the component connected to the mechanical port.

Moreover, in design mode (Type=Known_pi or Type=Known_W), there is an extra equation which matches the flow coefficient to the design flow coefficient given as data in order to fix the design condition:

$$\phi = \phi_d$$

If Type=Off_Design, the inlet area (A_{in}) and the average radius (r_m) will be set as data and the component will calculate the power and the pressure decrease. The mass flow will be imposed by another component. In this case, the rotational speed will be set by the component connected to the mechanical port.

To have a general overview, see *Appendix G*.

Error Messages

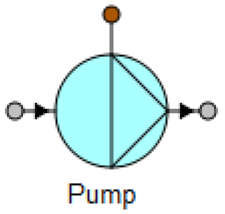
Only liquids can go through this component. If the working fluid is not in a liquid state, a FATAL ASSERT will appear.

4.6 Component Pump

4.6.1 Description

This component type represents a pump.

It is inherited from abstract component “FluidInFluidOut”. It is defined in the LPRES_CompBasic.el file.



4.6.2 Construction Parameters

Name	Type	Default	Units
Type	ENUM OnOffDesign	Design	–

4.6.3 Ports

Name	Type	Direction	Description
m	Mechanical	IN	Inlet mechanical port

4.6.4 Data

Name	Type	Default	Description	Units	Condition
eta_d	REAL	0.8	Design efficiency (η_d)	–	
phi_d	REAL	0.05	Design flow coefficient (ϕ_d)	–	
psi_d	REAL	0.7	Design loading coefficient (ψ_d)	–	
A_in	REAL	0.01	Input area (A_{in})	m^2	Type=Off_Design
r_m	REAL	0.05	Average radius (r_m)	m	Type=Off_Design
dp	REAL	5000000	Design pressure increase (dp)	Pa	Type=Design
U_0	REAL	500	Initial blade speed for iterative calculations	$\frac{m}{s}$	

4.6.5 Variables

Name	Type	Inital	Description	Units
A_in_d	REAL		Design inlet area (A_{in})	m^2
r_m_d	REAL		Design average radius (r_m)	m
eta	REAL		Efficiency (η)	$-$
psi	REAL		Loading coefficient (ψ)	$-$
phi	REAL		Flow coefficient (ϕ)	$-$
U	ALG REAL	U_0	Blade speed (U)	$\frac{m}{s}$
H	REAL		Head (H)	m
tau	REAL		Specific work (τ)	$\frac{J}{kg}$
Power	REAL		Mechanical power (W)	W

4.6.6 Formulation

This component represents a pump. Its model is detailed below.

The model used for pumps is similar to the one used for the “Compressor” component but suitable for liquids. The specific work is:

$$\tau = -\frac{W}{\dot{m}}$$

because the power is defined as negative in pumps since they require external energy to work.

Coefficients ψ and ϕ are:

$$\psi = \frac{\tau}{U^2}$$

$$\phi = \frac{\dot{m}}{A_{in} \cdot U \cdot \rho}$$

where

$$U = \omega \cdot r_m$$

The main tools for selecting the pump size are 2 non-dimensional numbers, classics in turbomachinery: the specific speed, N_s, and the specific diameter, D_s (Mårtensson, Andersson, Trollheden, & Brodin, 2007).

The red circle in the figure below marks the optimal choice of specific speed and specific diameter. Commonly American literature does not use SI-units and comes out differently in terms of size. SI-units will be used in this case, and conversions made when using diagrams or numbers quoted from American literature.

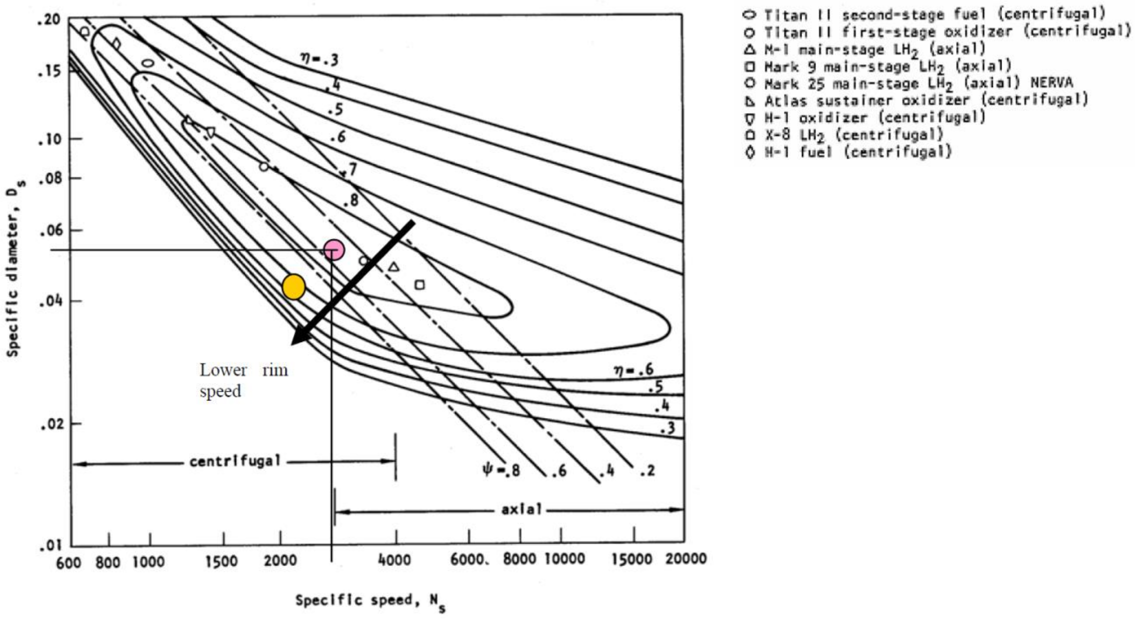


Figure 11 - N_s-D_s diagram with from NASA SP-8109

The selected point at maximum efficiency gives (converted to SI units):

$$D_s = 0.055 \cdot 50.5 \; = \; 2.77$$

$$N_s = 3000 \cdot 3.66 \cdot 10^{-4} = 1.1$$

The design point for a single-stage pump is the diameter and shaft speed for the machines. These values have been given as the default values of the design specific diameter and the design specific speed.

The flow coefficient and the loading coefficient can be related to the specific diameter and the specific speed:

$$D_s \propto \frac{\psi^{\frac{1}{4}}}{\phi^{\frac{1}{2}}}$$

$$N_s \propto \frac{\phi^{\frac{1}{2}}}{\psi^{\frac{3}{4}}}$$

And, turning the equations for the design coefficients, the following is obtained:

$$\psi_d \propto {N_{s,d}}^{-2} \cdot {D_{s,d}}^{-2}$$

$$\phi_d \propto {N_{s,d}}^{-1} \cdot {D_{s,d}}^{-3}$$

Therefore, ψ_d and ϕ_d can be calculated from the design specific diameter and the design specific speed given.

With coefficients ψ and ϕ a performance equation can be obtained:

$$\psi = 1 - \frac{1-\psi_d}{\phi_d}\phi$$

And for the efficiency, the performance equation is:

$$\eta = \eta_d$$

Following the development done in *Appendix D*, but in this case for perfect liquids (**the liquids with which the library works are perfect liquids: ideal liquids and calorically perfect**), the energy equation of a pump is:

$$W = \dot{m}\left(c(T_{in} - T_{out}) + \frac{p_{t,in} - p_{t,out}}{\rho}\right)$$

And with the definition of the specific work:

$$\tau = c\Delta T + \frac{\Delta p_t}{\rho}$$

The efficiency is defined as the ratio between the power that would be required in an ideal process (constant entropy) and the power required in a real situation:

$$\eta = \frac{(\tau)_{ideal}}{(\tau)_{real}}$$

In an ideal situation, the entropy is constant and, consequently for perfect liquids, $\Delta T = 0$. Hence:

$$(\tau)_{ideal} = \frac{\Delta p_t}{\rho}$$

And the efficiency results:

$$\eta = \frac{p_{t,out} - p_{t,in}}{\tau \cdot \rho}$$

Finally, the energy equation can be rewritten by using the efficiency, yielding:

$$c\Delta T = \tau - \frac{\Delta p_t}{\rho} = \frac{\Delta p_t/\rho}{\eta} - \frac{\Delta p_t}{\rho} = \frac{\Delta p_t}{\rho}\left(\frac{1}{\eta} - 1\right)$$

Besides, the pum head is calculated as:

$$H = \frac{p_{t,out} - p_{t,in}}{g_0 \cdot \rho}$$

The component also sets the power of the inlet mechanical port to W:

$$W = W_{in}$$

On-off design mode

If Type=Design, the component will set the pressure increase as data, where:

$$dp = p_{t,out} - p_{t,in}$$

Moreover, in design mode, there is an extra equation which matches the flow coefficient to the design flow coefficient given as data in order to fix the design condition:

$$\phi = \phi_d$$

The component will calculate the inlet area (A_{in}), the average radius (r_m) and the power. The rotational speed will be set by the turbine connected to the mechanical port.

Otherwise, if Type=Off_Design, the inlet area (A_{in}) and the average radius (r_m) will be set as data and the component will calculate the rotational speed and the pressure increase. In this case, the power will be set by the turbine connected to the mechanical port.

The mass flow will be set by another component in both cases.

To have a general overview, see *Appendix G*.

Error Messages

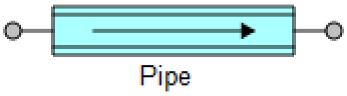
Only liquids can go through this component. If the working fluid is not in a liquid state, a FATAL ASSERT will appear.

4.7 Component Pipe

4.7.1 Description

This component type represents a liquid pipe with pressure drop.

It is inherited from abstract component “FluidInFluidOut”. It is defined in the LPRES_CompBasic.el file.



4.7.2 Data

Name	Type	Default	Description	Units	Condition
L	REAL	1	Length (L)	m	
D	REAL	0.1	Diameter (D)	m	
K	REAL	5	Additional pressure losses (K)	$-$	
rug	REAL	1.5e-6	Absolute rugosity (ε)	m	

4.7.3 Variables

Name	Type	Inital	Description	Units
v	REAL		Liquid speed (v)	$\frac{m}{s}$
f	REAL		Darcy friction factor (f)	$-$
Re	REAL		Reynolds number (Re)	$-$

4.7.4 Formulation

The model used to simulate the pressure drop of the pipe is the known Darcy-Weisbach equation:

$$p_{t,out} = p_{t,in} - \frac{1}{2}\rho v^2 \left(f\frac{L}{D} + K\right)$$

where:

$$v = \frac{\dot{m}}{\rho \cdot \pi \frac{D^2}{4}}$$

This model calculates the pressure drop due to friction and it also has a factor, K, that includes additional pressure losses, such as elbows.

This component uses the “hdc_fric” function to calculate the friction factor:

$$f = hdc\_fric(D, \varepsilon, Re)$$

To calculate the Reynolds number:

$$Re = \frac{\rho v D}{\mu}$$

Besides, it is considered that friction does not raise the liquid temperature. Hence:

$$T_{in} = T_{out}$$

Error Messages

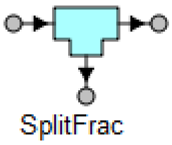
Only liquids can go through this component. If the working fluid is not in a liquid state, a FATAL ASSERT will appear.

4.8 Component SplitFrac

4.8.1 Description

This component type represents a low splitter with pressure drop.

It is defined in the LPRES_CompBasic.el file.



4.8.2 Ports

Name	Type	Direction	Description
f_in	Fluid	IN	Inlet fluid port
f_out	Fluid	OUT	Outlet fluid port
f_b	Fluid	OUT	Branch fluid port

4.8.3 Data

Name	Type	Default	Description	Units	Condition
TPL	REAL	0.9	Total pressure loss (TPL)	–	

4.8.4 Formulation

This component represents a flow splitter. Either for liquids or gases,

$$p_{t,out} = TPL \cdot p_{t,in}$$

$$T_{t,out} = T_{t,in}$$

and

$$p_{t,b} = TPL \cdot p_{t,in}$$

$$T_{t,b} = T_{t,in}$$

Note that for liquids $T = T_t$. TPL is a factor included in order to take into account the losses generated by the split. If $TPL = 1$ for liquids, it is assumed that the Bernoulli equation is fulfilled:

$$p + \frac{1}{2}\rho v^2 = cte$$

The Bernoulli equation comes from the deduction made in the *Appendix E*, but in this case:

$$\widetilde{w} = \frac{p}{\rho}$$

If $TPL = 1$ for gases, $p_t = cte$ and $T_t = cte$. These results are evidenced in *Appendix E*.

From the continuity equation, with the same assumptions that have been used before (see *Appendix C*), it is possible to conclude that the inlet mass flow must be the same as the outlet mass flow. Therefore:

$$\dot{m}_{in} = \dot{m}_{out} + \dot{m}_b$$

As this component has not been defined as a "FluidInFluidOut" component, it is necessary to establish that:

$$Y_{i,in} = Y_{i,out}$$

$$Y_{i,in} = Y_{i,b}$$

because it is assumed that there is no phase change in it. To establish it, the equation included is:

$$fluid_{in} = fluid_{out}$$

$$fluid_{in} = fluid_b$$

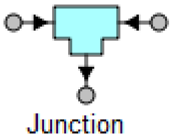
By the way, components defined as "FluidInFluidOut" transmit the auxiliary variables of the "fluid" array between their ports.

4.9 Component Junction

4.9.1 Description

This component type represents a junction with pressure drop.

It is defined in the LPRES_CompBasic.el file.



4.9.2 Ports

Name	Type	Direction	Description
f_in1	Fluid	IN	Inlet fluid port
f_in2	Fluid	IN	Inlet fluid port
f_out	Fluid	OUT	Outlet fluid port

4.9.3 Data

Name	Type	Default	Description	Units	Condition
TPL	REAL	0.9	Total pressure loss (TPL)	–	

4.9.4 Formulation

This component represents a flow splitter. Both for liquids or gases,

$$p_{t,out} = TPL \cdot p_{t,in1}$$

and

$$p_{t,out} = TPL \cdot p_{t,in2}$$

TPL is a factor included in order to take into account the losses generated by the split. If $TPL = 1$ for liquids, it is assumed that the Bernoulli equation is fulfilled:

$$p + \frac{1}{2}\rho v^2 = cte$$

The Bernoulli equation comes from the deduction made in the *Appendix E*, but in this case:

$$\widetilde{w} = \frac{p}{\rho}$$

If $TPL = 1$ for gases, $p_t = cte$. These results are evidenced in *Appendix E*.

To calculate the temperature of the fluid, the adiabatic mixing equation is presented. It states:

$$\dot{m}_{in1} c_{P,in1} \left(T_{t,out} - T_{t,in1}\right) + \dot{m}_{in2} c_{P,in2} \left(T_{t,out} - T_{t,in2}\right) = 0$$

Note that for liquids $T = T_t$.

From the continuity equation, with the same assumptions that have been used before (see *Appendix C*), it is possible to conclude that the inlet mass flow must be the same as the outlet mass flow. Therefore:

$$\dot{m}_{in1} + \dot{m}_{in2} = \dot{m}_{out}$$

As this component has not been defined as a “FluidInFluidOut” component, it is necessary to establish that:

$$Y_{i,out} = Y_{i,in1}$$

$$Y_{i,out} = Y_{i,in2}$$

because it is assumed that there is no phase change in it. To establish it, the equation included is:

$$fluid_{out} = fluid_{in1}$$

$$fluid_{out} = fluid_{in2}$$

By the way, components defined as “FluidInFluidOut” transmit the auxiliary variables of the array “fluid” between their ports.

4.10 Component Regulator

4.10.1 Description

This component type represents a pressure regulator.

It is inherited from abstract component “FluidInFluidOut”. It is defined in the LPRES_CompBasic.el file.

