# LPRES Library 1.3.5

## Getting Started

**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 Introduction

This document is a start guide in which an example of how to use the LPRES Library to model the F-1 Rocket Engine is shown.

The LPRES Library, which name means **Liquid Propellant Rocket Engine Simulation**, is an educational library for the EcosimPro program that contains components to predict the behavior of the different configurations that a liquid propellant rocket engine can have.

EcosimPro and PROOSIS are powerful mathematical tools capable of modelling any kind of dynamic system represented by differential-algebraic equations (DAE) or ordinary-differential equations (ODE) and discrete events. They are integrated visual environments that provide intuitive tools to make simulation easy.

The tools provide an object-oriented acausal approach to create reusable component libraries. They are based on symbolic and numerical methods capable of processing complex systems of differential-algebraic equations. Their smart wizards provide modellers with an easy way to build consistent mathematical models.

EcosimPro and PROOSIS are used by leading companies in the Aerospace and Energy sectors. The European Space Agency (ESA) has chosen EcosimPro as its recommended tool for simulation in several fields, including propulsion, environmental control systems and life support (ECLSS) and power systems. Over the last 15 years, ESA has turned to EcosimPro to model, among others, complex systems of the International Space Station (ISS), rocket propulsion systems and biological systems.

The European Space Propulsion System Simulation (ESPSS) is an ESA toolkit to model rocket and satellite propulsion systems. It consists of a set of libraries based on the EcosimPro simulation environment.

The ESPSS toolkit provides components and functions for the simulation of launch vehicle and spacecraft propulsion systems able to work under transient and steady conditions. ESPSS also includes a complete database of fluids to be used as propellants, pressurizing fluids or other applications.

The ESPSS is of a professional nature, so it is highly complex. 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 easier than the learning curve of the ESPSS toolkit.

A complete list of the components of the LPRES Library can be found in the **LPRES Library Reference Manual**. This manual also includes a complete description of the physical modelling of each component.

A complete manual of the examples performed with the LPRES Library can be found in the **LPRES Library Test Cases**. These examples are included in the LPRES_EXAMPLES Library.

# 2 F-1 rocket engine simulation with the LPRES Library

The first step will be the installation of the library in the EcosimPro program and then learning about the key concepts of EcosimPro. This will be followed by a description of the rocket engine to be simulated and finally, the simulation.

## 2.1 Library Installation

First of all, the LPRES Library folder may be placed anywhere on the computer, although it is recommended to create a folder (named, for example, “MY_LIBS”) inside the EcosimPro folder itself and place in it the folder where the LPRES Library is included.

It should be borne in mind that the folder where the LPRES Library is located can not be compressed.

Open EcosimPro.

Create a new workspace by clicking on **File** > **New** > **Workspace**.

Before opening the LPRES Library in the workspace, it is necessary to open the MATH Library with version 3.1 (at least), which is one of the libraries included with the EcosimPro program.

To do this, click **File** > **Open** > **Library**.

The open dialogue box appears to specify the library to be used.

Select the EcosimPro folder, then the **USER_LIBS** folder and finally the **MATH** folder.

Select the **MATH.lsp.xml** file.

Once the library is open, it will be displayed at the top of the workspace area and the files, items and other components belonging to it will be displayed at the bottom.

Open the LPRES Library repeating the same steps, but this time, searching for the **LPRES.lsp.xml** file inside the LPRES Library folder.

Now the library is already loaded and it can be compiled. To do this, right-click on the LPRES Library and click on **Compile Sources**.

## 2.2 Key Concepts

EcosimPro has five fundamental concepts:

- **Components:** These represent models of the systems simulated by means of variables, differential-algebraic equations, topology and event-based behaviour. A component is the equivalent of the "class" concept in object-oriented programming.

- **Ports:** These define a set of variables to be interchanged in connections and the behaviour and restrictions when there are connections between more than two ports. For instance, an electrical connection type uses voltage and current as variables to be used in connections. The connection port avoids having to connect individual variables; instead, sets of variables are managed together.