4.10.2 Construction Parameters

Name	Type	Default	Units
Type	ENUM Type_Regulator	Design	–

4.10.3 Data

Name	Type	Default	Description	Units	Condition

Name	Type	Default	Description	Units	Condition
dp_min	REAL	1500	Minumum total pressure drop $(\Delta p_{t,min})$	Pa	Type=Known_pt_out
pt_out	REAL	1200000	Outlet total pressure (p_t)	Pa	Type=Known_pt_out
dp	REAL	100000	Imposed toatl pressure drop (Δp_t)	Pa	Type=Known_dp
dpr	REAL	0.1	Imposed total pressure drop ratio $(TPDR)$	$-$	Type=Known_dpr

4.10.4 Variables

Name	Type	Inital	Description	Units
dp_d	REAL		Design total pressure drop (Δp_t)	Pa

4.10.5 Formulation

This component imposes a pressure drop:

$$p_{t,out} = p_{t,in} - \Delta p_t$$

It can be done in several ways:

Mode

If Type=Known_pt_out, the component imposes the outlet total pressure with data p_t, but it is required a minimum difference between $p_{t,in}$ and p_t for the regulator to work. The minimum difference, $\Delta p_{t,min}$, is set as data.

$$p_{t,out} = \min(p_t \,, p_{t,in} - \Delta p_{t,min})$$

Hence, $p_{t,out}$ is obtained and Δp_t is calculated.

If Type=Known_dp, Δp_t is imposed as data.

If Type=Known_dpr, $TPDR$ is imposed as data, it is defined as:

$$TPDR = \frac{p_{t,out}}{p_{t,in}}$$

Hence, $p_{t,out}$ is obtained and Δp_t is calculated.

Finally, if Type=Design, Δp_t is calculated by knowing $p_{t,out}$ and $p_{t,in}$ from the components connected to the regulator.

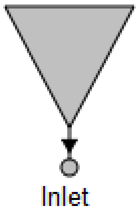
To have a general overview, see *Appendix G*.

4.11 Component Inlet

4.11.1 Description

This component type specifies the conditions of fluid inlet in the system.

It is defined in the LPRES_CompBasic.el file.



4.11.2 Construction Parameters

Name	Type	Default	Units
Type	ENUM Type_Inlet	All	–

4.11.3 Ports

Name	Type	Direction	Description
f	Fluid	OUT	Oulet fluid port

4.11.4 Data

Name	Type	Default	Description	Units	Condition
Tt	REAL	288.15	Total temperature (T_t)	K	
pt	REAL	101325	Total pressure (p_t)	Pa	
fluid	ENUM LiquidsGases	LOX	Working fluid name	$-$	
W	REAL	1	Mass flow ($\dot{m}$)	$\frac{kg}{s}$	Type=All

4.11.5 Formulation

This component sets the port variables with the data of the component. It is used to simplify the partitions when the values of the variables are known. It can also serve to simulate tanks.

Mode

If Type=All, the component will set the mass flow. Otherwise, if Type=Unknown_W, the component will not set the mass flow, it will be calculated by another component.

To have a general overview, see *Appendix G*.

Initialisation

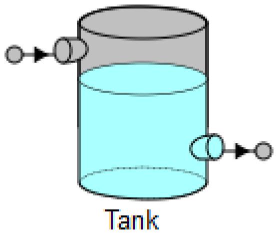
This component initialises the "fluid" port variable with the "fluid" data by using the "Init_fluid" function. It performs it with the maximum priority, because it is the first thing that the program should do to start the calculation. This is because the model of some components changes depending on the state of the fluid, which is set by the "fluid" port variable.

4.12 Component Tank

4.12.1 Description

This component type represents a liquid tank.

It is defined in the LPRES_CompBasic.el file.



4.12.2 Construction Parameters

Name	Type	Default	Units
Type	ENUM OnOffDesign	Design	–

4.12.3 Ports

Name	Type	Direction	Description
g	Fluid	IN	Inlet fluid port
l	Fluid	OUT	Outlet fluid port

4.12.4 Data

Name	Type	Default	Description	Units	Condition
fluid_l	ENUM Liquids	LOX	Working liquid name	–	
T_d	REAL	288.15	Tank temperature (T_d)	K	
A_g	REAL	0.001	Gas input area (A_g)	m^2	Type=Off_design
p_d	REAL	1000000	Design tank pressure (p_d)	Pa	Type=Design

4.12.5 Variables

Name	Type	Inital	Description	Units
p_g	REAL		Pressurisation gas pressure (p_g)	Pa

Name	Type	Inital	Description	Units
rho_g	REAL		Pressurisation gas density (ρ_g)	$\frac{kg}{m^3}$
M_g	ALG REAL	0.1	Pressurisation gas Mach number (M_g)	$-$
A_g_d	REAL		Design pressurisation gas input area (A_g)	m^2

4.12.6 Formulation

This component represents a pressurised liquid tank. The “fluid” data establishes the “fluid” port variable by using the “Init_fluid” function. The same happens with the temperature:

$$T_d = T_l$$

To know the pressure, the Bernoulli equation is used, hence:

$$p_{t,l} = p_g$$

On-off design mode

If Type=Design, the component will set the tank pressure, being:

$$p_g = p_d$$

The component will calculate the gas inlet area (A_g).

Otherwise, if Type=Off_Design, the inlet area (A_g) will be set as data and the component will calculate the tank pressure (p_g).

In both cases the component will calculate the gas mass flow. The liquid mass flow will be set by another component.

To have a general overview, see *Appendix G*.

The mass flow equation is:

$$\dot{m}_g = \frac{A_g p_{t,g}}{\sqrt{R_g T_{t,g}}} \cdot \sqrt{\gamma} M_g \left(\frac{1}{1 + \frac{\gamma - 1}{2} M_g^2} \right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

Besides, the inlet density can be calculated as:

$$\left(\frac{\rho_{t,g}}{\rho_g}\right)^{\gamma-1} = 1 + \frac{\gamma-1}{2} M_g^2 \quad ; \quad \rho_{t,g} = \frac{p_{t,g}}{R_g T_{t,g}}$$

and the inlet pressure as:

$$\left(\frac{p_{t,g}}{p_g}\right)^{\frac{\gamma-1}{\gamma}} = 1 + \frac{\gamma-1}{2} M_g^2$$

The last equation included in the component is

$$\frac{\dot{m}_l}{\rho_l} = \frac{\dot{m}_g}{\rho_g}$$

which comes from the continuity equation. The continuity equation sets:

$$\frac{dM_g}{dt} = \dot{m}_g$$

where M_g is the gas mass inside the tank, which can be written as:

$$M_g = V_g \cdot \rho_g$$

Hence, the resulting equation is:

$$\rho_g \frac{dV_g}{dt} + V_g \frac{d\rho_g}{dt} = \dot{m}_g$$

V_g is the gas volume inside the tank, so:

$$V_g = V_T - V_l$$

where V_T is the total vulume of the tank and V_l is the liquid volume inside it. Therefore:

$$\frac{dV_g}{dt} = -\frac{dV_l}{dt} = -\frac{d}{dt}\left\{\frac{M_l}{\rho_l}\right\} = \frac{\dot{m}_l}{\rho_l}$$

With this result and dividing by ρ_g, the equation is:

$$\frac{V_g}{\rho_g}\frac{d\rho_g}{dt} + \frac{\dot{m}_l}{\rho_l} = \frac{\dot{m}_g}{\rho_g}$$

The term $\frac{V_g}{\rho_g}\frac{d\rho_g}{dt}$ can be neglected when the engine has reached the steady state:

$$\frac{V_g}{t_c} \sim \frac{\dot{m}_l}{\rho_l}\ ;\ \dot{m}_l \sim \frac{M_l}{t_b} \quad \Rightarrow \quad \frac{t_c}{t_b} \sim \frac{V_g}{M_l/\rho_l}$$

Error Messages

Only gases can enter this component. If the gas port working fluid is not in a gas state, a FATAL ASSERT will appear.

If the gas inlet is choked ($M_g > 1$), a KILLPOINT ASSERT will appear.

Initialisation

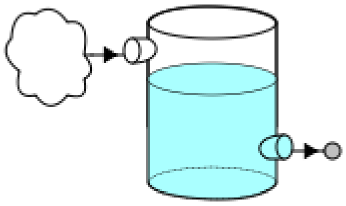
This component initialises the “fluid” port variable with the “fluid” data by using the “Init_fluid” function. It performs it with the maximum priority, because it is the first thing that the program should do to start the calculation. This is because the model of some components changes depending on the state of the fluid, which is set by the “fluid” port variable.

4.13 Component TankOpen

4.13.1 Description

This component type represents a liquid tank that is pressurised by the atmosphere.

It is defined in the LPRES_CompBasic.el file.



TankOpen

4.13.2 Ports

Name	Type	Direction	Description
1	Fluid	OUT	Outlet fluid port

4.13.3 Data

Name	Type	Default	Description	Units	Condition
fluid	ENUM Liquids	LOX	Working liquid name	–	
T_d	REAL	288.15	Tank temperature (T_d)	K	

4.13.4 Variables

Name	Type	Inital	Description	Units
p_d	REAL		Tank pressure (p_d)	Pa

4.13.5 Formulation

This component represents a liquid tank that is pressurised by the atmosphere. The “fluid” data establishes the “fluid” port variable by using the “Init_fluid” function. The same happens with the temperature:

$$T_d = T_l$$

To know the pressure, the Bernoulli equation is used, hence:

$$p_{t,l} = p_d$$

The difference between this tank and the previous one is the way that they establish the tank pressure, p_d. In this case, as the tank is open to the atmosphere, the atmosphere sets it:

$$p_d = ISA\_Pressure(z)$$

The “Ambient” component must not be connected to this component to set the ambient pressure.

Initialisation

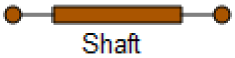
This component initialises the “fluid” port variable with the “fluid” data by using the “Init_fluid” function. It performs it with the maximum priority, because it is the first thing that the program should do to start the calculation. This is because the model of some components changes depending on the state of the fluid, which is set by the “fluid” port variable.

4.14 Component Shaft

4.14.1 Description

This component type represents a mechanical shaft without acceleration.

It is defined in the LPRES_CompBasic.el file.



4.14.2 Ports

Name	Type	Direction	Description
m_1	Mechanical	OUT	Outlet mechanical port
m_2	Mechanical	OUT	Outlet mechanical port

4.14.3 Data

Name	Type	Default	Description	Units	Condition
eta	REAL	1	Efficiency (η)	$-$	

4.14.4 Variables

Name	Type	Inital	Description	Units
rpm	REAL		Rotational speed (rpm)	rpm

4.14.5 Formulation

This component represents a mechanical shaft which is used to transmit power between components with mechanical elements. As it is a steady library, the component does not have acceleration.

Both component ports are outlet ports, and mechanical ports in other components of the library are inlet ports. It is defined in this way to avoid problems with mechanical power sign. Thus, a component which produces power or another who receives it, can be connected, no matter to what port, to the shaft without any kind of problem. Even two components can be connected to the same ports.

In the component it is established that the number of revolutions of both ports is the same, whose value is $\omega = rpm \cdot 2\pi/60$, being rpm data of the component.

Finally, as the acceleration is null, the equation which governs the behaviour of the component is the equilibrium of the power of the ports, which can be positive or negative. This equilibrium is modelled using an efficiency to represent the losses.

If W_1 provides power ($W_1 > 0$):

$$0 = \eta \cdot W_1 + W_2$$

Otherwise, if the power is provided by W_2:

$$0 = W_1 + \eta \cdot W_2$$

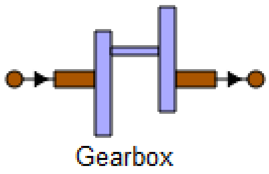
This condition is due to the fact that the power received must be lower than the power provided because of the losses.

4.15 Component Gearbox

4.15.1 Description

This component type represents a mechanical shaft with a gearbox and without acceleration.

It is defined in the LPRES_CompBasic.el file.



4.15.2 Ports

Name	Type	Direction	Description

Name	Type	Direction	Description
m_in	Mechanical	OUT	Outlet mechanical port
m_out	Mechanical	OUT	Outlet mechanical port

4.15.3 Data

Name	Type	Default	Description	Units	Condition
GR	REAL	1	Gear ratio (GR)	−	
eta	REAL	1	Efficiency (η)	−	

4.15.4 Formulation

This component represents a mechanical shaft with a gearbox which is used to transmit power between components with mechanical elements. As it is a steady library, the component does not have acceleration.

Both component ports are outlet ports, and mechanical ports in other components of the library are inlet ports. It is defined in this way to avoid problems with mechanical power sign. Thus, a component which produces power or another who receives it, can be connected, no matter to what port, to the shaft without any kind of problem. Even two components can be conected to the same ports.

In the component it is established that the number of revolutions of both ports is not the same, there is a gearbox which changes the rotational speed between the sides of the component by using a gear ratio:

$$\omega_{out} = GR \cdot \omega_{in}$$

Finally, as the acceleration is null, the equation which governs the behaviour of the component is the equilibrium of the power of the ports, which can be positive or negative. This equilibrium is modelled using an efficiency to represent the losses.

If W_{in} provides power ($W_{in} > 0$):

$$0 = \eta \cdot W_{in} + W_{out}$$

Otherwise, if the power is provided by W_{out}:

$$0 = W_{in} + \eta \cdot W_{out}$$

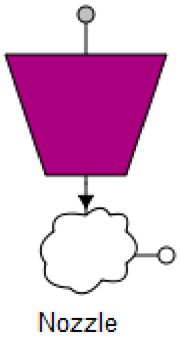
This condition is due to the fact that the power received must be lower than the power provided because of the losses.

4.16 Component Nozzle

4.16.1 Description

This component type represents a convergent nozzle with the ambient already connected.

It is defined in the LPRES_CompBasic.el file.



4.16.2 Construction Parameters

Name	Type	Default	Units
Type	ENUM OnOffDesign	Design	–

4.16.3 Ports

Name	Type	Direction	Description
g	Fluid	IN	Inlet fluid port
i	Info (n = 1)	OUT	Outlet information port

4.16.4 Data

Name	Type	Default	Description	Units	Condition
A	REAL	0.02	Discharge area (A)	m^2	Type=Off_design

4.16.5 Variables

Name	Type	Inital	Description	Units
p_amb	REAL		Ambient pressure (p_{amb})	Pa
A_d	REAL		Design discharge area (A)	m^2
A_sl	REAL		Sonic lock area (A_{sl})	m^2
PR	REAL		Pressure ratio (PR)	$-$
PR_sl	REAL		Sonic lock pressure ratio (PR_{sl})	$-$
p_out	REAL		Outlet pressure (p_{out})	Pa
p_out_ch	REAL		Choked outlet pressure ($p_{out,ch}$)	Pa
M_out	REAL		Outlet Mach number (M_{out})	$-$
T_out	REAL		Outlet temperature (T_{out})	K
v_out	REAL		Outlet speed (v_{out})	$\frac{m}{s}$
Thrust	REAL		Thrust (E)	N

4.16.6 Formulation

This component represents a convergent nozzle. The ambient is already connected to it. Thereby, the component is able to know if the outlet of the nozzle is choked or not because it knows the ambient pressure using the "ISA_Pressure" function:

$$p_{amb} = ISA\_Pressure(z)$$

The flux inside the nozzle fulfils (see *Appendix E*):

$$p_t = cte$$

$$T_t = cte$$

From that, it is possible to calculate the Mach number in any section of the nozzle from the static pressure or vice versa:

$$\left(\frac{p}{p_t}\right)^{\frac{\gamma-1}{\gamma}} = \frac{1}{1+\frac{\gamma-1}{2}M^2}$$

The component calculates the sonic lock pressure ratio, PR_{sl}, which is defined as:

$$PR_{sl} = \frac{p_{t,in}}{p_{out,ch}}$$

and it is the relation between p_t and p when $M = 1$, the maximum Mach number that can be reached in a convergent nozzle if the inlet flux is subsonic, as is the case. Therefore:

$$PR_{sl} = \left(\frac{\gamma+1}{2}\right)^{\frac{\gamma}{\gamma-1}}$$

Finally, the outlet pressure is:

$$p_{out} = max(p_{out,ch}\,, p_{amb})$$

Furthermore, the pressure ratio is defined:

$$PR = \frac{p_{t,in}}{p_{amb}}$$

The outlet Mach number is calculated with a conditional sentence which depend on whether PR is bigger or smaller than PR_{sl}. If the outlet is choked ($PR > PR_{sl}$):

$$M_{out} = 1$$

otherwise ($PR < PR_{sl}$):

$$M_{out} = \sqrt{\frac{2}{\gamma-1}\left(PR^{\frac{\gamma-1}{\gamma}} - 1\right)}$$

Last equation comes from the relation between p_t and p that has been previously developed.

Mass flow, $\dot{m}$, is constant throughout the component, so:

$$\dot{m} = \rho v A = \rho M a A = A\rho_t a_t \cdot M\left(\frac{1}{1+\frac{\gamma-1}{2}M^2}\right)^{\frac{\gamma+1}{2(\gamma-1)}} =$$

$$= A\frac{p_t}{\sqrt{R_g T_t}} \cdot \sqrt{\gamma} M \left(\frac{1}{1+\frac{\gamma-1}{2}M^2}\right)^{\frac{\gamma+1}{2(\gamma-1)}} = A_{sl}\frac{p_t}{\sqrt{R_g T_t}} \cdot \sqrt{\gamma}\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} =$$

$$= A_{sl}\frac{p_t}{\sqrt{R_g T_t}} \cdot \Gamma(\gamma) = \mathrm{cte}$$

where A_{sl} is the outlet area that the nozzle would have if it were choked with the same inlet conditions.

It is necessary to establish an equation that determines a relationship between A_{sl}, A and M. This equation comes from the above equation:

$$A M_{out} \rho_t a_t \left(\frac{1}{1+\frac{\gamma-1}{2}{M_{out}}^2}\right)^{\frac{\gamma+1}{2(\gamma-1)}} = A_{sl}\rho_t a_t \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

Hence:

$$\frac{A_{sl}}{A} = \frac{M_{out}}{\left(\frac{2+(\gamma-1){M_{out}}^2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}}}$$

Note that, according to this equation, if $M_{out} = 1$, $A_{sl} = A$.

Finally, this component evaluates the thrust, which comes from:

$$E = \dot{m} \cdot v_{out} + A \cdot (p_{out} - p_{amb})$$

To calculate v_{out}:

$$v_{out} = M_{out}\sqrt{\gamma R_g T_{out}}$$

And to calculate T_{out}:

$$\frac{T_{t,out}}{T_{out}} = 1 + \frac{\gamma-1}{2}{M_{out}}^2$$

Once the thrust is calculated, it is sent through the “Info” port to the “ControlPanel” component (if it is connected).

On-off design mode

If Type=Off_Design, the outlet area (A) will be set as data and the component will calculate the mass flow. Otherwise, if Type=Design, the component will calculate the outlet area (A) and the mass flow will be set by another component.

To have a general overview, see *Appendix G*.

Error Messages

Only gases can go through this component. If the working fluid is not in a gas state, a FATAL ASSERT will appear.

Initialisation

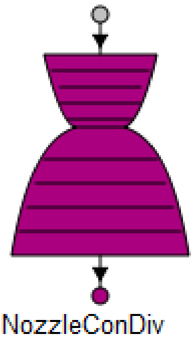
The initial value given for $p_{t,in}$ is $10000000\ Pa$. This value is given to help reaching the convergence. It is simply an initial value to start the iterations; it is not the final value.

4.17 Component NozzleConDiv

4.17.1 Description

This component type represents a choked convergent-divergent nozzle.

It is defined in the LPRES_CompBasic.el file.



4.17.2 Construction Parameters

Name	Type	Default	Units
Type	ENUM OnOffDesign	Design	–

4.17.3 Ports

Name	Type	Direction	Description
g_in	Fluid	IN	Inlet fluid port
g_out	GasNozzle	OUT	Outlet gas port through a nozzle

4.17.4 Data

Name	Type	Default	Description	Units	Condition
AR	REAL	10	Area ratio (ε)	$-$	
A_th	REAL	0.05	Throat area (A_{th})	m^2	Type=Off_design
W	REAL	100	Mass flow ($\dot{m}$)	$\frac{kg}{s}$	Type=Design

4.17.5 Variables

Name	Type	Inital	Description	Units
A_th_d	REAL		Design throat area (A_{th})	m^2
A_out	REAL		Output area (A_{out})	m^2
p_out_ch	ALG REAL	100	Choked outlet pressure ($p_{out,ch}$)	Pa

4.17.6 Formulation

This component represents a choked convergent-divergent nozzle. It uses the "GasNozzle" port, which has been created to be able to add nozzle extensions and to calculate the thrust of the rocket, because the components that do it will need some extra information apart from that sent by a common "Fluid" port. They will need the outlet area of the nozzle.

The flow inside the nozzle fulfils (see *Appendix E*):

$$p_{t,in} = p_{t,out}$$

$$T_{t,in} = T_{t,out}$$

Since it is choked:

$$p_{out} = p_{out,ch}$$

and the choked outlet pressure comes from the following equation:

$$\frac{A_{out}}{A_{th}} = \frac{\Gamma(\gamma)}{\left(\frac{p_{out,ch}}{p_{t,in}}\right)^{\frac{1}{\gamma}} \sqrt{\frac{2 \cdot \gamma}{\gamma - 1}\left(1 - \left(\frac{p_{out,ch}}{p_{t,in}}\right)^{\frac{\gamma-1}{\gamma}}\right)}}$$

Which comes from the same equation used in the “Nozzle” component that sets a constant mass flow, $\dot{m}$, throughout the component, knowing that the section in which $M = 1$ is the throat:

$$\dot{m} = A \frac{p_t}{\sqrt{R_g T_t}} \cdot \sqrt{\gamma} M \left(\frac{1}{1 + \frac{\gamma - 1}{2} M^2}\right)^{\frac{\gamma+1}{2(\gamma-1)}} =$$

$$= A_{out} \frac{p_t}{\sqrt{R_g T_t}} \cdot \sqrt{\gamma} M_{out} \left(\frac{1}{1 + \frac{\gamma - 1}{2} {M_{out}}^2}\right)^{\frac{\gamma+1}{2(\gamma-1)}} = A_{th} \frac{p_t}{\sqrt{R_g T_t}} \cdot \Gamma(\gamma)$$

and

$$M_{out} = \sqrt{\frac{2}{\gamma - 1}\left(\left(\frac{p_{t,in}}{p_{out,ch}}\right)^{\frac{\gamma-1}{\gamma}} - 1\right)}$$

but one thing must be kept in mind, the equation that sets a constant mass flow has two available solutions to M_{out}: a subsonic solution and a supersonic solution. That is the reason of why an initial value is given for variable “p_out_ch”. The given value is 100 because this means that the iterations will start in low pressure values which would mean that the divergence of the nozzle is working with subsonic flow.

Moreover, this equation is also included in the component to relate the throat area and the mass flow:

$$\dot{m} = A_{th} \frac{p_t}{\sqrt{R_g T_t}} \cdot \Gamma(\gamma)$$

The area ratio, ε, is defined as:

$$\varepsilon = \frac{A_{out}}{A_{th}}$$

Besides, this component sends A_{out} through the "GasNozzle" port.

As this component has not been defined as a "FluidInFluidOut" component, it is necessary to establish that:

$$Y_{i,out} = Y_{i,in}$$

because it is assumed that there is no phase change in it. To establish it, the included equation is:

$$fluid_{out} = fluid_{in}$$

By the way, components defined as "FluidInFluidOut" transmit the auxiliary variables of the "fluid" array between their ports. Besides (see *Appendix C*):

$$\dot{m}_{in} = \dot{m}_{out}$$

This component could not have been defined as a "FluidInFluidOut" component because there is no outlet "Fluid" port.

On-off design mode

If Type=Off_Design, the throat area (A_{th}) will be set as data and the component will calculate the mass flow. Otherwise, if Type=Design, the component will calculate the throat area (A_{th}) and the mass flow will be set as data.

To have a general overview, see *Appendix G*.

Error Messages

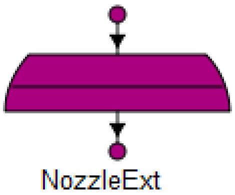
Only gases can go through this component. If the working fluid is not in a gas state, a FATAL ASSERT will appear.

4.18 Component NozzleExt

4.18.1 Description

This component type represents a nozzle extension.

It is defined in the LPRES_CompBasic.el file.



4.18.2 Ports

Name	Type	Direction	Description
g_in	GasNozzle	IN	Inlet gas port through a nozzle
g_out	GasNozzle	OUT	Outlet gas port through a nozzle

4.18.3 Data

Name	Type	Default	Description	Units	Condition
AR	REAL	2	Extension area ratio (ε)	$-$	

4.18.4 Variables

Name	Type	Inital	Description	Units
A_in	REAL		Input area (A_{in})	m^2
A_out	REAL		Output area (A_{out})	m^2

4.18.5 Formulation

This component represents a nozzle extension. It is thought to be useful in future developments of the library, for example, in order to be able to refrigerate only a zone of the nozzle, in which case it would be necessary to include a heat port in the component with their associated equations. Both, the inlet port and the outlet port are “GasNozzle” ports.

The flux inside the nozzle fulfils (see *Appendix E*):

$$p_{t,in} = p_{t,out}$$

$$T_{t,in} = T_{t,out}$$

Moreover, it is defined the area ratio, ε, as:

$$\varepsilon = \frac{A_{out}}{A_{in}}$$

Besides, this component sends A_{out} by the “GasNozzle” port. It does not matter that the convergent-divergent nozzle to which this component is connected has previously set an outlet area. The important outlet area in terms of thrust calculation is the real outlet area, which is the one set by the last component of the chain.

As this component has not been defined as a “FluidInFluidOut” component, it is necessary to establish that:

$$Y_{i,out} = Y_{i,in}$$

because it is assumed that there is no phase change in it. To establish it, the included equation is:

$$fluid_{out} = fluid_{in}$$

By the way, components defined as “FluidInFluidOut” transmit the auxiliary variables of the “fluid” array between their ports. Besides (see *Appendix C*):

$$\dot{m}_{in} = \dot{m}_{out}$$

This component could not have been defined as a “FluidInFluidOut” component because there is neither an outlet “Fluid” port nor an inlet “Fluid” port.

Error Messages

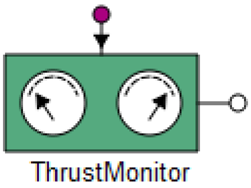
Only gases can go through this component. If the working fluid is not in a gas state, a FATAL ASSERT will appear.

4.19 Component ThrustMonitor

4.19.1 Description

This component type represents a monitor to measure the thrust.

It is defined in the LPRES_CompBasic.el file.



4.19.2 Ports

Name	Type	Direction	Description
g	GasNozzle	IN	Inlet gas port through a nozzle
i	Info (n = 1)	OUT	Outlet information port

4.19.3 Variables

Name	Type	Inital	Description	Units
Thrust	REAL		Thrust (E)	N
p_amb	REAL		Ambient pressure (p_{amb})	Pa
p_out	REAL		Outlet pressure (p_{out})	Pa
T_out	REAL		Outlet temperature (T_{out})	K
v_out	REAL		Outlet speed (v_{out})	$\frac{m}{s}$
A_out	REAL		Output area (A_{out})	m^2
M_out	ALG REAL	100	Gas Mach number (M_{out})	$-$

4.19.4 Formulation

This component is used to calculate the thrust.

The thrust comes from:

$$E = \dot{m} \cdot v_{out} + A_{out} \cdot (p_{out} - p_{amb})$$

The component takes A_{out} from the “GasNozzle” port.

This component uses the “ISA_Pressure” function to set the ambient pressure:

$$p_{amb} = ISA\_Pressure(z)$$

To calculate v_{out}:

$$v_{out} = M_{out}\sqrt{\gamma R_g T_{out}}$$

Obtaining M_{out} from the mass flow equation:

$$\dot{m} = A\frac{p_t}{\sqrt{R_g T_t}} \cdot \sqrt{\gamma} M \left(\frac{1}{1+\frac{\gamma-1}{2}M^2}\right)^{\frac{\gamma+1}{2(\gamma-1)}} =$$

$$= A_{out}\frac{p_t}{\sqrt{R_g T_t}} \cdot \sqrt{\gamma} M_{out} \left(\frac{1}{1+\frac{\gamma-1}{2}{M_{out}}^2}\right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

And T_{out} from:

$$\frac{T_{t,out}}{T_{out}} = 1 + \frac{\gamma-1}{2}{M_{out}}^2$$

To calculate p_{out}:

$$\left(\frac{p_{t,out}}{p_{out}}\right)^{\frac{\gamma-1}{\gamma}} = 1 + \frac{\gamma-1}{2}{M_{out}}^2$$

Once the thrust is calculated, it is sent through the “Info” port to the “ControlPanel” component (if it is connected).

Error Messages

Only gases can go through this component. If the working fluid is not in a gas state, a FATAL ASSERT will appear.

If the gas inlet is not choked ($M_{out} < 1$), a KILLPOINT ASSERT will appear.

This component can only be used if the nozzle does not have a region of detachment, so, according to the Schmucker criterion:

$$\frac{p_{out}}{p_{amb}} > (1.88M_{out} - 1)^{-0.64}$$

If the criterion is not fulfilled, a WARNING ASSERT will appear.

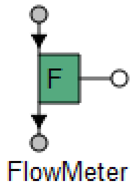
The library will only be able to detect whether the nozzle has a region of detachment or not if this component is connected. The same applies to whether is choked or not.

4.20 Component FlowMeter

4.20.1 Description

This component type represents a mass flow meter.

It is inherited from abstract component “FluidInFluidOut”. It is defined in the LPRES_CompBasic.el file.



4.20.2 Ports

Name	Type	Direction	Description
i	Info (n = 1)	OUT	Outlet information port

4.20.3 Formulation

This component is used to send the mass flow to a “ControlPanel” component. It is included in a flow line and it measures the mass flow that goes through it.

As it does not represent any physical component:

$$p_{t,in} = p_{t,out}$$

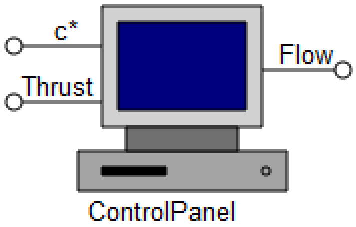
$$T_{t,in} = T_{t,out}$$

4.21 Component ControlPanel

4.21.1 Description

This component type performs the calculations with the measurements done by other components.

It is defined in the LPRES_CompBasic.el file.



4.21.2 Ports

Name	Type	Direction	Description
i_W	Info (n = 1)	IN	Inlet information port
i_Thrust	Info (n = 1)	IN	Inlet information port
i_Comb	Info (n = 1)	IN	Inlet information port

4.21.3 Variables

Name	Type	Inital	Description	Units
Thrust	REAL		Thrust (E)	N
W_tot	REAL		Total mass flow ($\dot{m}_T$)	$\frac{kg}{s}$
c_star	REAL		Characteristic exhaust velocity (c^*)	$\frac{m}{s}$
Isp	REAL		Specific impulse [m/s] (I_{sp})	$\frac{m}{s}$
Isp_0	REAL		Specific impulse [s] (I_{sp}^0)	s
C_E	REAL		Thrust coefficient (C_E)	$-$

4.21.4 Formulation

This component uses the information provided by other components to calculate the specific impulse and the thrust coefficient. To do this, the other components must be properly connected to it.

The “ThrustMonitor” component must be connected to the port with labelled “Thrust”.

The “FlowMeter” components must be connected to the port labelled “Flow”. Several “FlowMeter” components can be connected to this port. $\dot{m}_T$ will be the sum of the mass flow provided by all the “FlowMeter” components connected to this port.

Finally, component “CombCha” must be connected to the port labelled “c*”. Component “GasGen”can also be connected to this port.

The component performs:

$$I_{sp} = \frac{E}{\dot{m}_T}$$

$$I_{sp} = C_E \cdot c^*$$

$$I_{sp} = g_0 \cdot I_{sp}^0$$

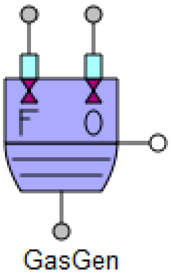
Note that these parameters are ideal because the divergence effects, the viscosity, etc, are not included in the model of the convergent-divergent nozzle.

4.22 Component GasGen

4.22.1 Description

This component type represents a gas generator.

It is defined in the LPRES_CompGasGen.el file.