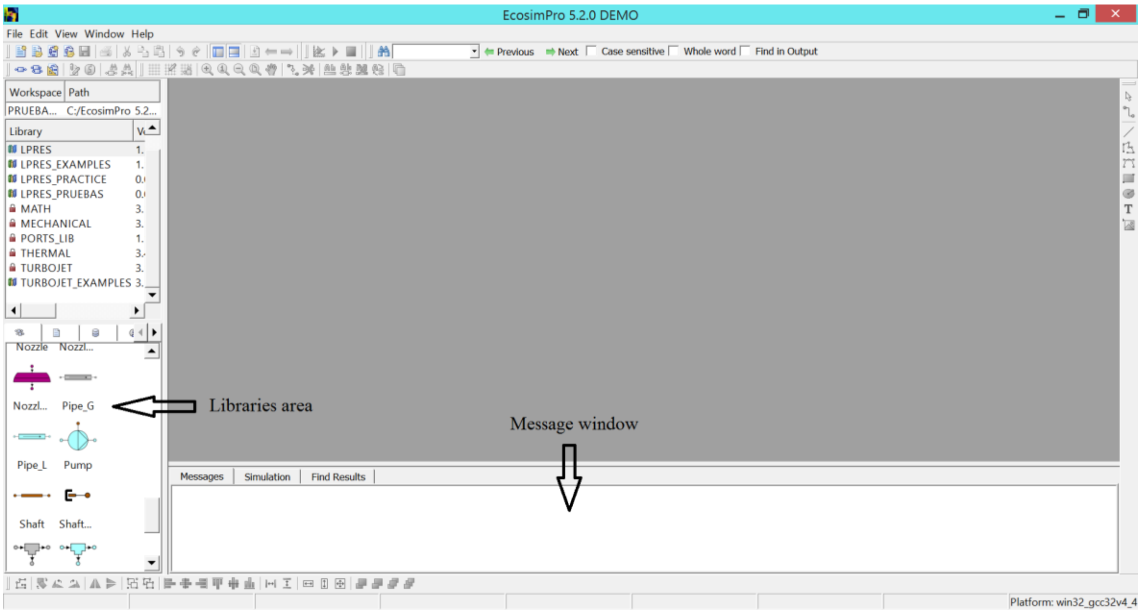
- **Partitions:** To simulate a component, you first have to define its associated mathematical model; this is called a partition. A component may have more than one partition. For example, if a component has several different boundary conditions, depending on the set of variables selected, each set of variables produces a different mathematical model, or partition. The next step is to generate experiments for each partition. The partition defines the causality of the final model.

- **Experiments:** The experiments performed for each partition of the component are the different simulation cases. They may be trivial to calculate a steady state experiment or they may be very complex to calculate an experiment with many steady and transient states changing multiple variables in the model.

- **Libraries:** All components are classified by disciplines into libraries.

### 2.2.1 Overview of the working screen

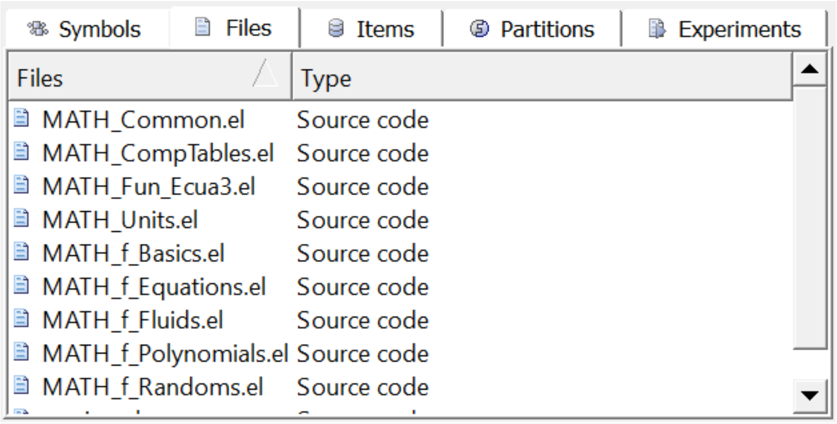
This is the screen that appears once EcosimPro is opened with a workspace and libraries loaded in it:



| Workspace | Path |
|---|---|
| PRUEBA... | C:/EcosimPro 5.2... |

| Library | V |
|---|---|
| LPRES | 1. |
| LPRES_EXAMPLES | 1. |
| LPRES_PRACTICE | 0. |
| LPRES_PRUEBAS | 0. |
| MATH | 3. |
| MECHANICAL | 3. |
| PORTS_LIB | 1. |
| THERMAL | 3. |
| TURBOJET | 3. |
| TURBOJET_EXAMPLES | 3. |

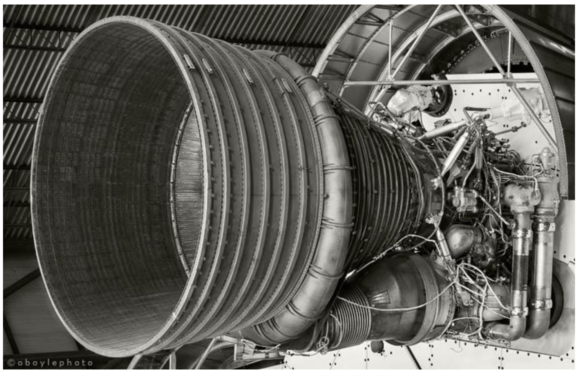
The **Libraries area** consists of 5 tabs that group the files and text and graphical modelling functionalities, as well as the different types of experiments that can be designed based on the modelling.



| Files | Type |
|---|---|
| MATH_Common.el | Source code |
| MATH_CompTables.el | Source code |
| MATH_Fun_Ecua3.el | Source code |
| MATH_Units.el | Source code |
| MATH_f_Basics.el | Source code |
| MATH_f_Equations.el | Source code |
| MATH_f_Fluids.el | Source code |
| MATH_f_Polynomials.el | Source code |
| MATH_f_Randoms.el | Source code |

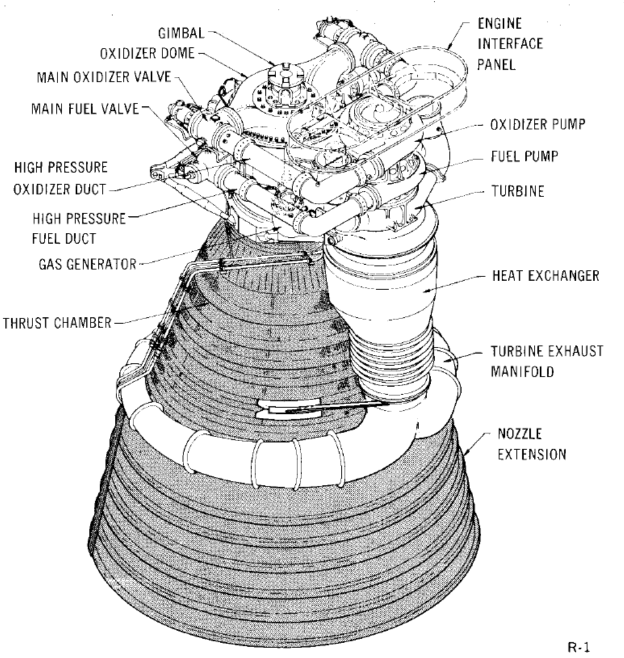
## 2.3 F-1 rocket engine

The F-1 is a gas-generator cycle rocket engine developed in the United States by Rocketdyne in the late 1950s and used in the Saturn V rocket in the 1960s and early 1970s. Five F-1 engines were used in the S-IC first stage of each Saturn V, which served as the main launch vehicle of the Apollo program. The F-1 remains the most powerful single-combustion chamber liquid-propellant rocket engine ever developed.