4.22.2 Ports

Name	Type	Direction	Description
f_O	Fluid	IN	Inlet fluid port
f_F	Fluid	IN	Inlet fluid port
g	Fluid	OUT	Outlet fluid port
i	Info (n = 1)	OUT	Outlet information port

4.22.3 Data

Name	Type	Default	Description	Units	Condition
TPL_d	REAL	0.9	Design total pressure loss (TPL_d)	$-$	
eta_d	REAL	0.9	Design combustion efficiency (η_d)	$-$	
OF_st	REAL	8	Stoichiometric mixture ratio (OF_{st})	$-$	
Q_comb	REAL	2000000	Heat of combustion per oxidant mass flow unit (Q_{comb})	$\frac{J}{kg}$	
cp_P	REAL	4182	Specific heat at constant pressure of the products using a stoichiometric mixture $(c_{P,P})$	$\frac{J}{kg \cdot K}$	
M_P	REAL	32	Molar mass of the products using a stoichiometric mixture $(M_{m,P})$	$\frac{g}{mol}$	
OF	REAL	4	Mixture ratio (OF)	$-$	
W_F0	REAL	100	Initial fuel mass flow for iterative calculations	$\frac{kg}{s}$	

4.22.4 Variables

Name	Type	Initial	Description	Units
TPL	REAL		Total pressure loss (TPL)	$-$
T_in	REAL		Average inlet temperature (T_{in})	K
T_c	REAL		Combustion temperature (T_c)	K
eta	REAL		Combustion efficiency (η)	$-$
phi	REAL		Equivalence ratio (ϕ)	$-$
W_F_st	REAL		Fuel mass flow that provides a stoichiometric mixture when combined with the oxidant mass flow $(\dot{m}_{Fst})$	$\frac{kg}{s}$
W_F	ALG REAL	W_F0	Fuel mass flow $(\dot{m}_F)$	$\frac{kg}{s}$
W_O	REAL		Oxidant mass flow $(\dot{m}_O)$	$\frac{kg}{s}$
W_IF	REAL		Inert mass flow through the fuel port $(\dot{m}_{IF})$	$\frac{kg}{s}$
W_IO	REAL		Inert mass flow through the oxidant port $(\dot{m}_{IO})$	$\frac{kg}{s}$
fluid_F	HIDDEN REAL[ChemName]		Fuel fluid	$-$
fluid_O	HIDDEN REAL[ChemName]		Oxidant fluid	$-$
fluid_P	HIDDEN REAL[ChemName]		Combustion products fluid	$-$
cp_R	REAL		Reactant specific heat at constant pressure $(c_{P,R})$	$\frac{J}{kg \cdot K}$
Combustion	BOOLEAN		TRUE if there is combustion and FALSE if either oxidant or fuel is lacking	

Name	Type	Initial	Description	Units
Q_comb_effective	REAL		Effective heat of combustion per oxidant mass flow unit ($Q_{comb,ef}$)	$\frac{J}{kg}$
c_star	REAL		Characteristic exhaust velocity (c^*)	$\frac{m}{s}$

4.22.5 Formulation

In this component a gas generator is modelled. It has two inlets: one for the oxidant and another for the fuel. Combustion between the oxidant and the fuel is what generates the new gas.

It is not necessary to use injectors in the inlets of this component (in fact, it is no possible), they are already in it modelled as:

$$p_{t,out} = p_c = TPL \cdot min(p_{t,O} \,, p_{t,F})$$

Where p_c is the combustion pressure. TPL, the total pressure loss comes from:

$$TPL = TPL_d$$

Where TPL_d is the design total pressure loss which is established as data. The same happens with the combustion efficiency:

$$\eta = \eta_d$$

From the continuity equation, with the same assumptions that have been used before (see *Appendix C*), it is possible to conclude that the inlet mass flow must be the same as the outlet mass flow. Therefore:

$$\dot{m}_{inO} + \dot{m}_{inF} = \dot{m}_{out}$$

Besides:

$$\dot{m}_{inO} = \dot{m}_O + \dot{m}_{IO}$$

$$\dot{m}_{inF} = \dot{m}_F + \dot{m}_{IF}$$

which means that not all the mass flow that comes from the port ($\dot{m}_{inO}$ and $\dot{m}_{inF}$) will react. This can be possible if another “GasGen” component has been placed before. In this case, the outlet of the first “GasGen” component will be composed of combustion gases, which are inert, and by excess fuel or oxidant, which can still react. The outlet of the first “GasGen” component can be the fuel or oxidant inlet port (depending on whether there is fuel or oxidant excess, respectively) of a second “GasGen” component. The leftover fuel or oxidant from the first “GasGen” component will react in the second one with the oxidant or fuel that enters through the other port.

The nomenclature used is $\dot{m}_O$ and $\dot{m}_F$ to represent the part of the inlet mass flow through the ports ($\dot{m}_{inO}$ and $\dot{m}_{inF}$) that can react. $\dot{m}_{IO}$ and $\dot{m}_{IF}$ represent the part of the inlet mass flow through the ports ($\dot{m}_{inO}$ and $\dot{m}_{inF}$) that cannot react (inert). To calculate them:

$$\dot{m}_O = \dot{m}_{inO} \cdot (1 - Y_{inO,Comb\_prod})$$

$$\dot{m}_F = \dot{m}_{inF} \cdot (1 - Y_{inF,Comb\_prod})$$

Where $Y_{inO,i}$ and $Y_{inF,i}$ are the mass fractions of the fluid that enters through the ports.

Similary, $Y_{O,i}$ and $Y_{F,i}$ are the mass fractions of the part of the inlet mass flow through the ports that can react. To calculate them:

$$Y_{O,i} = \frac{Y_{inO,i}}{(1 - Y_{inO,Comb\_prod})}$$

$$Y_{F,i} = \frac{Y_{inF,i}}{(1 - Y_{inF,Comb\_prod})}$$

$$Y_{O,Comb\_prod} = 0 \quad ; \quad Y_{F,Comb\_prod} = 0$$

$Y_{O,i}$ and $Y_{F,i}$ are contained in variables “fluid_O” and “fluid_F” respectively. The last two components (the ones with the indexes “Comb_prod_cp” and “Comb_prod_M”) of arrays “fluid_O” and “fluid_F” are set to 0.

It may well be that, if fuel enters through the fuel port of the second “GasGen”, by mistake, instead of excess oxidant from the first “GasGen” (whose outlet is the inlet of the oxidant port of the second “GasGen”), it is fuel that goes out. Thus, the combustion will not occur in the second “GasGen”. The same will happen if oxidant enters through the oxidant port and in the first “GasGen” there is oxidant excess instead of fuel excess. To simulate it, variables “Combustion” and “Q_comb_effective” ($Q_{comb,ef}$) have been created. If this mistake occurs, “Combustion” will be equal to FALSE and $Q_{comb,ef} = 0$. Otherwise, “Combustion” will be equal to TRUE and $Q_{comb,ef} = Q_{comb}$. To know if “Combustion” is true or false, the component performs:

$$\sum_{i}^{LiquidsGases} Y_{O,i} \cdot Y_{F,i}$$

If the result is equal to 0, “Combustion” will be TRUE because it means that the oxidant is different to the fuel, as it should be. Otherwise, If the result is not 0, “Combustion” will be FALSE because it means that the oxidant and the fuel are the same chemical, which is a mistake.

The mixture ratio, OF, is defined as:

$$OF = \frac{\dot{m}_O}{\dot{m}_F}$$

and it is data of the component.

The equivalence ratio, ϕ, is:

$$\phi = \frac{OF_{st}}{OF}$$

and the fuel mass flow that provides a stoichiometric mixture when combined with the oxidant mass flow is:

$$\dot{m}_{Fst} = \frac{\dot{m}_F}{\phi}$$

To calculate the combustion temperature, T_c, the combustion model states (see *Apendix F*):

$$\eta \cdot Q_{comb,ef} = \frac{1 + OF}{min(OF\ , OF_{st})} \cdot \left(c_{P,P'}\left(T_c - T_{ref}\right) - c_{P,R}\left(T_{in} - T_{ref}\right)\right)$$

where T_{in} is calculated as (see *Apendix F*):

$$\left(1 + \frac{\phi}{OF_{st}}\right) c_{P,R} T_{in} = c_{P,O} T_{t,O} + \frac{\phi}{OF_{st}} c_{P,F} T_{t,F}$$

where $c_{P,R}$ (see *Apendix F*):

$$c_{P,R} = \frac{\dot{m}_O c_{P,O} + \dot{m}_F c_{P,F}}{\dot{m}_O + \dot{m}_F}$$

Note that for liquids $T = T_t$. To calculate $c_{P,P'}$, $c_{P,F}$ and $c_{P,O}$, function “cp” is called, where the variables “fluid_P”, “fluid_F” and “fluid_O” the function inputs, respectively.

To calculate variable “fluid_P” (see *Apendix F*), if “Combustion” is equal to TRUE:

$$Y_{P,i} = \frac{\max(1 - \phi, 0) \cdot Y_{O,i} \dot{m}_O + \max(\phi - 1{,}0) \cdot Y_{F,i} \dot{m}_{Fst}}{\dot{m}_O + \dot{m}_F}$$

$$Y_{P,Comb\_prod} = \frac{(1 - \max(1 - \phi, 0)) \cdot (\dot{m}_O + \dot{m}_{Fst})}{\dot{m}_O + \dot{m}_F}$$

The last two components of the array (the ones with indexes “Comb_prod_cp” and “Comb_prod_M”) are set to $c_{P,P}$ and $M_{m,P}$, respectively.

Otherwise, if “Combustion” is equal to FALSE:

$$Y_{P,i} = \frac{Y_{O,i}\dot{m}_O + Y_{F,i}\dot{m}_F}{\dot{m}_O + \dot{m}_F}$$

$$Y_{P,Comb\_prod} = 0$$

In this case, the last two components of the array (the ones with indexes “Comb_prod_cp” and “Comb_prod_M”) are set to 0.

The adiabatic mixing equation between the combustion products and inert mass flow that entered through the ports is used to calculate $T_{t,out}$, which states:

$$(\dot{m}_O + \dot{m}_F)c_{P,P\prime}\left(T_{t,out} - T_c\right) + \dot{m}_{IO}c_{P,IO}\left(T_{t,out} - T_{t,O}\right) + \dot{m}_{IF}c_{P,IF}\left(T_{t,out} - T_{t,F}\right) = 0$$

where $c_{P,IO}$ and $c_{P,IF}$ are the components with index “Comb_prod_cp” of variable “fluid” of the oxidant and fuel port, respectively.

To obtain the “fluid” variable of the outlet port:

$$Y_{out,i} = \frac{Y_{P,i}(\dot{m}_O + \dot{m}_F)}{\dot{m}_{inO} + \dot{m}_{inF}}$$

$$Y_{out,Comb\_prod} = \frac{Y_{P,Comb\_prod}(\dot{m}_O + \dot{m}_F) + \dot{m}_{IO} + \dot{m}_{IF}}{\dot{m}_{inO} + \dot{m}_{inF}}$$

The last two components of the array (the ones with indexes “Comb_prod_cp” and “Comb_prod_M”) are set to the same value as the same components of variable “fluid_P”. This is because, if there are two consecutive “GasGen” components, the introduced values of $c_{P,P}$ and $M_{m,P}$ will be the same because both will work with the same chemicals and $c_{P,P}$ and $M_{m,P}$ are the values for stoichiometric mixture. Therefore, they are properties of the chemicals used and they do not depend on OF…

Finally, the characteristic exhaust velocity can be obtained as:

$$c^* = \frac{\sqrt{R_{g,out}T_{t,out}}}{\Gamma(\gamma_{out})}$$

Once it is calculated, it is sent by the “Info” port to component “ControlPanel” (if it is connected).

Besides, it is important to remember that $1 - Y_{CombProd}$ is the excess amount of oxidiser or fuel. As an example, if there is LOX excess, $Y_{LOX} = 1 - Y_{Comb\_prod}$. Thus, in spite of gases being what exits the gas generator, the excess amount of LOX is considered a liquid inside the “fluid” array because it is the state of the oxygen when it enters the gas generator. Even though it has no physical sense, it is easier for the library to work with it due to the model used.

Initialisation

This component initialises variable “fluid” of the outlet port to “Comb_prod” by using the “Init_fluid” function. It performs it with the maximum priority, because it is the first thing that the program should do to start the calculation. This is because the model of some components changes depending on the state of the fluid, which is set by port variable “fluid”. Variable “fluid” of the outlet port will change once the calculations are performed, but at least, this initialisation fixes that it is a gas that goes out from the port.

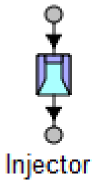
The “Combustion” variable is initialised to TRUE.

4.23 Component Injector

4.23.1 Description

This component type represents an injector.

It is defined in the LPRES_CompCombCha.el file.



4.23.2 Construction Parameters

Name	Type	Default	Units
Type	ENUM OnOffDesign	Design	–

4.23.3 Ports

Name	Type	Direction	Description
f_in	Fluid	IN	Inlet fluid port
f_out	FluidInj	OUT	Outlet fluid injection port

4.23.4 Data

Name	Type	Default	Description	Units	Condition
C_D	REAL	0.5	Discharge coefficient (used only in liquid state) (C_D)	$-$	
A	REAL	0.05	Output area (A)	m^2	Type=Off_design
W	REAL	100	Mass flow ($\dot{m}$)	$\frac{kg}{s}$	Type=Design

4.23.5 Variables

Name	Type	Inital	Description	Units
A_d	REAL		Design output area (A)	m^2
A_sl	REAL		Sonic lock area (calculated only for gases) (A_{sl})	m^2
PR	REAL	10	Pressure ratio (PR)	$-$
PR_sl	REAL		Sonic lock pressure ratio (calculated only for gases) (PR_{sl})	$-$
M_out	REAL		Outlet Mach number (calculated only for gases) (M_{out})	$-$
p_out_ch	REAL		Choked outlet pressure (calculated only for gases) ($p_{out,ch}$)	Pa
v_ideal	REAL		Ideal outlet speed (v_{ideal} for liquids, v for gases)	$\frac{m}{s}$
Re	REAL		Outlet Reynolds number (calculated only for liquids) (Re)	$-$

4.23.5 Formulation

This component represents an injector. It is created in order to inject fluid to a combustion chamber. Its outlet port can only be connected to a “CombCha” component.

Gases

For gases, the model used for this component is almost the same as the one used in component "Nozzle".

The difference is that to know if the outlet is choked or not it needs the combustion chamber pressure which comes from the "FluidInj" port. Meanwhile, component "Nozzle" needs the ambient pressure to know it.

That is the reason why it does not use a "Fluid" port: the component needs the combustion pressure, p_c, and this value can not be calculated from the total pressure of the outlet port. The same happened with the "Nozzle" component and the solution given was to calculate the ambient pressure inside it. In that case it was possible because the ambient pressure is calculated from the altitude, which is a global variable.

The flux inside the injector fulfils (see *Appendix E*):

$$p_t = cte$$

$$T_t = cte$$

From that, it is possible to calculate the Mach number in any section of the injector from the static pressure or vice versa:

$$\left(\frac{p}{p_t}\right)^{\frac{\gamma-1}{\gamma}} = \frac{1}{1+\frac{\gamma-1}{2}M^2}$$

The component calculates the sonic lock pressure ratio, PR_{sl}, which is defined as:

$$PR_{sl} = \frac{p_{t,in}}{p_{out,ch}}$$

and it is the relation between p_t and p when $M = 1$, the maximum Mach number that can be reached in a convergent nozzle if the inlet flux is subsonic, as is the case. Therefore:

$$PR_{sl} = \left(\frac{\gamma+1}{2}\right)^{\frac{\gamma}{\gamma-1}}$$

Finally, the outlet pressure is:

$$p_{out} = max(p_{out,ch}\, , p_c)$$

Furthermore, the pressure ratio is defined:

$$PR = \frac{p_{t,in}}{p_c}$$

The outlet Mach number is calculated with a conditional sentence which depend on whether PR is bigger or smaller than PR_{sl}. If the outlet is choked ($PR > PR_{sl}$):

$$M_{out} = 1$$

otherwise ($PR < PR_{sl}$):

$$M_{out} = \sqrt{\frac{2}{\gamma - 1}\left(PR^{\frac{\gamma-1}{\gamma}} - 1\right)}$$

The last equation comes from the relation between p_t and p that has been previously developed.

Besides:

$$T_{out} = \frac{T_{t,in}}{1 + \frac{\gamma - 1}{2}{M_{out}}^2}$$

and the outlet speed is calculated as:

$$v = M_{out}\sqrt{\gamma R_g T_{out}}$$

Mass flow, $\dot{m}$, is constant throughout the component, so:

$$\dot{m} = \rho v A = \rho M a A = A\rho_t a_t \cdot M\left(\frac{1}{1 + \frac{\gamma - 1}{2}M^2}\right)^{\frac{\gamma+1}{2(\gamma-1)}} =$$

$$= A\frac{p_t}{\sqrt{R_g T_t}} \cdot \sqrt{\gamma} M\left(\frac{1}{1 + \frac{\gamma - 1}{2}M^2}\right)^{\frac{\gamma+1}{2(\gamma-1)}} = A_{sl}\frac{p_t}{\sqrt{R_g T_t}} \cdot \sqrt{\gamma}\left(\frac{2}{\gamma + 1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} =$$

$$= A_{sl}\frac{p_t}{\sqrt{R_g T_t}} \cdot \Gamma(\gamma) = cte$$

where A_{sl} is the outlet area that the nozzle would have if it were choked with the same inlet conditions.

Finally, if Type=Design it is necessary to establish an equation that determines a relationship between A_{sl}, A and M in order to calculate A. This equation comes from the above equation:

$$A M_{out} \rho_t a_t \left(\frac{1}{1+\frac{\gamma-1}{2}{M_{out}}^2}\right)^{\frac{\gamma+1}{2(\gamma-1)}} = A_{sl} \rho_t a_t \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

Hence:

$$\frac{A_{sl}}{A} = \frac{M_{out}}{\left(\frac{2+(\gamma-1){M_{out}}^2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}}}$$

Note that, according to this equation, if $M_{out} = 1$, $A_{sl} = A$.

Otherwise, if Type=Off_design, it uses the mass flow equation to calculate it, because A is known:

$$\dot{m} = A \frac{p_t}{\sqrt{R_g T_t}} \cdot \sqrt{\gamma} M_{out} \left(\frac{1}{1+\frac{\gamma-1}{2}{M_{out}}^2}\right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

Liquids

For liquids, this component uses the Bernoilli equation modified by a discharge coefficient, C_D, to estimate the pressure drop in it. This coefficient is given to the component as data. The modified Bernoulli equation results:

$$p_{t,in} = p_{out} + \frac{1}{2}\rho(v_{ideal})^2$$

where:

$$\dot{m} = \rho v_{ideal} C_D A$$

The subscript "ideal" of v_{ideal} is because it is not the real outlet speed, it is the speed that the liquid would have if there were no pressure drop and the area of the injector were equal to $C_D A$.

Besides:

$$T_{in} = T_{out}$$

and

$$p_c = p_{out}$$

The outled Reynolds number is evaluated for liquids in the following way:

$$Re = \frac{\rho v_{ideal} C_D \sqrt{4A/\pi}}{\mu}$$

Common

Both for gases and for liquids, since this component has not been defined as a "FluidInFluidOut" component, it is necessary to establish that:

$$Y_{i,out} = Y_{i,in}$$

because it is assumed that there is no phase change in it. To establish it, the equation included is:

$$fluid_{out} = fluid_{in}$$

By the way, components defined as "FluidInFluidOut" transmit the auxiliary variables of the "fluid" array between their ports. Besides (see *Appendix C*):

$$\dot{m}_{in} = \dot{m}_{out}$$

This component could not have been defined as a "FluidInFluidOut" component because there is no outlet "Fluid" port. The "Nozzle" component was not defined as a "FluidInFluidOut" component either, but last equations were not necessary because there were not any outlet port.

The variables that are not used because they were defined only for liquids or only for gases are set to 0.

Note that the model of this component is very simple. An injection system is more complex (swirl injectors, etc), so, the calculation of the injection area will not be accurate because the pressure drop will be higher. To simulate the additional pressure drop for gases, the user must place a "Regulator" or a "Pipe" before the injector. For liquids, the additional pressure drop can be included in the value of C_D (by decreasing C_D).

On-off design mode

If Type=Off_Design, the output area (A) will be set as data and the component will calculate the mass flow. Otherwise, if Type=Design, the component will calculate the output area (A) and the mass flow will be set as data.

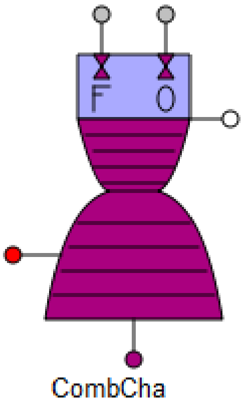
To have a general overview, see *Appendix G*.

4.24 Component CombCha

4.24.1 Description

This component type represents a combustion chamber.

It is defined in the LPRES_CompCombCha.el file



4.24.2 Construction Parameters

Name	Type	Default	Units
Type	ENUM OnOffDesign	Design	–
Cooled	ENUM YesNo	No	–

4.24.3 Ports

Name	Type	Direction	Description
f_O	FluidInj	IN	Inlet fluid injection port
f_F	FluidInj	IN	Inlet fluid injection port
g	GasNozzle	OUT	Outlet gas port through a nozzle
i	Info (n = 1)	OUT	Outlet information port
h	Heat	OUT	Outlet heat exchange port

4.24.4 Data

Name	Type	Default	Description	Units	Condition
eta_d	REAL	0.9	Design combustion efficiency (η_d)	$-$	
OF_st	REAL	8	Stoichiometric mixture ratio (OF_{st})	$-$	
Q_comb	REAL	2000000	Heat of combustion per oxidant mass flow unit (Q_{comb})	$\frac{J}{kg}$	
cp_P	REAL	4182	Specific heat at constant pressure of the products using a stoichiometric mixture ($c_{P,P}$)	$\frac{J}{kg}$	
M_P	REAL	32	Molar mass of the products using a stoichiometric mixture ($M_{m,P}$)	$\frac{J}{kg}$	
AR	REAL	10	Area ratio (ε)	$\frac{J}{kg}$	
A_th	REAL	0.05	Throat area (A_{th})	m^2	Type=Off_design
p_c	REAL	5000000	Design combustion pressue (p_c)	Pa	Type=Design
AR_r	REAL	10/2	Area at the characteristic section of heat exchange divided by the throat area (ε_r)	$-$	Cooled=Yes
A_wet	REAL	1	Nozzle wet area of the cooled zone (A_{wet})	m^2	Cooled=Yes
Zone	ENUM ConvDiv	Divergent	Convergent if the characteristic section of heat exchange is placed in the convergent zone of the nozzle, Divergent if it is placed in the divergent zone		Cooled=Yes

Name	Type	Default	Description	Units	Condition
p_c0	REAL	5000000	Initial combustion pressure for iterative calculations	Pa	
T_c0	REAL	4000	Initial combustion temperature for iterative calculations	$\frac{J}{kg}$	
W_F0	REAL	1000	Initial fuel mass flow for iterative calculations	$\frac{kg}{s}$	

4.24.5 Variables

Name	Type	Initial	Description	Units
T_in	REAL		Average inlet temperature (T_{in})	K
T_c	REAL		Combustion temperature (T_c)	K
eta	REAL		Combustion efficiency (η)	$-$
OF	REAL		Mixture ratio (OF)	$-$
phi	REAL	1	Equivalence ratio (ϕ)	$-$
W_F_st	REAL		Fuel mass flow that provides a stoichiometric mixture when combined with the oxidant mass flow ($\dot{m}_{Fst}$)	$\frac{kg}{s}$
W_F	ALG REAL	W_F0	Fuel mass flow ($\dot{m}_F$)	$\frac{kg}{s}$
W_O	REAL		Oxidant mass flow ($\dot{m}_O$)	$\frac{kg}{s}$
W_IF	REAL		Inert mass flow through the fuel port ($\dot{m}_{IF}$)	$\frac{kg}{s}$
W_IO	REAL		Inert mass flow through the oxidant port ($\dot{m}_{IO}$)	$\frac{kg}{s}$

Name	Type	Initial	Description	Units
fluid_F	HIDDEN REAL[ChemName]		Fuel fluid	$-$
fluid_O	HIDDEN REAL[ChemName]		Oxidant fluid	$-$
fluid_P	HIDDEN REAL[ChemName]		Combustion products fluid	$-$
cp_R	REAL		Reactant specific heat at constant pressure ($c_{P,R}$)	$\frac{J}{kg \cdot K}$
Combustion	BOOLEAN		TRUE if there is combustion and FALSE if either oxidant or fuel is lacking	
Q_comb_effective	REAL		Effective heat of combustion per oxidant mass flow unit ($Q_{comb,ef}$)	$\frac{J}{kg}$
A_th_d	REAL		Design throat area (A_{th})	m^2
A_out	REAL		Output area (A_{out})	m^2
p_out_ch	ALG REAL	100	Choked outlet pressure ($p_{out,ch}$)	Pa
c_star	REAL		Characteristic exhaust velocity (c^*)	$\frac{m}{s}$
A_r	REAL		Area at the characteristic section of heat exchange (A_r)	m^2
M_r	REAL		Mach number at the characteristic section of heat exchange (M_r)	$-$
Pr_r	REAL		Prandtl number at the characteristic section of heat exchange (Pr_r)	$-$
visc_r	REAL		Dynamic viscosity at the characteristic section of heat exchange (μ_r)	$Pa \cdot s$

Name	Type	Initial	Description	Units
T_aw	REAL		Adiabatic wall temperature of combustion gases (T_{aw})	K
h_g	REAL		Combustion gases heat transfer coefficient (h_g)	$\frac{W}{m^2K}$

4.24.6 Formulation

In this component a combustion chamber is modelled. It has two inlets: one for the oxidant and another for the fuel. The model used in this component is a mix between the models used in components “NozzleConvDiv” and “GasGen”.

Combustion

This component needs injectors to introduce the fluid in it. Otherwise, no component of the library can be connected to the inlet fluid injection ports.

The combustion efficiency, comes from:

$$\eta = \eta_d$$

Where η_d is the design combustion efficiency which is established as data.

From the continuity equation, with the same assumptions that have been used before (see *Appendix C*), it is possible to conclude that the inlet mass flow must be the same as the outlet mass flow. Therefore:

$$\dot{m}_{inO} + \dot{m}_{inF} = \dot{m}_{out}$$

Besides:

$$\dot{m}_{inO} = \dot{m}_O + \dot{m}_{IO}$$

$$\dot{m}_{inF} = \dot{m}_F + \dot{m}_{IF}$$

which means that not all the mass flow that comes from the port ($\dot{m}_{inO}$ and $\dot{m}_{inF}$) will react. This can be possible if a “GasGen” component has been placed before. In this case, the outlet of the “GasGen” component will be composed of combustion gases, which are inert, and by excess fuel or oxidant, which can still react. The outlet of the “GasGen” component can be the fuel or oxidant inlet port (depending on whether there is fuel or oxidant excess, respectively) of the “CombCha” component. The leftover fuel or oxidant from the “GasGen” component will react in the “CombCha” component with the oxidant or fuel that enters through the other port.

The nomenclature used is $\dot{m}_O$ and $\dot{m}_F$ to represent the part of the inlet mass flow through the ports ($\dot{m}_{inO}$ and $\dot{m}_{inF}$) that can react. $\dot{m}_{IO}$ and $\dot{m}_{IF}$ represent the part of the inlet mass flow through the ports ($\dot{m}_{inO}$ and $\dot{m}_{inF}$) that cannot react (inert). To calculate them:

$$\dot{m}_O = \dot{m}_{inO} \cdot (1 - Y_{inO,Comb\_prod})$$

$$\dot{m}_F = \dot{m}_{inF} \cdot (1 - Y_{inF,Comb\_prod})$$

Where $Y_{inO,i}$ and $Y_{inF,i}$ are the mass fractions of the fluid that enters through the ports.

Similary, $Y_{O,i}$ and $Y_{F,i}$ are the mass fractions of the part of the inlet mass flow through the ports that can react. To calculate them:

$$Y_{O,i} = \frac{Y_{inO,i}}{(1 - Y_{inO,Comb\_prod})}$$

$$Y_{F,i} = \frac{Y_{inF,i}}{(1 - Y_{inF,Comb\_prod})}$$

$$Y_{O,Comb\_prod} = 0 \quad ; \quad Y_{F,Comb\_prod} = 0$$

$Y_{O,i}$ and $Y_{F,i}$ are contained in variables “fluid_O” and “fluid_F” respectively. The last two components (the ones with the indexes “Comb_prod_cp” and “Comb_prod_M”) of arrays “fluid_O” and “fluid_F” are set to 0.

It may well be that, if fuel enters through the fuel port of the “CombCha”, by mistake, instead of excess oxidant from a previous “GasGen” (whose outlet is the inlet of the oxidant port of the “CombCha”), it is fuel that goes out. Thus, the combustion will not occur in the “CombCha”. The same will happen if oxidant enters through the oxidant port and in the “GasGen” there is oxidant excess instead of fuel excess. To simulate it, variables “Combustion” and “Q_comb_effective” ($Q_{comb,ef}$) have been created. If this mistake occurs, “Combustion” will be equal to FALSE and $Q_{comb,ef} = 0$. Otherwise, “Combustion” will be equal to TRUE and $Q_{comb,ef} = Q_{comb}$. To know if “Combustion” is true or false, the component performs:

$$\sum_{i}^{LiquidsGases} Y_{O,i} \cdot Y_{F,i}$$

If the result is equal to 0, “Combustion” will be TRUE because it means that the oxidant is different to the fuel, as it should be. Otherwise, If the result is not 0, “Combustion” will be FALSE because it means that the oxidant and the fuel are the same chemical, which is a mistake.

The mixture ratio, OF, is defined as:

$$OF = \frac{\dot{m}_O}{\dot{m}_F}$$

The equivalence ratio, ϕ, is:

$$\phi = \frac{OF_{st}}{OF}$$

and the fuel mass flow that provides a stoichiometric mixture when combined with the oxidant mass flow is:

$$\dot{m}_{Fst} = \frac{\dot{m}_F}{\phi}$$

To calculate the combustion temperature, T_c, the combustion model states (see *Apendix F*):

$$\eta \cdot Q_{comb,ef} = \frac{1 + OF}{min(OF\ , OF_{st})} \cdot \left(c_{P,P'}\left(T_c - T_{ref}\right) - c_{P,R}\left(T_{in} - T_{ref}\right)\right)$$

where T_{in} is calculated as (see *Apendix F*):

$$\left(1 + \frac{\phi}{OF_{st}}\right) c_{P,R} T_{in} = c_{P,O} T_{t,O} + \frac{\phi}{OF_{st}} c_{P,F} T_{t,F}$$

where $c_{P,R}$ (see *Apendix F*):

$$c_{P,R} = \frac{\dot{m}_O c_{P,O} + \dot{m}_F c_{P,F}}{\dot{m}_O + \dot{m}_F}$$

Note that for liquids $T = T_t$. To calculate $c_{P,P'}$, $c_{P,F}$ and $c_{P,O}$, function “cp” is called, where variables “fluid_P”, “fluid_F” and “fluid_O” are the function inputs, respectively.

To calculate variable “fluid_P” (see *Apendix F*), if “Combustion” is equal to TRUE:

$$Y_{P,i} = \frac{\max(1 - \phi, 0) \cdot Y_{O,i} \dot{m}_O + \max(\phi - 1,0) \cdot Y_{F,i} \dot{m}_{Fst}}{\dot{m}_O + \dot{m}_F}$$

$$Y_{P,Comb\_prod} = \frac{(1 - \max(1 - \phi, 0)) \cdot (\dot{m}_O + \dot{m}_{Fst})}{\dot{m}_O + \dot{m}_F}$$

The last two components of the array (the ones with indexes “Comb_prod_cp” and “Comb_prod_M”) are set to $c_{P,P}$ and $M_{m,P}$, respectively.