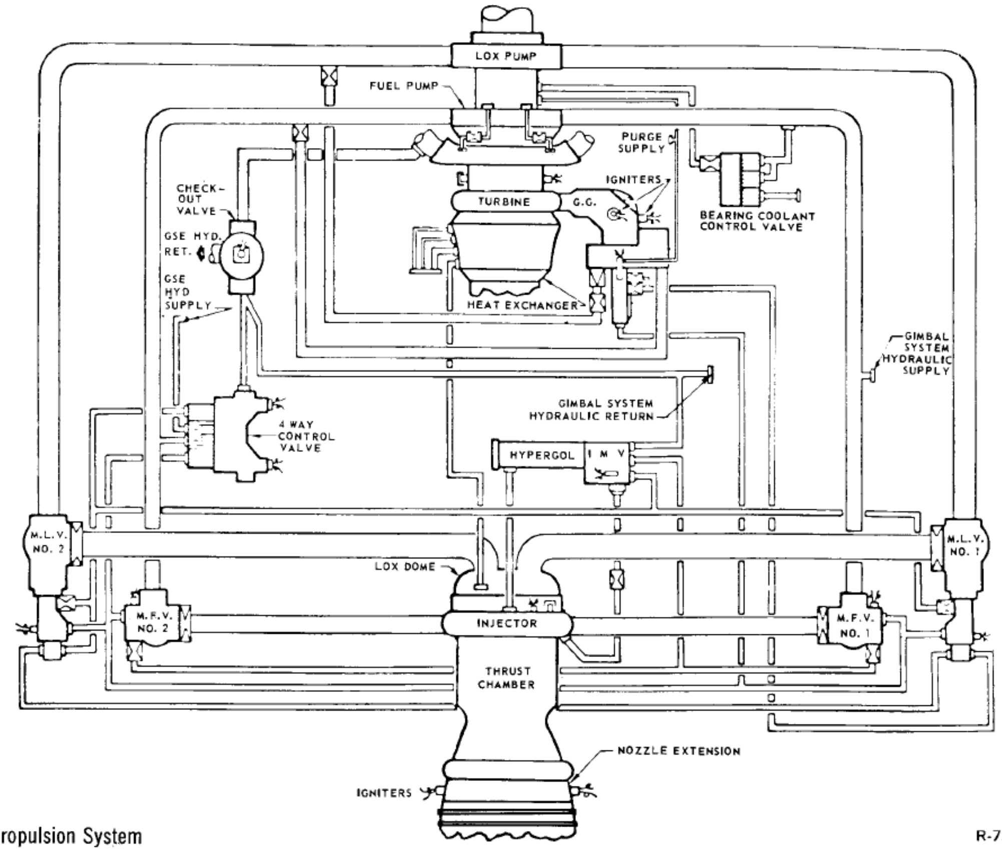


The F-1 engine is a single-start, 6.77 MN-fixed-thrust, bipropellant rocket system. The engine uses LOX (liquid oxygen) as the oxidiser and RP-1 (kerosene) as fuel. This is a fact sheet which summarises the performance of the rocket engine:

| Magnitude | Value |
|---|---|
| Thrust (sea level) | 6.77 MN |
| Specific impulse (sea level) | 263 s |
| Thrust (vacuum) | 7.77 MN |
| Specific impulse (vacuum) | 304 s |
| Combustion temperature in the gas generator | 1069 K |
| Combustion temperature in the combustion chamber | 3573 K |
| Expansion area ratio | 16:1 with nozzle extension<br>10:1 without nozzle extension |
| Combustion chamber pressure | 68 bars |
| Oxidizer flowrate | 1789 kg/s |
| Fuel flowrate | 788 Kg/s |
| Mixture ratio | 2.27:1 oxidiser to fuel |
| Increasing pressure in the oxidizer pump | 90 bars |
| Increasing pressure in the fuel pump | 131 bars |



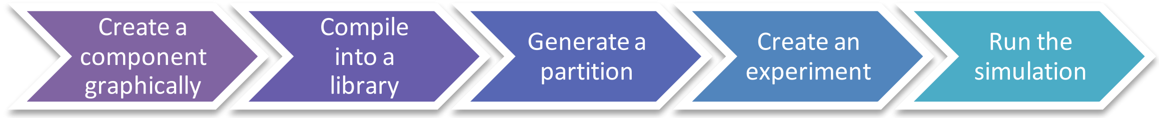
Some of these values will be given to the program and values similar to the others should be obtained from the simulation.