Otherwise, if “Combustion” is equal to FALSE:

$$Y_{P,i}=\frac{Y_{O,i}\dot{m}_O+Y_{F,i}\dot{m}_F}{\dot{m}_O+\dot{m}_F}$$

$$Y_{P,Comb\_prod}=0$$

In this case, the last two components of the array (the ones with indexes "Comb_prod_cp" and "Comb_prod_M") are set to 0.

The adiabatic mixing equation between the combustion products and inert mass flow that entered through the ports is used to calculate $T_{t,out}$, which states:

$$(\dot{m}_O+\dot{m}_F)c_{P,P'}(T_{t,out}-T_c)+\dot{m}_{IO}c_{P,IO}(T_{t,out}-T_{t,O})+\dot{m}_{IF}c_{P,IF}(T_{t,out}-T_{t,F})=0$$

where $c_{P,IO}$ and $c_{P,IF}$ are the component with index "Comb_prod_cp" of variable "fluid" of the oxidant and fuel port, respectively.

To obtain the "fluid" variable of the outlet port:

$$Y_{out,i}=\frac{Y_{P,i}(\dot{m}_O+\dot{m}_F)}{\dot{m}_{inO}+\dot{m}_{inF}}$$

$$Y_{out,Comb\_prod}=\frac{Y_{P,Comb\_prod}(\dot{m}_O+\dot{m}_F)+\dot{m}_{IO}+\dot{m}_{IF}}{\dot{m}_{inO}+\dot{m}_{inF}}$$

The last two components of the array (the ones with indexes "Comb_prod_cp" and "Comb_prod_M") are set to the same value as the same components of variable "fluid_P". This is because, if there is a "GasGen" component and later a "CombCha" component, the introduced values of $c_{P,P}$ and $M_{m,P}$ will be the same because both will work with the same chemicals and $c_{P,P}$ and $M_{m,P}$ are the values for stoichiometric mixture. Therefore, they are properties of the chemicals used and they do not depend on OF…

Finally, the characteristic exhaust velocity can be obtained as:

$$c^*=\frac{\sqrt{R_{g,out}T_{t,out}}}{\Gamma(\gamma_{out})}$$

Once it is calculated, it is sent by the "Info" port to component "ControlPanel" (if it is connected).

Besides, it is important to remember that $1-\ Y_{CombProd}$ is the excess amount of oxidiser or fuel. As an example, if there is LOX excess, $Y_{LOX}=1-\ Y_{Comb\_prod}$. Thus, in spite of gases being what exits the gas generator, the excess amount of LOX is considered as liquid inside the "fluid" array because it is the state of the oxygen when it enters the gas generator. Even though it has no physical sense, it is easier for the library to work with it due to the model used.

Initialisation (combustion)

This component initialises variable “fluid” of the outlet port to “Comb_prod” by using the “Init_fluid” function. It performs it with the maximum priority, because it is the first thing that the program should do to start the calculation. This is because the model of some components changes depending on the state of the fluid, which is set by port variable “fluid”. Variable “fluid” of the outlet port will change once the calculations are performed, but at least, this initialisation fixes that it is a gas that goes out from the port.

The “Combustion” variable is initialised to TRUE.

Convergent-divergent nozzle

Now, the equations that it gets from component “NozzleConDiv” are rewritten. It uses the “GasNozzle” port, that has been created to be able to add nozzle extensions and to calculate the thrust of the rocket, because the components that do it will need some extra information than the one sent by a common “Fluid” port. They will need the outlet area of the nozzle.

The flow inside the nozzle fulfils (see *Appendix E*):

$$p_t = cte$$

$$T_t = cte$$

Since it is choked:

$$p_{out} = p_{out,ch}$$

and the choked outlet pressure comes from the following equation:

$$\frac{A_{out}}{A_{th}} = \frac{\Gamma(\gamma_{out})}{\left(\frac{p_{out,ch}}{p_{t,out}}\right)^{\frac{1}{\gamma}} \sqrt{\frac{2 \cdot \gamma_{out}}{\gamma_{out} - 1}\left(1 - \left(\frac{p_{out,ch}}{p_{t,out}}\right)^{\frac{\gamma_{out}-1}{\gamma_{out}}}\right)}}$$

Which comes from the same equation used in the “Nozzle” component that sets a constant mass flow, $\dot{m}$, throughout the component, knowing that the section in which $M = 1$ is the throat:

$$\dot{m} = A\frac{p_t}{\sqrt{R_g T_t}} \cdot \sqrt{\gamma} M \left(\frac{1}{1 + \frac{\gamma - 1}{2} M^2}\right)^{\frac{\gamma+1}{2(\gamma-1)}} =$$

$$= A_{out}\frac{p_{t,out}}{\sqrt{R_{g,out}T_{t,out}}}\cdot\sqrt{\gamma_{out}}M_{out}\left(\frac{1}{1+\frac{\gamma_{out}-1}{2}M_{out}{}^{2}}\right)^{\frac{\gamma_{out}+1}{2(\gamma_{out}-1)}} =$$

$$= A_{th}\frac{p_{t,out}}{\sqrt{R_{g,out}T_{t,out}}}\cdot\Gamma(\gamma_{out})$$

and

$$M_{out} = \sqrt{\frac{2}{\gamma_{out}-1}\left(\left(\frac{p_{t,out}}{p_{out,ch}}\right)^{\frac{\gamma_{out}-1}{\gamma_{out}}}-1\right)}$$

but one thing must be kept in mind: the equation that sets a constant mass flow has two available solutions to M_{out}: a subsonic solution and a supersonic solution. That is the reason of why an initial value is given for variable "p_out_ch". The given value is 100 because this means that the iterations will start in low pressure values which would meab that the divergence of the nozzle is working with subsonic flow.

Moreover, this equation is also included in the component to relate the throat area and the mass flow:

$$\dot{m} = A_{th}\frac{p_{t,out}}{\sqrt{R_{g,out}T_{t,out}}}\cdot\Gamma(\gamma_{out})$$

The area ratio, ε, is defined as:

$$\varepsilon = \frac{A_{out}}{A_{th}}$$

Besides, this component sends A_{out} through the "GasNozzle" port.

Finally, as already explained in the development of the injectors, it is necessary to send the combustion pressure through the inlet ports, so:

$$p_{t,out} = p_{c,O}$$

$$p_{t,out} = p_{c,F}$$

On-off design mode (convergent-divergent nozzle)

If Type=Off_Design, the throat area (A_{th}) will be set as data and the component will calculate the combustion pressure (which is equal to $p_{t,out}$ because p_t is constant along the nozzle). Otherwise, if Type=Design, the component will calculate the throat area (A_{th}) and the combustion pressure (p_c) will be set as data, where:

$$p_c = p_{t,out}$$

To have a general overview, see *Appendix G*.

Initialisation (convergent-divergent nozzle)

The initial value given for $p_{t,out}$ is “p_c0” and for $T_{t,out}$ is “T_c0”. These values are given to help the convergence to be reached. It is simply an initial value to start the iterations; it is not the final value.

Cooling

If Cooled=No, component “CombCha” is not refrigerated by a cooling jacket.

Otherwise, if Cooled=Yes, the component is refrigerated by a cooling jacket. It calculates the heat flux exchanged between the combusted gases and thrust chamber wall $(\dot{Q})$. The heat transfer mechanisms are convection and radiation.

In the thrust chamber, before the combusted gases can transfer heat to the wall, the heat energy must pass through a layer of stagnant gas along the wall, boundary layer. This basic correlation for this complicated convective heat transfer can be expressed by the following equation:

$$\dot{Q} = A_{wet} h_g (T_{aw} - T_{w,hot})$$

A_{wet} is established by the user as data.

The adiabatic wall temperature of combustion gas at a given location in the thrust chamber may be obtained from the following expression (Iqbal, Sheikh, Ali, Khushnood, & Arif, 2012):

$$T_{aw} = T_{t,out} \left[\frac{1 + r\left(\frac{\gamma - 1}{2}\right) M^2}{1 + \left(\frac{\gamma - 1}{2}\right) M^2} \right]$$

where recovery factor r can be estimated for turbulent flows as:

$$r = (Pr)^{0.33}$$

In this component, the location of the thrust chamber in which T_{aw} is evaluated is given by the user. The user must set data ε_r, which is the area at the characteristic section of heat exchange divided by the throat area:

$$\varepsilon_r = \frac{A_r}{A_{th}}$$

A_r is the area of the section in which the fluid properties are evaluated in order to calculate the heat flux exchanged between the combusted gases and thrust chamber wall $(\dot{Q})$. Moreover, this section can be either in the convergent or in the divergent part of the nozzle. To select the desired part, set variable "Zone" to "Convergent" or to "Divergent" respectively. The characteristic section of heat exchange is usually placed in the convergent part of the nozzle if the combustion chamber plus the convergent region of the nozzle is longer than the divergent region and vice versa.

To evaluate T_{aw}, $M = M_r$ and $Pr = Pr_r$.

Hence, to calculate the Mach number at the characteristic section:

$$\varepsilon_r = \frac{\Gamma(\gamma_{out})}{\sqrt{\gamma_{out}}M_r\left(\frac{1}{1+\frac{\gamma_{out}-1}{2}{M_r}^2}\right)^{\frac{\gamma_{out}+1}{2(\gamma_{out}-1)}}}$$

This equation comes from the continuity equation developed above. This equations has two M_r solutions for the same ε_r value: the subsonic solution and the supersonic solution. Therefore, if Zone=Convergent, the subsonic solution will be the right one. Consequently, the initial value for M_r to start the iterations needed to solve the equation in this case will be 0.001. Otherwise, if Zone=Divergent, the supersonic solution will be the right one. Consequently, the initial value for M_r to start the iterations needed to solve the equation in this case will be 100.

To calculate the Prandtl number, kinetic theory can be used to get the approximate result (Bartz, 1957):

$$Pr_r = \frac{4\gamma_{out}}{9\gamma_{out}-5}$$

To calculate h_g, the model used is the Bartz Correlation (Bartz, 1957) which is:

$$h_g = \left[\frac{0.026}{{D_{th}}^{0.2}}\left(\frac{\mu^{0.2}c_p}{Pr^{0.6}}\right)\left(\frac{p_c}{c^*}\right)^{0.8}\right]\left(\frac{A_{th}}{A}\right)^{0.9}\sigma$$

where D_{th} is the throat diameter and σ a factor which contains all the corrections for property variations across the boundary layer. Note that this equation is dimensionally consistent.

In this component, the Bartz Correlation is evaluated in the characteristic section of heat exchange and an additional simplification has been done: $\sigma = 1$. Finally:

$$h_g = \left[\frac{0.026}{\left(\sqrt{4 \cdot \frac{A_{th}}{\pi}}\right)^{0.2}} \left(\frac{{\mu_r}^{0.2} c_{p,out}}{{Pr_r}^{0.6}}\right) \left(\frac{p_{t,out}}{c^*}\right)^{0.8} \right] \left(\frac{A_{th}}{A_r}\right)^{0.9}$$

To calculate the dynamic viscosity at the characteristic section of heat exchange, the NBS data for the viscosity of air at high temperatures can be used to get a correlation equation which should be reasonably accurate for most mixtures consisting principally of diatomic gases (Bartz, 1957):

$$\mu_r = 1.184 \cdot 10^{-7} \cdot \left(M_{m,out}\right)^{0.5} (T_{aw})^{0.6}$$

Being T_{aw} and $M_{m,out}$ measured in standard SI units.

The variables that are sent by the "Heat" port are $\dot{Q}$, $T_{w,hot}$ and A_{wet}. If Cooled=Yes and the "CoolingJacket" component is not connected, only the variable A_{wet} is known. Besides, the relation between Q and $T_{w,hot}$ is known, and has already been shown: $\dot{Q} = A_{wet} h_g (T_{aw} - T_{w,hot})$.

If the component is not refrigerated, M_r is set to 1, A_r to A_{th}, h_g to 0, the port variables $\dot{Q}$ and A_{wet} are set to 0 and the port variable $T_{w,hot}$ is set to T_{aw}.

Error Messages (cooling)

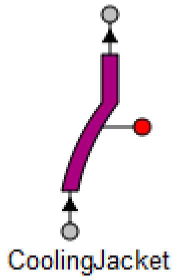
If $\varepsilon_r < 1$, a KILLPOINT ASSERT will appear.

4.25 Component CoolingJacket

4.25.1 Description

This component type represents a cooling jacket.

It is defined in the LPRES_CompHeat.el file.



4.25.2 Construction Parameters

Name	Type	Default	Units
Type	ENUM Type_Cooling	Darcy	$-$

4.25.3 Ports

Name	Type	Direction	Description
l	Fluid	IN	Inlet fluid port
g	Fluid	OUT	Outlet fluid port
h	Heat	IN	Inlet heat exchange port

4.25.4 Data

Name	Type	Default	Description	Units	Condition
L	REAL	1	Length of the channels (L)	m	
a	REAL	0.002	Average width of the channels (a)	m	
b	REAL	0.004	Average height of the channels (b)	m	
N	INTEGER	100	Number of channels (N)	$-$	
rug	REAL	5e-5	Absolute rugosity (ε)	m	
k_w	REAL	370	Thermal conductivity of the nozzle wall material (k_w)	$\frac{W}{m \cdot K}$	
t_w	REAL	0.003	Nozzle wall thickness (t_w)	m	
dp	REAL	1e5	Imposed total pressure drop if Type=Known_dp (dp)	Pa	Type=Known_dp

4.25.5 Variables

Name	Type	Inital	Description	Units
Q	REAL		Heat flux ($\dot{Q}$)	W
T_w_hot	REAL		Wall temperature of the combustion gases side ($T_{w,hot}$)	K
T_w_cold	REAL	500	Wall temperature of the cooling liquid side ($T_{w,cold}$)	K
A_wet_cooling	REAL		Cooling wet area ($A_{wet,cooling}$)	m^2
A_wet_nozzle	REAL		Nozzle wet area ($A_{wet,nozzle}$)	m^2
h_l	REAL		Cooling liquid heat transfer coefficient (h_l)	$\frac{W}{m^2K}$
Pr	REAL		Cooling liquid Prandtl number (Pr)	$-$
Nu	REAL		Cooling liquid Nusselt number (Nu)	$-$
Re	REAL		Cooling liquid Reynolds number (Re)	$-$
v	REAL		Cooling liquid speed (v)	$\frac{m}{s}$
f	REAL		Darcy friction factor (f)	$-$
D_eq	REAL		Circular equivalent diameter of a rectangular duct for equal friction and flow capacity (D_{eq})	m
D_hy	REAL		Hydraulic diameter (diameter of the circle with the same area) (D_{hy})	m

4.25.6 Formulation

This component is a cooling jacket to refrigerate component “CombCha”. A phase transition occurs in this component.

Variables $\dot{Q}$, $T_{w,hot}$ and $A_{wet,nozzle}$ come through the “Heat” port.

This component uses the “Vaporisation” and “Init_fluid” functions to set the “fluid” outlet port variable. The working fluid is changed because some fluids in the library are defined twice, once for liquid state and another for gas state.

As this component has not been defined as a “FluidInFluidOut” component because there is a phase transition inside it, it is necessary to establish that (see *Appendix C*):

$$\dot{m}_l = \dot{m}_g$$

Heat transfer in a regeneratively cooled chamber can be described as the heat flow between two moving fluids, through a multilayer partition.

$$\dot{Q} = cte$$

where $\dot{Q}$ is the same value that enters through the port.