F-1 Propulsion System R-7A

## 2.4 Simulation

The process to simulate a physical system in EcosimPro consists of the following steps:



Create a component graphically

Compile into a library

Generate a partition

Create an experiment

Run the simulation

### 2.4.1 First steps

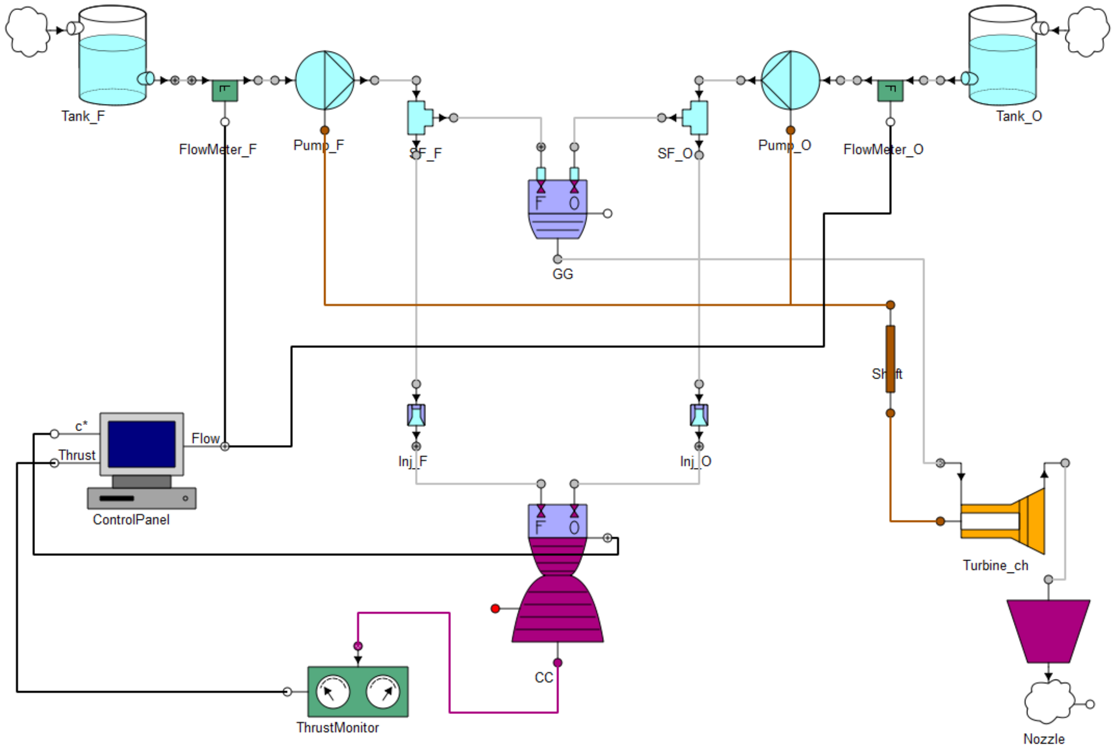
To start, create a new library by clicking on **File** > **New** > **Library**.

And now create a new schematic inside this library. To do this, click on **New Schematic** on the general toolbar of the application.

The schematics are where components are created graphically by adding components that are already created on them. On the left-hand side of the window is the palette of available components (one icon for each component). To select the components of the

LPRES Library, click on it, and the palette with them automatically appears in the **Symbols tab**.

Drag and drop components from the palette onto the canvas. Select the **Draw Connector** option on the Drawing toolbar to join the components between them in order to create what appears in the following figure:



Double clicking on an object displays a window to edit the object attributes. The attributes that the objects must have are shown in the table below. A component has construction parameters, data and variables. The attributes are the construction parameters and the data. The variables are calculated in the simulation, they are not attributes of the component, so they can not be established here.

| Object | Name | Data name | Value |
|---|---|---|---|
| **Tank which feeds the fuel channel** | Tank_F | fluid | RP_1 |
| | | T_d | 300 |
| **Tank which feeds the oxidant channel** | Tank_O | fluid | LOX (dafault) |
| | | T_d | 85 |
| **Fuel channel flow meter** | FlowMeter_F | | |
| **Oxidant channel flow meter** | FlowMeter_O | | |
| **Fuel channel pump** | Pump_F | eta_d | 0.85 |
| | | phi_d | 0.0428 |

| Object | Name | Data name | Value |
|---|---|---|---|
| | | psi_d | 0.108 |
| | | U_0 | 500 (default) |
| | | Type | Design (default) |
| | | dp | 131e5 |
| **Oxidant channel pump** | Pump_O | eta_d | 0.9 |
| | | phi_d | 0.0428 |
| | | psi_d | 0.108 |
| | | U_0 | 500 (default) |
| | | Type | Design (default) |
| | | dp | 90e5 |
| **Fuel channel split frac** | SF_F | TPL | 1 |
| **Oxidant channel split frac** | SF_O | TPL | 1 |
| **Gas generator** | GG | TPL_d | 1 |
| | | eta_d | 1 |
| | | OF_st | 2.27 |
| | | Q_comb | 11e6 |
| | | cp_P | 2169 |
| | | M_P | 23 |
| | | OF | 0.18 |
| | | W_F0 | 100 (default) |
| **Choked turbine** | Turbine_ch | eta_d | 0.8 |
| | | alpha_2 | 45 (default) |
| | | Type | Known_W |
| | | rpm | 5550 |
| | | W | 77.11 |
| **Nozzle** | Nozzle | Type | Design (default) |
| **Shaft** | Shaft | eta | 1 (default) |
| **Injector connected with the fuel inlet of the combustion chamber** | Inj_F | C_D | 0.5(default) |
| | | Type | Design (default) |
| | | W | 100 (default) |
| **Injector connected with the oxidant inlet of the combustion chamber** | Inj_O | C_D | 0.5(default) |
| | | Type | Design (default) |
| | | W | 100 (default) |
| **Combustion chamber** | CC | eta_d | 1 |
| | | OF_st | 2.27 |
| | | Q_comb | 11e6 |
| | | cp_P | 2169 |
| | | M_P | 23 |
| | | AR | 10 (default) |
| | | Pc_0 | 5000000 (default) |
| | | Tc_0 | 4000 (default) |
| | | W_F0 | 1000 (default) |
| | | Type | Design (default) |
| | | p_c | 6800000 |
| | | Cooled | No |

| Object | Name | Data name | Value |
|---|---|---|---|
| **Thrust monitor** | ThrustMonitor | | |
| **Control panel** | ControlPanel | | |

Finally, selecting the **Compile** option creates a new component in EcosimPro if there are no mistakes.

### 2.4.2 Partition creation

Now is time for the partition generation. To create a custom partition, go to the **Partitions tab** and right-click on the component the partition is being created for. A context menu will appear, where the **New Custom Partition** option should be selected.

Once the wizard was opened, select the **Use wizard to transform data into unknown variables** setting and click next.

The panel that appears is used to convert component data to variables. To do this, they have to be selected by checking the box on the left of the name and clicking the **< Select** button. Perform it with the `Inj_F.W` and `Inj_O.W` data and click next.

In the next panel, the user establishes the given values of the system of equations that appears when creating a component in order to have the same number of equations and unknowns. Some variables are considered boundary variables, whose values will be given in the experiment. Switch the selected item of the combo box **Categories** to **Able to be selected**. Then, choose thevariables listed below:

- `LPRES.Altitude`
- `FlowMeter_F.f_in.W`
- `FlowMeter_O.f_in.W`

If the names of these variables are not found, an equivalent variable can be selected. The equivalent variables for `FlowMeter_F.f_in.W` are:

- `FlowMeter_F.f_out.W`
- `Pump_F.f_in.W`
- `Pump_F.f_out.W`
- `SF_F.f_in.W`
- `Tank_F.l.W`

For `FlowMeter_O.f_in.W`, the equivalent variables are:

- `FlowMeter_O.f_out.W`
- `Pump_O.f_in.W`
- `Pump_O.f_out.W`
- `SF_O.f_in.W`
- `Tank_O.l.W`

Click next and click next too in the following panels while possible. Finally, click generate and finish in the last panel. The partition is finished.