The heat transfer between the coolant and thrust chamber wall is by forced convection (Iqbal, Sheikh, Ali, Khushnood, & Arif, 2012):

$$\dot{Q} = A_{wet,cooling} h_l (T_{w,cold} - T_l)$$

where T_l is the inlet liquid temperature. $A_{wet,cooling}$ can be calculated from:

$$A_{wet,cooling} = N \cdot 2 \cdot (a + b) \cdot L$$

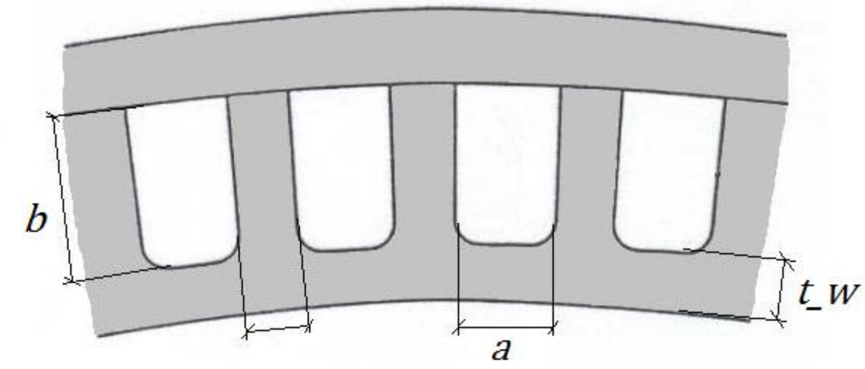


Figure 12 – Cooling channels geometry

To calculate h_l, the correlations used for coolant side heat transfer are mainly based on the conventional Dittus-Boelter equation for turbulent, thermally fully developed flow for fluids with constant property values. The following correlation is generally used for regenerative cooling analysis (Iqbal, Sheikh, Ali, Khushnood, & Arif, 2012):

$$Nu = \frac{h_l D_{eq}}{k_l} = 0.027 \cdot Re^{0.8} Pr^{0.33}$$

D_{eq} is the equivalent diameter, which is the diameter of a circular duct or pipe that gives the same pressure loss as an equivalent rectangular duct or pipe. It is calculated by using the Huebscher Correlation (1948), based on the air friction chart developed by Wright (1945). The Huebscher Correlation is:

$$D_{eq} = 1.3 \frac{(a \cdot b)^{0.625}}{(a + b)^{0.25}}$$

To calculate the Prandtl number:

$$Pr = \left(\frac{\mu \cdot c}{k}\right)_l$$

To calculate the Reynolds number:

$$Re = \left(\frac{\rho v D_{eq}}{\mu}\right)_l$$

where:

$$v = \frac{\dot{m}}{\rho \cdot (a \cdot b \cdot N)}$$

To calculate a relationship between $T_{w,hot}$ and $T_{w,cold}$, the heat flux passes through the first metallic wall of thickness t_w so that (Iqbal, Sheikh, Ali, Khushnood, & Arif, 2012):

$$\dot{Q} = A_{wet,nozzle} \frac{k_w}{t_w} \left(T_{w,hot} - T_{w,cold}\right)$$

Besides, applying the energy equation to the fluid:

$$\dot{Q} = \dot{m} \cdot c_{P,g} \cdot (T_{t,g} - T_l)$$

Note that this is the only case in the component that the gas properties are used to perform a calculation. In all other cases, the liquid properties are used to evaluate Re, Pr, etc. This has been decided by testing the component and evaluating the results, and the chosen model is the best one. It must be done in that way because, since it is a basic library, the fluid properties do not vary with temperature, etc., but they change for liquid or gas state. Besides, the component is not discretised. Therefore, the properties can only be evaluated at the inlet (liquid) or at the outlet (gas). This is not a problem in the other components because there is not phase change.

Finally, the model used to simulate the pressure drop of the cooling jacket is the known Darcy-Weisbach equation:

$$p_{t,g} = p_{t,l} - \frac{1}{2}\rho v^2 \left(f\frac{L}{D_{hy}}\right)$$

This model calculates the pressure drop due to friction if Type=Darcy.

This component uses function “hdc_fric” to calculate the friction factor:

$$f = hdc\_fric(D_{hy}, \varepsilon, Re)$$

To calculate the hydraulic diameter, D_{hy}:

$$D_{hy} = \frac{4 \cdot Area}{Perimeter} = \frac{2ab}{a+b}$$

Otherwise, this component has the possibility, if Type=Known_dp, of imposing the pressure drop instead of using the Darcy-Weisbach equation:

$$p_{t,g} = p_{t,l} - dp$$

Error Messages

Only liquids can enter this component. If the working fluid is not in a liquid state, a FATAL ASSERT will appear.

Initialisation

This component initialises outlet port variable “fluid” by using functions “Vaporisation” and “Init_fluid”. It performs it with 90% priority, because it is the second thing that the program should do to start the calculation. The initialisation is done with 100% priority in the other components that set the “fluid” variable. This is because the models of some components change depending on the state of the fluid, which is set by the “fluid” port variable. Hence, this variable must be initialised in first position. But, for this component, it is necessary for the inlet fluid of the component to use the “Vaporisation” function in order to know the outlet fluid. Hence, initialisation of the other components that sets the “fluid” variable must be done beforehand.

5 References

Bartz, D. R. (1957). A Simple Equation for Rapid Estimation of Rocket Nozzle Convective Heat Transfer Coefficients. *Jet Propulsion* , January, pp.49-51.

Brokaw, R. (1968). *Viscosity of Gas Mixtures.* NASA TECHNICAL NOTE D-4496.

Burkhardt, H., Sippel, M., Herbertz, A., & Klevanski, J. (2002). Comparative Study of Kerosene and Methane Propellant Engines for Reusable Liquid Booster Stages. *4th International Conference on Launcher Technology "Space Launcher Liquid Propulsion".* Liège (Belgium).

Churchill, S. (1977). Friction factor equation spans all fluid-flow regimes. *Chemical Engineering* , 84, pp. 91-92.

de Assas, P., Barrero, A., Graña, J. C., & Pérez, I. (2013). *Termodinámica.* ETSIAE Publicaciones.

Di Mateo, F. (2010-2011). *Modelling and Simulation of Liquid Rocket Engine Ignition Transients (PhD thesis).* Università di Roma.

Empresarios Agrupados. (2015). *ESPSS Software Transfer Document.*

Empresarios Agrupados. (2015). *ESPSS: European Space Propulsion System Simulation - EcosimPro Libraries User Manual.*

Empresarios Agrupados. (2015). *Software Verification and Validation Plan.*

Franchini, S., & López García, Ó. (2012). *Introducción a la Ingeniería Aeroespacial.* Garceta.

Iqbal, M., Sheikh, N., Ali, H., Khushnood, S., & Arif, M. (2012). Comparison of empirical correlations for the estimation of conjugate heat transfer in a thrust chamber. *Life Science Journal* , 9(4), pp.708-716.

ISO 2533:1975. *Standard Atmosphere.*

Liñán, A., Rodríguez, M., & Higuera, F. J. (2013). *Mecánica de Fluidos. Tomo I.* ETSIAE Publicaciones.

Liñán, A., Rodríguez, M., & Higuera, F. J. (2013). *Mecánica de Fluidos. Tomo II.* ETSIAE Publicaciones.

Mårtensson, H., Andersson, S., Trollheden, S., & Brodin, S. (2007). Rocket Engines: Turbomachinery. *RTO-EN-AVT-150 - Advances on Propulsion Technology for High-Speed Aircraft* .

Sutton, G. P., & Biblarz, O. (2001). *Rocket Propulsion Elements.* John Wiley & Sons.

Tizón, J. M. (2015). *Motores Cohete.* ETSIAE Publicaciones.

Tizón, J. M. (2015). *Turbomáquinas.* ETSIAE Publicaciones.

Vilá Vacas, J. (2012). *Diseño y Optimización de Motores Cohete de Propulstante líquido con EcosimPro/ESPSS (Master's Thesis).* Universidad Politécnica de Madrid.

Appendix A: Nomenclature

In this appendix, it will be explained how is the pattern that has been followed to write the code of the library.

- To choose the type name of all components in the library, the iten name has been used, beginning with a capital letter. If the item name has more than one word, all the words are included in the type name, each one beginning with a capital letter and with no spaces between them. If a word is too long, it has been shortened in order to make the programming easier.

- If the variables, etc. of the library are accompanied by a subscript, an underscore is used to represent it in the code except in cases such as, for example, the total pressure or the total temperature because they will usually have another subscript. The code is merely “pt” and “Tt”, which would represent p_t and T_t respectively.

- If a variable, etc. is to be represented by a greek letter, the name of the variable in the code will be the full name of the letter in english.

- To write the equations, a space is given between each variable and each mathematical sign except after an open parenthesis, before a close parenthesis and if it is something raised to a power.

- If an equation works with real numbers, each integer included in it should be followed by a point.

- To write the equations, a line is left between two equations if these two equations do not have a close relationship.

Subscripts “out”, “in”, “O”, “F”, etc. have also been used in this manual so as to refer a magnitude with a port.

Symbols

The arrows drawn in the symbols of some components of the library indicate the sense of the flow in the flow line. If a component is connected upside-down with respect to the other components of the flow line, it will not work properly.

The sky blue colour has mainly been used in the symbols of liquid components.

The gray colour has mainly been used in the symbols of gas components.

The purple colour has mainly been used in the symbols of components related to the nozzle.

The brown colour has mainly been used in the symbols of mechanical components.

The green colour has mainly been used in the symbols of metering components.

Appendix B: Tables of physical properties

Liquids

Name	Fluid	Molar mass (M_m) [g/mol]	Specific heat (c) [J/(kg•K)]	Density (ρ) [kg/m^3]	Thermal conductivity (k) [W/(m•K)]	Dynamic viscosity (μ) [centipoise]
LOX	Liquid oxygen (O_2)	31.9988	1680	1200	0.149	0.19
NTO	Nitrogen tetroxide (N_2O_4). Also valid for MON	92.011	1600	1477	0.131	0.423
H2O2	Hydrogen peroxide (H_2O_2)	34.0147	2629	1450	0.353	1.27
HNO3	Nitric acid (HNO_3)	63.01	1720	1560	0.294	1
LF2	Liquid fluorine (F_2)	38	1640	1500	0.1	0.079
RP_1	Rocket Propellat	172	2093	809	0.137	0.21
LCH4	Liquid methane (CH_4)	16.0426	3480	422	0.15	0.02
LH2	Liquid hydrogen (H_2)	2.01594	7320	73	0.117	0.02
N2H4	Hidrazine (N_2H_4)	32.04516	3080	1008	0.488	0.97
UDMH	Undymmetrical dimethyldrazine ($(CH_3)_2N_2H_2$)	60.1	2720	783	0.035	0.754
MMH	Monomethyldrazine (CH_3NHNH_2)	46.07	2840	874	0.246	0.4
JP_10	Jet fuel	136.234	1675	920	0.11	2.8
Kerox	Kerosene	170.34	1800	749	0.17	4.5
Oil	Oil	60	1800	950	0.12	4.5
H2O	Water (H_2O)	18	4182	1000	0.607	1

Name	Fluid	Molar mass (M_m) [g/mol]	Specific heat (c) [J/(kg•K)]	Density (ρ) [kg/m^3]	Thermal conductivity (k) [W/(m•K)]	Dynamic viscosity (μ) [centipoise]
IPA	Isopropyl alcohol (C_3H_8O)	60.1	2400	786	0.145	2

Gases

Name	Fluid	Molar mass (M_m) [g/mol]	Specific heat at constant pressure (c_P) [J/(kg•K)]
Air	Air	28.958538	1004
Ar	Argon (Ar)	39.948	520
CH4	Methane (CH_4)	16.0426	3000
CO	Carbon monoxide (CO)	28.0104	1080
CO2	Carbon dioxide (CO_2)	44.0098	1000
H2	Hydrogen (H_2)	2.01594	15800
He	Helium (He)	4.0026	5193
N2	Nitrogen (N_2)	28.01348	1039
O2	Oxygen (O_2)	31.9988	1000
MMH_vapour	Vapour monomethyldrazine (CH_3NHNH_2)	46.07	1600

Enum “Liquid” is composed of the names of the liquids.

Enum “Gases” is composed of the names of the gases.

Enum “LiquidsGases” is composed of both, the names of the gases and the liquids.

Enum “ChemName” is composed of the names of the gases, the name of the liquids and “Comb_prod”, “Comb_prod_M” and “Comb_prod_cp”.

Enum “LV” is composed of “LOX”, “LCH4”, “LH2” and “MMH”.

Array “Chem_M” is composed of the molar mass of liquids and gases.

Array “Chem_cp” is composed of the specific heat of liquids and the specific heat at constant pressure of gases.

Array “Chem_rho” is composed of the density of liquids.

Array “Chem_cond” is composed of the thermal conductivity of liquids.

Array “Chem_visc” is composed of the dynamic viscosity of liquids.

Appendix C: The continuity equation for the FluidInFluidOut components

The continuity equation is one of the Navier-Stokes equations. In its integral form:

$$\frac{d}{dt}\int_{V_c(t)} \rho dV + \int_{\Sigma_c(t)} \rho(\vec{v} - \vec{v}_c) \cdot \vec{n} dA = 0$$

Throughout the whole library, one of the hypotheses made is that the Strouhal number, defined as:

$$St = \frac{l_c}{v_c \cdot t_c} = \frac{t_r}{t_c}$$

is $St \ll 1$, so the model can be considered as stationary and the temporal term is negligible. Therefore, the equation results:

$$\int_{\Sigma_c(t)} \rho(\vec{v} - \vec{v}_c) \cdot \vec{n} dA = 0$$

If the control surface is fixed ($\Sigma_c \neq \Sigma_c(t)$ and $\vec{v}_c = 0$) and its form is:

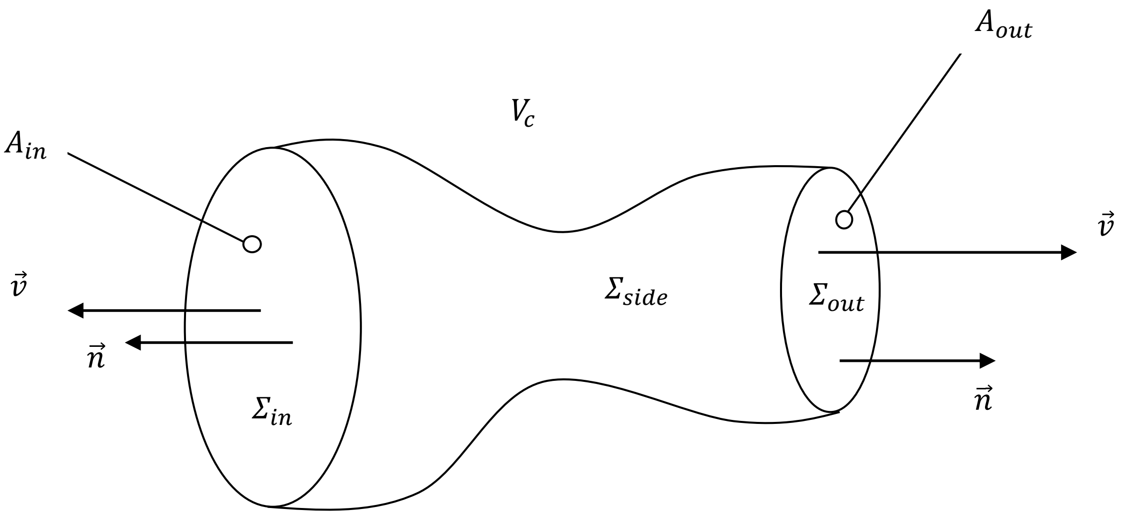


Figure 13 - Component GasInGasOut control volume

$$\Sigma_c = \Sigma_{out} + \Sigma_{in} + \Sigma_{side}$$

the equation can be divided as:

$$\int_{\Sigma_{in}} \rho\vec{v}\cdot\vec{n}dA + \int_{\Sigma_{out}} \rho\vec{v}\cdot\vec{n}dA + \int_{\Sigma_{side}} \rho\vec{v}\cdot\vec{n}dA = 0$$

In the side surface: $\vec{v}\cdot\vec{n} = 0$ because it is an impervious surface. In Σ_{in}, $\vec{v}$ is parallel to $\vec{n}$, and ρ and $\vec{v}$ are the same for all the surface. The same goes for Σ_{out}. Integrating:

$$\rho_{out}\cdot v_{out}\cdot A_{out} - \rho_{in}\cdot v_{in}\cdot A_{in} = 0$$

Where:

$$(\vec{v}\cdot\vec{n})_{in} = -v_{in}$$

$$(\vec{v}\cdot\vec{n})_{out} = v_{out}$$

Because v is the velocity, $\vec{v}$, multiplied by the unit vector that indicates the longitudinal direction of the duct, $\vec{n}$ (this vector will always be perpendicular to the cross sectional area of the duct).