### 2.4.3 Experiment creation and simulation

Go the **Experiments tab**. To create an experiment, right-click on the partition where it is to be created. From the menu which appears, select the **New Steady Experiment** option.

A source code file written in EL Language, the own language of the program EcosimPro program, will appear. It is time to modify the generated code to impose the desired boundary conditions, etc.

**EL Language**

To know a little bit about EL Language:

The piece of code headed by the keyword `BOUNDS` includes the values of variables considered as boundary variables in the partition. Replace the values by the ones desired. These values are the final values of the variables.

To indicate the results that you want to obtain from the experiment, write the following below the word `STEADY():`

```
PRINT(" $variable ")
```

Once the experiment is run, the program will write the value of the variable as a message.

It is possible to write more things between the "" such as the identification of the variable:

```
PRINT(" variable_1 = $variable ")
```

In this case, the message will be: `variable_1 = variable_value`

To know how to write a variable in EL Language:

- If the variable is associated to a port: `component_name.port_name.variable_name`
- If the variable is not associated to a port, but is as variable within the component itself: `component_name.variable_name`

This information is enough to know how to modify the generated code.

### Editing the generated code

The BOUNDS are:

- LPRES.Altitude = 30000
- FlowMeter_F.f_in.W = 788
- FlowMeter_O.f_in.W = 1789

And finally, for the BODY, here is an example:

```
BODY

      STEADY()


      PRINT(" ")
      PRINT("---------------------------------------------------
--------------")
      PRINT(" ")
      PRINT(" RESULTS:")
      PRINT(" ")
      PRINT(" Oxidizer pump input area = $Pump_O.A_in_d")
      PRINT(" Fuel Pump input area = $Pump_F.A_in_d")
      PRINT(" Oxidizer pump average radius = $Pump_O.r_m_d")
      PRINT(" Fuel Pump average radius = $Pump_F.r_m_d")
      PRINT(" Turbine input area = $Turbine_ch.A_in_d")
      PRINT(" Turbine discharge area = $Nozzle.A_d")
      PRINT(" Oxidizer injector area = $Inj_O.A_d")
      PRINT(" Fuel injector area = $Inj_F.A_d")
      PRINT(" Throat area = $CC.A_th_d")

      PRINT(" Combustion temperature in the combustion chamber =
$CC.T_c")
      PRINT(" Combustion temperature in the gas generator =
$GG.g.Tt")

      PRINT(" Thrust = $ControlPanel.Thrust")
      PRINT(" Isp_0 = $ControlPanel.Isp_0")
      PRINT(" C_E = $ControlPanel.C_E")
      PRINT(" c* = $ControlPanel.c_star")
      PRINT(" ")
```

The code can be compiled by clicking on the standard toolbar. Run the simulation by clicking on the button on the tool bar. The experiment will start and the associated messages will appear in the **Message window**.

## 2.5 Off-design calculation

The previous simulation was a design calculation. Now, an off-design calculation is performed.

Set the pumps, the nozzle situated downstream from the turbine, the turbine, the injectors and the combustion chamber to Type=Off_design mode. Set the area and average radius of each pump, the turbine input and discharge areas, the area of the injectors, and the throat area of the combustion chamber to the values obtained in the previous experiment.

To reach convergence, the value of data "W_F0" of the combustion chamber and the same data of the gas generator must be changed to values which are near to solution. The values given have been 150 in the gas generator and 2000 in the combustion chamber.

Compile and go to the **Partitions tab**, right-click on the component the partition is being created for. A context menu will appear. In this case, the **New Default Partition** option should be selected.

Then, create a steady experiment and set

```
LPRES.Altitude = 30000
```

and run the experiment.

To see the performance of the engine, the altitude can be changed to other values such as 0 for sea level and 1000000 for vacuum (space altitudes, out of the atmosphere).

It can be checked that the results obtained in off-design mode for the design altitude (30000 $m$) are the same as the results obtained in design mode.