The mass flow is defined as:

$$\dot{m} = \int_{\Sigma} \rho \vec{v} \cdot \vec{n} dA$$

Hence, following the same process that has been followed before, the result is as follows:

$$\dot{m} = \rho v A$$

Replacing this in the equation that resulted from the continuity equation:

$$\dot{m}_{in} = \dot{m}_{out}$$

Appendix D: The energy equation for the GasTurbo components

The energy equation is other of the Navier-Stokes equations. In its integral form:

$$\frac{d}{dt}\int_{V_c(t)} \rho(e+\frac{v^2}{2})dV + \int_{\Sigma_c(t)} \rho(e+\frac{v^2}{2})(\vec{v}-\vec{v_c})\cdot\vec{n}dA =$$

$$= -\int_{\Sigma_c(t)} p\vec{v}\cdot\vec{n}dA + \int_{V_c(t)} \rho\overrightarrow{f_m}\cdot\vec{v}dV +$$

$$+\int_{\Sigma_c(t)} \left(\overline{\overline{\tau'}}\cdot\vec{v}\right)\cdot\vec{n}dA + \int_{\Sigma_c(t)} k\nabla T\cdot\vec{n}dA + \int_{V_c(t)} q_v dV$$

As in the case of the continuity equation (*Appendix C*), the temporal term is null. The control volune is also the same.

The second term of the first member of the equation is reduced to the energy flow through the inlet and outlet sections.

The Reynolds number:

$$Re = \rho_c\frac{v_c\cdot l_c}{\mu_c}$$

is high enough to consider the viscous effects to be negligible.

The heat input per unit of volume and time (both, heat received by radiation and by chemical reactions), q_v, like the heat received by conduction, $\vec{q} = k\nabla T$, are considered null (the last one is also because $Re \gg 1$).

The mass forces, $\overrightarrow{f_m}$, are also negligible because the Froude number, defined as:

$$Fr = \frac{{v_c}^2}{f_{m,c} \cdot l_c}$$

is high.

Finally, the work exerted by the pressure forces on the movable surfaces is the power received by the fluid, W.

Consequently:

$$W = \int_{\Sigma_{in}+\Sigma_{out}} \rho(e + \frac{v^2}{2} + \frac{p}{\rho})\vec{v} \cdot \vec{n}dA$$

With the assumption that the total enthalpy, defined as:

$$h_t = e + \frac{v^2}{2} + \frac{p}{\rho}$$

is uniform along Σ_{in} and Σ_{out}, the equation can be written as:

$$W = \dot{m} \cdot (h_{t,in} - h_{t,out})$$

The final equation is:

$$W = \dot{m} \cdot c_P \cdot (T_{t,in} - T_{t,out})$$

Appendix E: The momentum equation for duct components

In this appendix, the yet undeveloped Navier-Stokes equation: the momentum equation, will be developed. Its integral form is:

$$\frac{d}{dt}\int_{V_c(t)} \rho\vec{v}dV + \int_{\Sigma_c(t)} \rho\vec{v}(\vec{v}-\vec{v_c})\cdot\vec{n}dA = -\int_{\Sigma_c(t)} p\vec{n}dA + \int_{V_c(t)} \rho\overrightarrow{f_m}dV + \int_{\Sigma_c(t)} \overline{\overline{\tau'}}\cdot\vec{n}dA$$

but in this case it is the differential form that will be analysed:

$$\rho\frac{\partial\vec{v}}{\partial t} + \rho\vec{v}\cdot\nabla\vec{v} = -\nabla p + \rho\overrightarrow{f_m} + \nabla\cdot\overline{\overline{\tau'}}$$

As with the other two equations (see *Apendix C* and *Appendix D*), the temporal term is null.

The Reynolds number is high enough to consider the viscous effects to be negligible. The mass forces, $\overrightarrow{f_m}$, are also negligible because the Froude number is high.

Considering the previous statements and projecting in streamlines:

$$\frac{\partial}{\partial l}\left(\frac{v^2}{2}\right) + \frac{1}{\rho}\frac{\partial p}{\partial l} = 0$$

It is also considered that there exists a barotropic fluid function, $\widetilde{w}$. For gases, this assumption can be considered as valid if there is an homentropic flux. To ensure so, it is necessary to use the differential energy equation:

$$\rho T\frac{Ds}{Dt} = \nabla \cdot (k\nabla T) + \Phi_\mu + q_v$$

With the same assumptions that were used to develop the integral energy equation:

$$\frac{Ds}{Dt} = \frac{\partial s}{\partial t} + \vec{v} \cdot \nabla s = 0$$

Since it is stationary:

$$\nabla s = 0$$

and this is valid $\forall t$ because it is stationary and $\forall \vec{x}$ because fluid particles come from a region with uniform entropy. Therefore, it can be considered a homentropic flux.

Hence:

$$\frac{1}{\rho}\frac{\partial p}{\partial l} = \frac{\partial \widetilde{w}}{\partial l}$$

and the final equation is:

$$\frac{v^2}{2} + \widetilde{w} = cte$$

For ideal gases, the value of $\widetilde{w}$ is:

$$\widetilde{w} = h$$

and, as just seen:

$$s = cte$$

because there is homentropic flux.

Hence:

$$p_t = cte$$

$$T_t = cte$$

in the streamlines.

Appendix F: Combustion model

Example: Excess fuel

Consider a reaction in which excess fuel (F) is measured by the equivalence ratio, ϕ, which means that the reaction is settled (stoichiometric ratio of oxidant and fuel) when $\phi = 1$.

$$\phi F + O \rightarrow P' = P + (\phi - 1)F$$

$$\phi = \frac{(O/F)_{st}}{(O/F)} = \frac{\mathrm{OF}_{st}}{OF}$$

At the beginning, P' contains the products with the equilibrium composition which is the most plausible hypothesis, due to the lack of details related to the evolution along the camera. Under these conditions, P (products corresponding to combustion in stoichiometric ratio) is the fictional result, of subtracting the excess fuel $(\phi - 1)F$ from P', which can certainly lead to inconsistencies of various kinds that are ignored.

The energy equation, if there are two gas inlets, is written (enthalpy per unit of mass):

$$\phi \dot{m}_{Fst}\left(\Delta_f h_F + \left[h_F\left(T_{t,F}\right) - h_F\left(T_{ref}\right)\right]\right) + \dot{m}_O\left(\Delta_f h_O + \left[h_O\left(T_{t,O}\right) - h_O\left(T_{ref}\right)\right]\right) =$$

$$= (\dot{m}_O + \dot{m}_{Fst})\left(\Delta_f h_P + \left[h_P(T_c) - h_O\left(T_{ref}\right)\right]\right) +$$

$$+(\phi - 1)\dot{m}_{Fst}\left(\Delta_f h_F + \left[h_F(T_c) - h_F\left(T_{ref}\right)\right]\right)$$

Where $\dot{m}_{Fst}$ is the amount of fuel that produces the complete reaction with $\dot{m}_O$. If the inlet of oxidant were a liquid inlet, $T_{t,O}$ would be replaced by T_O and/or the inlet of fuel were a liquid inlet, $T_{t,F}$ would be replaced by T_F.

Dividing the previous expression by the mass flow of oxidant:

$$\frac{\phi}{(O/F)_{st}}\left(\Delta_f h_F + \left[h_F(T_{t,F}) - h_F(T_{ref})\right]\right) + \left(\Delta_f h_O + \left[h_O(T_{t,O}) - h_O(T_{ref})\right]\right) =$$

$$= \left(1 + \frac{1}{(O/F)_{st}}\right)\left(\Delta_f h_P + \left[h_P(T_c) - h_P(T_{ref})\right]\right) +$$

$$+ \frac{\phi - 1}{(O/F)_{st}}\left(\Delta_f h_F + \left[h_F(T_c) - h_F(T_{ref})\right]\right)$$

Heat of combustion

It is possible to define the heat of combustion of the stoichiometric mixture, per unit mass of oxidant, (exclusive property of the nature of the reactants and not of their ratio) as the heat required to raise the enthalpy of the products involved when oxidant and fuel take part in the stoichiometric ratio and depart from the baseline, such as:

$$Q_{comb} = \Delta_f h_O + \frac{1}{(O/F)_{st}}\Delta_f h_F - \left(1 + \frac{1}{(O/F)_{st}}\right)\Delta_f h_P$$

Whereby the energy equation is:

$$Q_{comb} = \left(1 + \frac{1}{(O/F)_{st}}\right)\left[h_P(T_c) - h_P(T_{ref})\right] +$$

$$+ \frac{\phi - 1}{(O/F)_{st}}\left[h_F(T_c) - h_F(T_{ref})\right] - \left[h_O(T_{t,O}) - h_O(T_{ref})\right] -$$

$$- \frac{\phi}{(O/F)_{st}}\left[h_F(T_{t,F}) - h_F(T_{ref})\right]$$

It is possible to define the average specific heat at constant pressure of the combustion products as:

$$c_{P,P'} = \frac{(\dot{m}_O + \dot{m}_{Fst})c_{P,P} + (\phi - 1)\dot{m}_{Fst}c_{P,F}}{\phi\dot{m}_{Fst} + \dot{m}_O}$$

which is similar to the one obtained by supposing a perfect mix. Besides, the adiabatic mixing equation between the oxidant and the fuel is used to calculate T_{in}. This equation states:

$$\dot{m}_O c_{P,O}\left(T_{in} - T_{t,O}\right) + \dot{m}_F c_{P,F}\left(T_{in} - T_{t,F}\right) = 0$$

Which leads to:

$$\left(1+\frac{\phi}{OF_{st}}\right)c_{P,R}T_{in}=c_{P,O}T_{t,O}+\frac{\phi}{OF_{st}}c_{P,F}T_{t,F}$$

where $c_{P,R}$ is:

$$c_{P,R}=\frac{\dot{m}_O c_{P,O}+\dot{m}_F c_{P,F}}{\dot{m}_O+\dot{m}_F}$$

Finally, the energy equation results:

$$\eta\cdot Q_{comb}=\frac{1+OF}{OF}\cdot\left(c_{P,P'}\left(T_c-T_{ref}\right)-c_{P,R}\left(T_{in}-T_{ref}\right)\right)$$

Attributing to η the necessary definitions to enable the equation to enjoy some precision and highlight the influence of distinguished parameters as the mixture ratio.

Example: Excess oxidant

Consider a reaction in which there is excess oxidant (O). Now the reaction is adjusted:

$$\phi F+O\rightarrow P'=\phi P+(1-\phi)O$$

$$\phi=\frac{(O/F)_{st}}{(O/F)}=\frac{\mathrm{OF}_{st}}{OF}$$

With excess oxidant, there is not enough fuel to obtain all the products corresponding to combustion in the stoichiometric ratio. Instead, ϕP is obtained.

The energy equation, if there are two gas inlets, is written (enthalpy per unit of mass):

$$\phi\dot{m}_{F,st}\left(\Delta_f h_F+\left[h_F\left(T_{t,F}\right)-h_F\left(T_{ref}\right)\right]\right)+\dot{m}_O\left(\Delta_f h_O+\left[h_O\left(T_{t,O}\right)-h_O\left(T_{ref}\right)\right]\right)=$$

$$=\phi\left(\dot{m}_O+\dot{m}_{F,st}\right)\left(\Delta_f h_P+\left[h_P(T_c)-h_O\left(T_{ref}\right)\right]\right)+$$

$$+(1-\phi)\dot{m}_O\left(\Delta_f h_O+\left[h_O(T_c)-h_O\left(T_{ref}\right)\right]\right)$$

If the inlet of oxidant were a liquid inlet, $T_{t,O}$ would be replaced by T_O and/or the inlet of fuel were a liquid inlet, $T_{t,F}$ would be replaced by T_F.

The same procedure of the previous example is followed. Q_{comb} is defined equally as in the previous example. And for $c_{P,P'}$:

$$c_{P,P'}=\frac{\phi(\dot{m}_O+\dot{m}_{Fst})c_{P,P}+(1-\phi)\dot{m}_O c_{P,O}}{\phi\dot{m}_{Fst}+\dot{m}_O}$$

This leads to the following expression:

$$\eta \cdot Q_{comb} = \frac{1 + OF}{OF_{st}} \cdot \left(c_{P,P'}\left(T_c - T_{ref}\right) - c_{P,R}\left(T_{in} - T_{ref}\right) \right)$$

Defining T_{in} and $c_{P,R}$ in the same way as in the previous example.

Generalised expression

It is now possible to build a generalised expression to be used in both cases, that is, with excess fuel or with excess oxidant:

$$\eta \cdot Q_{comb} = \frac{1 + OF}{min(OF\ , OF_{st})} \cdot \left(c_{P,P'}\left(T_c - T_{ref}\right) - c_{P,R}\left(T_{in} - T_{ref}\right) \right)$$

where $c_{P,P'}$ is:

$$c_{P,P'} = \frac{(1 - \max(1 - \phi, 0)) \cdot (\dot{m}_O + \dot{m}_{Fst}) c_{P,P}}{\phi \dot{m}_{Fst} + \dot{m}_O} +$$

$$+ \frac{\max(\phi - 1, 0) \cdot \dot{m}_{Fst} c_{P,F} + \max(1 - \phi, 0) \cdot \dot{m}_O c_{P,O}}{\phi \dot{m}_{Fst} + \dot{m}_O}$$

Note that this equation matches the equation given in the model of the “GasGen” and “CombCha” components to calculate $c_{P,P'}$.

Appendix G: On-off design mode rules

Turbomachinery rules

If component “Pump” is in design mode, the “Turbine_ch”, “Turbine” or “Turbine_liq” components connected to it must be in design mode too. If component “Pump” is in off-design mode, the “Turbine_ch”, “Turbine” or “Turbine_liq”components connected to it must be in off-design mode too.

If component “Compressor” is in design mode, the “Turbine_ch”, “Turbine” or “Turbine_liq” components connected to it must be in design mode too. If component “Compressor” is in off-design mode, the “Turbine_ch”, “Turbine” or “Turbine_liq” components connected to it must be in off-design mode too.

In design mode, the pump/compressor sets the power. The rotational speed is set by the turbine.

In off-design mode, the pump/compressor sets the rotational speed. The power is set by the turbine.

The mass flow in a pump/compressor is always set by another component. The mass flow in a turbine is set by another component in off-design mode and by itself in design mode.

If a turbine in design mode does not have a pump/compressor connected to it, it would be possible to set the mass flow to it as a boundary or by another component only if the power is not given as a boundary.

Exhaust rules

The “Injector” components don’t have to be in the same mode as the “CombCha” component connected to it.

A “CombCha” component plus its connected “Injector” components set the oxidant and the fuel mass flow in any mode.

If component “Nozzle” or “Ambient” are in off-design mode, the exhaust mass flow is set. If component “Nozzle” or “Ambient” are in design mode, the exhaust mass flow must be set by another component.

If component “Nozzle” is connected to a “Turbine_ch”, “Turbine” or “Turbine_liq” component, they must be in the same mode. This is because in design mode the turbine imposes the mass flow (with a pump or compressor connected to it or by giving the power of the turbine as a boundary) but in off-design mode the turbine does not impose it. Note that more data is given in the turbine design mode, than in the off-design mode.

If there is an “Injector” component downstream from a turbine, both must be in the same mode. If they are in off-design mode, there is no problem. If they are in design mode, it is not possible to choose Type=Known_pi. Instead, Type=Known_W should be chosen for the design mode of the turbine and data “W” set to the same value as the “W” data of the injector. Otherwise, only one of the data will be taken into account and the rest will not.

Issues to Consider

Note that inside a component with on-off design mode, the variables/data will be repeated. As an example, component “Ambient” will be analysed. It has data “A” which is the area, and variable “A_d”, which is the design area. In design mode, “A_d” is calculated, and data “A” is not taken into account. In off-design mode, A=A_d. The same thing happens in the other components.

In the **On-off design mode** section of each component, it is always considered that the inlet total pressure, the inlet total temperature (or temperature in the event of liquids) and the inlet working fluid (“fluid” variable) are known to say what will be calculated by the component in on-design and in off-design mode. This is what will happen if the user does not change anything.

Component “Inlet” when Type=All can set the mass flow.

Note that when a component is in off-design mode, the analysis is done as if it were manufactured because the given values are areas, lengths, etc.

Whenever this appendix states that a component is downstream from another, they do not have to be directly connected, they can have other components in between, but they must belong to the same flow line, which means that the mass flow is constant through the line (there are no “Junction” or “SplitFrac” components).